Mobility Enhanced RPL for Wireless Sensor Networks

Inès El Korbi*, Mohamed Ben Brahim*, Cedric Adjih† and Leila Azouz Saidane*
*National School of Computer Science, University of Manouba, 2010 Tunisia
†Inria Paris-Rocquencourt, Domaine de Voluceau, 78153 Le Chesnay Cedex, France
* {ines.korbi, leila.saidane}@ensi.rnu.tn, brahim.muhamed@gmail.com
†cedric.adjih@inria.fr

***Abstract*—In this paper, we investigate the problem of supporting mobility over RPL (IPv6 Routing Protocol for Low power and Lossy Networks) when applied to route traffic in Wireless Sensor Networks (WSNs). RPL is a routing protocol adapted for information routing with low power, low storage and processing sensor devices, in static topologies commonly found in WSNs, but which is not directly designed for mobile scenarios. Specifically, RPL actively decreases control traffic, at the price of lower reactivity to topology changes. In this paper, we propose to introduce some new mechanisms to the native RPL that reconcile decrease in control traffic and reactivity. They are based on an identification of mobile nodes, and furthermore they enhance RPL behavior in case of node mobility. Our approach will be, henceforth, called ME-RPL (Mobility Enhanced RPL).**

I. Introduction

A typical Wireless Sensor Network (WSN) is a collection of wireless sensor nodes operating in a collaborative way to capture events (temperature, humidity, etc.) within a given area and forwarding the captured data to a collector node called "sink node" for processing. In the last years, WSNs raised much interest and are being deployed in various applications fields: environment, industry, military, e-heath, etc.

In such networks, sensor nodes communicate through wireless links to route the collected information to sink nodes. Due to wireless sensor characteristics, routing algorithms used for MANETs (Mobile Ad hoc NETworks) have rapidly shown their limits and do not perform well with WSNs. Then, several routing solutions have been proposed for WSNs [1], [2], [3], [4], [5], but each solution focuses on a particular aspect of wireless nodes (energy resource optimization, time processing reduction within the network, etc.). In the standardization area, RPL has been introduced at the IETF in 2008 [6] as a proposed standard routing protocol for LLNs (Low Lossy Networks) [7]: LLNs are defined as a class of computer networks characterized by their high loss, low capacity interconnection links, and low power equipments; they include wireless sensor networks. The RPL protocol is therefore a data link independent and reliable routing protocol, suitable for such networks.

In this article, we focus on the RPL protocol; we investigate the problem of adapting it to support mobility in WSNs. Indeed, in practical networks, some mobile nodes might be present: for instance, patrolmen, or nodes attached to mobile physical equipment, or apparent "mobility", due to regularly broken links. A common case, is the presence of a minority of mobile nodes within a larger set of fixed sensor nodes. The default RPL specification [6] does not consider mobility in its design goals: as a result, a number of issues appear when some nodes are mobile (see for instance [8], [9]), mostly related to the slow reaction of the RPL protocol with respect to topology changes. Our contribution is the proposal of ME-RPL (Mobility Enhanced RPL), a full set of extensions designed to integrate mobility in the RPL protocol. It is based on prior identification of mobile nodes (or equivalently, explicit identification of nodes with frequently unstable links). Then, the protocol can operate efficiently by enforcing a different behavior for mobile nodes (e.g. better reactivity), and by considering mobile nodes differently (e.g. avoiding routes through them).

The rest of the paper is organized as follows. In section II, we present some background concepts related to our study, including some well known routing solutions in wireless sensor networks and RPL basics. In section III, we expose the motivation of our current work, an analysis of the issues of native RPL with respect to mobility, and some related work. In section IV, we present the different mechanisms introduced by our proposed approach called ME-RPL (Mobility Enhanced RPL) to support mobility in RPL. Section V evaluates the performance of both ME-RPL and the native RPL using the COOJA/Contiki simulator [10]. Finally, we conclude the paper and present directions for future work in section VI.

II. Background

A. Routing protocols in wireless sensor networks

Due to the particularities of sensor devices (low energy, low memory and low processing capacities), routing protocols used in wireless ad hoc networks have rapidly shown their limits when being used in WSNs. New flat and routing hierarchical algorithms have then been proposed to fit with the wireless sensor nodes particularities and their applications. In flat routing, all the nodes within the network have the same role and collaborate together to perform the same task. In this category, SPIN (Sensor Protocols for Information via Negotiation) [1] and Direct Diffusion [2] are the most popular solutions. Other protocols [3], [11] have been offering flat routing solutions. Initially designed to ensure network scalability, hierarchical solutions allow energy-efficient routing when being used in WSNs. For instance, LEACH (Low

Energy Adaptive Clustering Hierarchy) [4] is a cluster-based routing protocol that randomly selects few cluster-heads in the network to aggregate data coming from cluster nodes and to send it the sink node. PEGASIS (Power-Efficient Gathering in Sensor Information Systems) [5] is another cluster based routing allowing the nodes only to communicate with their closest neighboring with period sink node communications. Both WSNs flat and hierarchical routing introduce mechanisms to take into account some sensor nodes characteristics. But these routing mechanisms lack several standardization features which makes their interoperability with other network technologies impossible. On the other hand, RPL builds upon prior research on WSNs and focuses on practical issues such as IP compatibility; it is a new IPv6 routing protocol designed for LLNs. It uses the 6LoWPAN (IPv6 Low power Wireless Personal Area Networks) [12] adaptation layer for sending IPv6 packets over LLNs data link layer using encapsulation and header compression mechanisms. 6LoWPAN is designed for the IEEE 802.15.4 medium access layer [13]. In the following, we present the RPL protocol basics.

B. The RPL protocol presentation

RPL is an IPv6 distance vector routing protocol for LLNs. It is designed to operate with low memory devices, and low data traffic; hence in the most minimal version each node is able to maintain only one route to one root node in the network (the sink), building a single tree for the network. In a more general case, RPL allows redundancy in the tree (several parents) and therefore RPL actually constructs a Destination Oriented Direct Acyclic Graph (DODAG), used to route traffic from multipoint-to-point devices inside the network towards one or several central control points (DODAG root(s)). Further options allow point-to multipoint traffic from the central control point(s) to the devices as well. Building the DODAG requires the computation of an objective function -that operates on a combination of metrics and constraints to compute the 'best' path- and the usage of new ICMPv6 (Internet Control Messages Protocol) messages adapted to the RPL context.

1) RPL messages: RPL introduces four control messages required for DODAG construction and maintenance:

- **DIO** (DODAG Information Object): broadcast message sent by the DODAG root(s) to initially trigger the DODAG construction, and later by router nodes in the DODAG. This message contains general information required to build the DODAG, for instance the DODAG_ID, the RPL_Instance_ID, the DODAG_Version_Number, the emitter node rank, the objective function with corresponding metrics/constraints.
- **DIS** (DODAG Information Solicitation): message designed to be sent by a new node to join the DODAG.
- **DAO** (Destination Advertisement Object): message sent by the non-root devices to permit parent nodes to record reverse paths to the multipoint devices.
- **DAO-ACK** (DAO Acknowledgment): message sent to acknowledge the reception of a DAO message.

2) DODAG Construction: the DODAG root initializes the DODAG building by sending a first DIO broadcast message. A node j receiving DIO messages uses the information conveyed in these DIO messages to determine a set of nodes -called the Parent Set- allowing him reaching the DODAG root.

Among the Parent Set, a "Preferred Parent (PP)" node -having the lowest rank- is selected by each node j to ensure routing the traffic to the DODAG root. After, the PP selection, the node j, computes its own rank using the objective function (involving a potential set of metrics/constraints received in DIO messages), generates a new DIO message and broadcasts it. The process is repeated until each node within the network has joined the DODAG, and then continues for DODAG maintenance.

3) DODAG Maintenance and Trickle: RPL is designed for energy-efficiency, hence one central mechanism is to decrease the amount of generated control traffic (RPL messages), when the network is in a stable state.

This mechanism is "Trickle" [14], an algorithm that progressively slows down control messages transmission every time it receives a "consistent" message (e.g. implying no change of topology), and reverts to a faster rate when there is a change (by resetting timer). For Trickle, a received message is "inconsistent" if it does not agree with the data that the receiver already has: by extension, in this case, we will call the receiver an "inconsistent node".

Precisely, RPL transmits the DIO messages with a Trickle (variable) interval size, and a Trickle timer. A received DIO message, that does not cause a change to the parent set, to DODAG version, etc., is consistent, and thus does not cause a Trickle timer/interval reset. If at the end of the current Trickle interval, all received DIO messages are consistent, the interval size is doubled. A DIO is generated at most once per interval.

Hence, in steady-state, the DIO message rate will be slowed exponentially, until reaching the limit configured for the DODAG.

To avoid broadcast storms, note that when a change is detected, Trickle has some latency: it does not transmit immediately; instead its timer is reset, and the transmission of a new message will occur normally within the (now shorter) interval [14].

4) Auxiliary operations: leaf nodes, P2P communications: Within the network, nodes can be configured as "router nodes". Then, they can start advertising topology information to their neighboring peers. Other nodes called "leaf nodes", can simply join the graph but do not send any DIO messages.

RPL was originally designed for MP2P (Multi Point to Point) traffic, however, P2P (Point to Point) communications can also be supported by RPL [15].

III. Motivation and Related Work

Since RPL is adapted to wireless sensor networks, we aim in this paper to use it to route information in networks where both fixed and mobile nodes coexist together to fit with particular application requirements. Compared to the fixed WSNs, mobile WSNs raise new challenges such as better energy efficiency, improved coverage, enhanced target tracking and superior channel capacity [16]. We start with an analysis of the challenges met when using RPL for mobile networks.

A. Overview of the issues with mobility

In this section, we give a short overview of some performances issues with RPL. We have identified three central issues:

- Lack of identification of mobile nodes. RPL does not differentiate between mobile nodes and non-mobiles nodes. Ideally, the routing protocol would take the difference into account for avoiding suboptimal choices, but this is not proposed in default RPL: the proposed metrics at the IETF (such as MRHOF or Objective Function Zero [17]) do not detail how to optimize routing in presence of mobile nodes.
- Inherent design for static networks: RPL integrates the Trickle mechanism to reduce control traffic, under the assumption that it would be redundant once the topology is acquired (the DODAG is built), see section II-B3. The general issue with Trickle is that, in case of temporary stability in one area of the network, the local control message rate may decrease to such a point that the discovery of topology changes (due to mobility) could be slower than desired.
- Limited, local, adaptability. It is a corollary of the precedent issue with mobility: when a node detects changes (which might be later than desired, see previous issue), RPL and Trickle will reset the timers locally to increase response time until the topology is stable again. However the node that detects the topology change is not necessarily the node(s) that should act to recover from the changes, and therefore it might not be sufficient to update control message timers, to resolve the situation efficiently. For instance, if a node has lost his (preferred) parent(s), the nodes able to resolve the issue, are its new potential parents, and not only the node itself.

B. Detailed analysis of the issues with mobility

In this section, we elaborate on the three mobility issues listed in previous section, on specific examples:

- Lack of identification of mobile nodes: specifically, RPL does not differentiate between fixed and mobile nodes in the preferred parent selection procedure. This may result in important packet loss within the sub-graph whose parent is a mobile node if this parent disappears.
- Both the issues of inherent design for static networks and of limited, local, adaptability appear when considering DIO messages and parent selection: specifically, the DODAG routing information is refreshed through DIO messages whose generation is based on the Trickle time interval. Once a node detects that it is inconsistent (that is either its rank, its parent set or its preferred parent changed) upon the reception of a DIO message [8], it reinitializes its Trickle interval to a minimum value that is to allow a fast DAG topology refresh. Otherwise the Trickle interval is increased until reaching a maximum value. The problem with this DIO message exchange procedure is that only the node observing inconsistent information (for example a preferred parent change) reinitializes its Trickle interval. The preferred parent causing the node inconsistency may however continue incrementing its Trickle interval. A frequent number of parent changes within a node j may denote that the node j is mobile or that is found in a mobile environment. Hence, reinitializing the Trickle interval for the node j may not be sufficient (that is the parent nodes have not updated their Trickle since they have experienced no node inconsistency). The node j may then solicit more frequent DAG update information to be sure to be always attached to the DAG. This aspect is not performed with the current version of RPL.
- On the other hand, the DODAG solicitation mechanism is designed to allow nodes within the network that want to join the DODAG to send a DIS message if no DIO message is received during an RPL_DIS_Interval [6]. For instance, DIS messages can be used to enforce a DIO solicitation (and potential Trickle resets) in the case of node mobility. The algorithm using this mechanism should be designed to solve some requirements due to mobility.

C. Solution space and related work

In order to handle mobility issues with standard RPL, it is possible to adjust RPL configuration and parameters. For instance:

- configure all mobile nodes as RPL leaf nodes
- configure Trickle with a low maximum message interval (forcing frequent control message generation)

These solutions would allow RPL to operate with the required reactivity for correctly managing mobility. However, they come at the expense of an unnecessary increase of the control message traffic. Using RPL mobile nodes exclusively as leaf nodes could also unnecessarily make the network unconnected.

Since such simple configuration of standard RPL is suboptimal, a few works have considered the impact of mobility on RPL behavior, and have proposed sophisticated solutions [8], [9]. The authors of [8], [9] considered the usage of RPL for the VANets (Vehicular Ad hoc Networks), where vehicles -mobile nodes- move within the network at different speed levels. They introduce a comprehensive description and analysis of RPL under mobility, and some improvements to the native RPL. Their improvements are two folds. First they improve the reactivity of the protocol with immediate control message generation: for assessing immediately link quality (immediate ETX probing for a new neighbor), for quickly propagating and updating the routing graph topology: immediate DIOs/DAOs message generation upon new parent election. Second they introduce a loop avoidance and detection mechanism.

The approach of the current article is complementary to their proposal: the modified version of RPL proposed in [8], [9] is designed for VANETs, and therefore assumes that all nodes in the networks are equal and mobile (except for the AP). However, in the context of WSNs, where both fixed and mobile nodes co-exist, we are able to introduce some differentiation between fixed and mobile nodes when establishing new routes: we are then able to derive new protocol changes by building upon this differentiation.

IV. Mobility-Enhanced-RPL

In this section, we present the new mechanisms of ME-RPL to support node mobility in WSNs over the native RPL protocol.

A. Protocol Overview

In section III-A, we have presented an overview of the issues of mobility for RPL. In this section, we summarize changes to the RPL protocol:

- explicit mobility advertisement: we assume that mobile nodes are identified (e.g. patrolman), and furthermore advertise their mobility status in control messages.
- improvement of the preferred parent selection: the parent selection algorithm is modified so that a node will prefer a fixed node as parent (and thus routes towards mobile nodes will be avoided).
- adaptation of the speed of solicitation messages, so that mobile nodes query their neighborhood faster, and at the same time causing signaling from nodes that have detected topology changes, to nodes that should help resolve them.

1) Handling the mobility information: In our context, we consider that we are in presence of wireless sensor networks where both mobile and sensor nodes co-exist within the network [16]. Moreover, we assume that the DODAG root is a fixed node. This assumption can be easily justified since in the most cases, the DODAG root is the collecting node that is generally connected to the wired LAN to forward the collected information to the application user for processing. The case of a truly mobile sink makes optimizations difficult and is typically handled by a simple increase of control message traffic (cf. section III-C).

A mobile node is a particular sensor node equipped with a mobility engine allowing it to move within the whole network for instance. In ME-RPL, this mobility capability is configured within each node, so as to make the difference between fixed and mobile nodes. Indeed, RPL already classifies a node within the network into three categories: DODAG root, router node and leaf node. In our mobility enhanced version of RPL, we add a fourth node category: a mobile node.

Hence, when a node is configured as a mobile node, this information has to be conveyed within the DIO message. Then, when receiving a DIO from a mobile node, a node has to take into consideration the mobile node received information when selecting its preferred parent. This new preferred parent selection procedure is presented in the next paragraph. Inserting the mobile node information within the DIO message is illustrated in figure 1.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| RPLInstanceID |Version Number |             Rank              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|G|0| MOP | Prf |     DTSN      |     Flags     |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                            DODAGID                            +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Mobility Option| 1 (Opt. Len.) |Mobility Status|   Option(s)...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 1. DIO base object format with mobility option

In the DIO message, we add a DIO option (respecting the format of RPL message options) the “Mobility Information” option situated within the set of options in the DIO base object. According to the RPL specification [6], DIO options may be added to DIO messages and are ignored by the receiver when it does not know how to handle them. Then, a mobility information option carrying a single 8-bit value (yielding a total option size of 3 bytes) within the DIO base object can be used to convey the node mobility status:

- If a node is configured to be a mobile node, the “Mobility Status” field will be initialized to ‘1’ in all the DIO messages.
- Otherwise, the “Mobility Status” field will be initialized to ‘0’, or alternatively, the Mobility Information Option will not be present in the DIO.

The receivers of the DIO messages will react differently depending on the “Mobility Status” field value.

2) The preferred parent selection procedure: In this paragraph, we present our new preferred parent selection procedure that takes into account the node mobility information conveyed in DIO messages. Indeed, as we have mentioned in the motivation section, the native RPL protocol does not consider the impact of choosing a mobile node as a preferred parent. This may engender important packet losses before a node detects that its preferred parent is out of range or has left the network.

Therefore, our purpose is to increase the route stability when building the DODAG by avoiding as much as possible the choice of a mobile node as a PP node within the DODAG. Then, we propose to consider the node mobility information as a new criterion when selecting the PP node from the parent set in addition to the rank comparison criterion that considers the different path metrics/constraints. Hence, the preferred parent selection procedure becomes:

- If two nodes within the parent set have almost the same rank, then;
    - If two nodes are either fixed or mobile nodes, then the node so far selected as a PP node is

maintained, so as to keep the DODAG stability.
   - If however, one of the nodes is a fixed node and the other node is a mobile node, then the fixed node is always chosen as a PP node.
- If the nodes have different ranks, then :
   - If two nodes are either fixed or mobile nodes, then the node with the lowest rank is chosen as a PP node.
   - If however, one of the nodes is a fixed node and the other node is a mobile node, then the fixed node is chosen as a PP node.

The flowchart in figure 2 summarizes the ME-RPL preferred parent selection procedure described above.

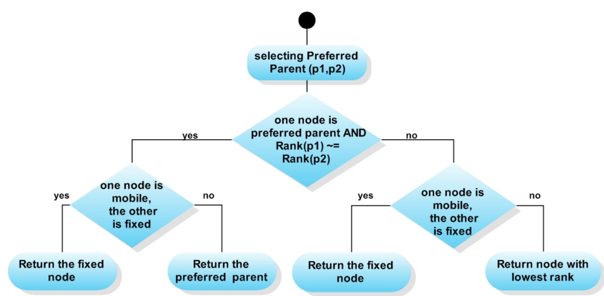


Fig. 2. ME-RPL preferred parent selection procedure

B. Dynamic DIS management procedure

As we have shown in the motivation section, the native RPL is not adapted to react rapidly to different topology changes. For instance, if a node j becomes inconsistent (that is one of its parent set, rank, or PP changed), it reinitializes its Trickle and sends a DIO message upon Trickle expiration. The node j preferred parent may not be affected by this new DIO message broadcast and continue increasing its own Trickle. At that time, the node j may quit the network and realizes too late that it is no more attached to the DAG. At that time, it will send a DIS message after a RPL_DIS_INTERVAL [6] (In the RPL implementation we use [18], the default RPL_DIS_INTERVAL is set to 60 seconds).

Hence, we propose in this paper a new procedure to refresh the DODAG information that adapts dynamically to the network topology changes. Moreover, our approach will allow the prediction of a mobile node behavior in the near future based on its behavior over the past intervals. That is a node that was inconsistent for several intervals has high probability to remain inconsistent in the future (this may be caused by its high mobility or high mobility within its neighborhood). On the contrary, a node experiencing few inconsistencies is likely to keep this stability in the near future.

For instance, as indicated before, node inconsistency can occur due to rank change, parent set change or preferred parent change [8]. In our case:

- A node rank can change due to its neighborhood because of mobile node arrivals.
- In the same way, a node that may be a fixed node can see its parent set changing if some other mobile nodes come to its vicinity and may affect its parent set without having any impact on its selected PP.
- The PP change indicates, however, that either a node failure or mobility occurred in the network.

Since, we do not consider nodes failures, the PP change will be a good indicator for node mobility in the network. In the rest of our study, we consider the PP change as the only inconsistency parameter indicating node mobility within the network.

Then, when a node experiences high mobility (its PP changed during several intervals), it has not to wait for the next DIO period to update its parent set information and has to enforce the graph refresh information by sending a DIO solicitation message. As a result, we propose in our approach to use the DIS messages, not only to allow nodes joining the DAG, but also to force a node experiencing a lot of inconsistency to update its routing information very frequently to be sure being always attached to the DAG.

The inter-DIS interval will be computed by each node based on the number of PP changes within the previous inter-DIS time period. Indeed, if the number of PP changes is important, this implies that the node is in an unstable environment in the current period and is likely to remain in the same environment in the next period. Then, it might be able to refresh its routing information by sending explicit DIO solicitation if needed. For this, the next inter-DIS period should be smaller that the current inter-DIS period. In the same way, if the PP remains unchanged for several inter-DIS periods, then the next inter-DIS period should be increased. When we consider that the inter-DIS time interval should increase, we set:

$$New(Inter\_DIS\_Period) = Old(Inter\_DIS\_Period) \times 2 \quad (1)$$

Similarly, for decreasing the inter-DIS period, we set:

$$New(Inter\_DIS\_Period) = Old(Inter\_DIS\_Period)/2 \quad (2)$$

Our dynamic DIS management procedure will depend on the following parameters:

- **N_Down_DIS:** the number of PP changes above which the inter-DIS interval should be divided by 2. In fact, if we decide to increase the inter-DIS period each PP change, this would introduce fluctuation in control traffic within the network. Hence we decide to decrement the inter-DIS period only after observing several PP changes given by N_Down_DIS.
- **N_Up_DIS:** The number of times the node chooses the same PP node above which the inter-DIS interval should be multiplied by 2.
- **I_DIS_min:** The minimum time interval below which the Inter-DIS period could not be decremented. Indeed, the inter-DIS interval could not be divided indefinitely each N_Down_DIS PP changes within the current period. This will in fact drastically increase the control traffic generated by the DIS messages.
- **I_DIS_max:** The maximum time interval beyond which the inter-DIS period could not be incremented.

This maximum value will be taken as the default RPL_DIS_INTERVAL in the original Contiki RPL [18].

Precisely, the DIS management procedure in ME-RPL is described by the algorithm 1.

Algorithm 1: Dynamic DIS Management

begin
- $dag \leftarrow Current\_DAG\_Reference$
- $next\_dis \leftarrow next\_dis + 1$
- **if** $PP\_changed = 1$ **then**
  - `/* Increase the number of PP changes */`
  - $nb\_parent\_changed \leftarrow nb\_parent\_changed + 1$
- **else**
  - $nb\_same\_parent \leftarrow nb\_same\_parent + 1$
- **if** $PP\_changed = 1$ **then**
  - **if** $period\_dis \geq 2 \times I\_DIS\_min$ **and** $nb\_parent\_changed \geq N\_Down\_DIS$ **then**
    - $DIS\_period \leftarrow DIS\_period/2$
    - $nb\_parent\_changed \leftarrow 0$
  - **if** $period\_dis \leq RPL\_DIS\_INTERVAL/2$ **and** $nb\_same\_parent \geq N\_Up\_DIS$ **then**
    - $DIS\_period \leftarrow DIS\_period \times 2$
    - $nb\_same\_parent \leftarrow 0$

The above procedure shows that each node experiencing a number of PP changes within the current inter DIS period greater than N_Down_DIS, decides to divide its inter_DIS period by 2 if this latter is greater than I_DIS_min. In the same way, each node experiencing no PP changes within N_Up_DIS successive inter DIS periods decides to multiply its inter_DIS period by 2 if this latter is smaller than RPL_DIS_INTERVAL. Another improvement to the ME-RPL protocol could be done by enforcing the refresh of reverse paths, hence reducing the packet loss rate for the DODAG root upcoming traffic. This could be done by sending immediate DAO messages upon PP changes as illustrated in [8], [9].

V. Performance Evaluation

A. Simulation environment: COOJA and Contiki

In this paper, we opt for the COOJA [10] simulator using the RPL implementation of the OS Contiki [18]. In contrast to other simulators, COOJA enables simultaneous simulation at many levels combining low-level simulation of hardware and high-level behavior in a single simulation. It allows nodes instantiation using real code compiled for actual hardware, and may use an operating system such as Contiki that already implements the RPL protocol [10]. The implementation of ME-RPL is then a modification of the RPL-Contiki (also our implementation actually places the "mobility status" in the "reserved" field of DIO messages, the 8th byte, without impact on performance or semantics).

B. Effects of preferred parent selection and dynamic DIS management algorithm

Before, comparing the performance of the ME-RPL solution with the native RPL protocol, we first illustrate the functioning and verify the correct behavior of each procedure proposed by ME-RPL and implemented under COOJA/Contiki.

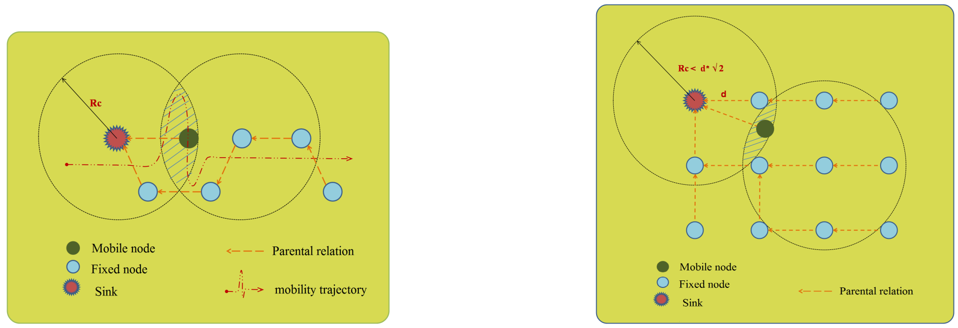
1) Effect of the preferred parent selection procedure: To verify the impact of the preferred parent selection procedure, we consider the two particular topologies: linear and grid topologies. In the linear topology given by figure 3(a), each node has just one possibility to join the DODAG among fixed nodes: nodes are placed so as to offer a unique fixed neighbor as potential preferred parent. One mobile node moves in a trajectory offering temporarily the shortest path to some nodes. In figure 3(b), however, nodes have more than one choice of fixed (non-mobile) parent to join the RPL DODAG. In addition, choosing the mobile node may offer better paths for some nodes, at least temporarily. For both topologies, we evaluate for each node, the number of times, during the whole simulation, that it selected a mobile node as a preferred parent. Figures 4(a) and 4(b) show that nodes have high tendency to choose the fixed nodes as preferred parents rather than mobile ones after the modifications introduced in the preferred parent selection procedure, as is the goal of ME-RPL. Notice that statistics are for each call of the preferred parent selection algorithm, which gets invoked several times before a node can actually select a fixed node as parent.

2) Effect of the Dynamic DIS management procedure: In this section, we evaluate the impact of the minimum inter DIS interval values on the packet delivery ratio. In the first case, we consider the variable N_Down_DIS as parameter. The figure 5(a) depicts the variation of packet delivery ratio with different minimum inter DIS values. In the second case, we vary the parameter N_Up_DIS to pinpoint the effect of these parameters on the data delivery ratio. Figure 5(b) depicts the variation of the same metric with the minimum inter DIS and for different N_Up_DIS values. If we analyze the two charts, we can conclude that for small values of minimum inter DIS, the values of N_Down_DIS =1, N_Up_DIS = 5 and I_DIS_min = 3 sec is a good trade-off to have a maximum packet delivery ratio.

C. Comparison of RPL and ME-RPL

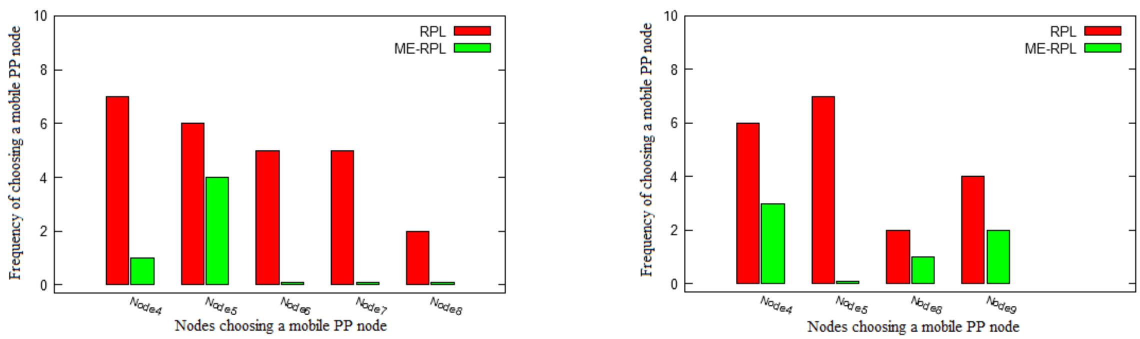
In this section, we compare the performance of both RPL and ME-RPL routing protocols in terms of packet loss rate and route stability.

1) Packet loss rate: One of the most important performance parameters for the comparison of RPL with ME-RPL is the packet loss rate aggregated over the whole network. Figures 6(a) and 6(b) depict the packet loss rate of RPL and ME-RPL as a function of the total number of nodes with a number of mobile nodes of 6 and 9 respectively. The curves in figures 6(a) and 6(b) show that the packet loss rate is always lower with ME-RPL than with the native RPL, no matter is the number of total nodes and mobile nodes present in the network. To further



(a) ***Linear topology*** **(b)** ***Grid topology***

Fig. 3. Test topologies to illustrate the preferred parent selection behavior



(a) ***Linear topology*** **(b)** ***Grid topology***

Fig. 4. Frequency of choosing a mobile node as a preferred parent with RPL and ME-RPL

illustrate this aspect, we evaluate in the next paragraph, our route stability metric for both RPL and ME-RPL.

D. Route stability

To evaluate the route stability, we derive the percentage of time during which routes are valid and can be effectively used to route data information. Hence, the route stability is computed as:

$$S(\%) = (x_t - x_p)\mu_p / N\mu_s \times 100 \qquad (3)$$

where N is the number of nodes within the network, μ_p the packet inter-arrival period, μ_s the whole simulation time and x_t (respectively x_p) gives the total number of sent (respectively lost) packets in the network. Figure 7 shows that the percentage of stable routes is always higher with ME-RPL than with RPL, no matter the number of sensor nodes present in the network. This can be explained by the fact that ME-RPL favor the choice of fixed nodes in the route establishment which indeed increases the route stability.

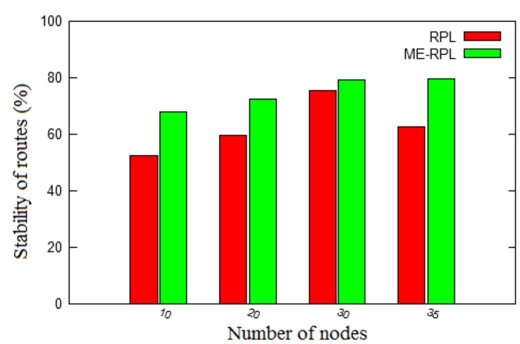
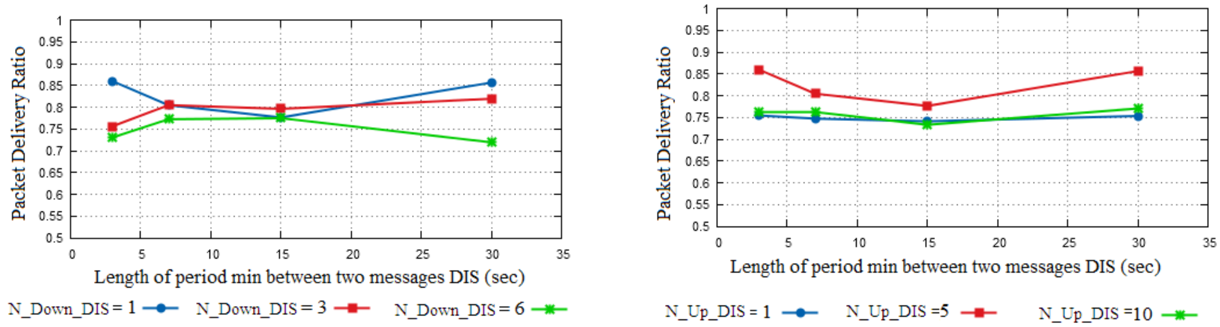


Fig. 7. Route stability with RPL and ME-RPL

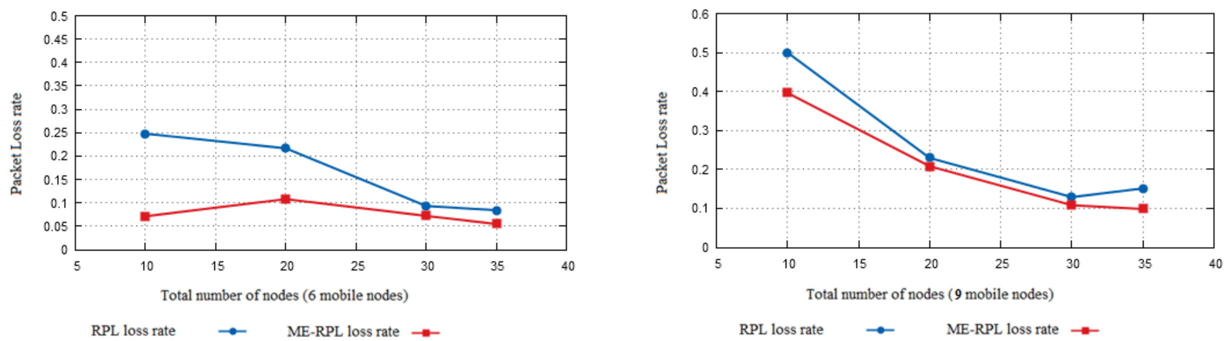
VI. Conclusions and Future Work

In this paper, we proposed extensions to support mobility in the RPL protocol when used in wireless sensor networks. We presented some background concepts related to the existing routing solutions in WSNs and RPL basics. We then presented the current work motivations and described the different procedures that we introduced to support mobility over RPL. These procedures include the modification of DIO messages to exchange the mobility status of the node, the modification of the preferred parent selection procedure to favor the choice of fixed nodes as preferred parents and a dynamic DIS management procedure to allow a quick update of the DODAG information. All the simulation results show that our proposed approach ME-RPL outperforms RPL in terms of packet delivery ratio and routes stability within the



(a) ***Impact of N_Down_DIS*** **(b)** ***Impact of N_Up_DIS***

Fig. 5. Packet delivery ratio as function of the minimum inter DIS period



(a) ***6 mobile nodes*** **(b)** ***9 mobile nodes***

Fig. 6. Packet loss rate with RPL and ME-RPL

network. In our future work, we intend to design an automated mobility detection algorithm for RPL and to study RPL rank calculation extensions.

REFERENCES

[1] W. Heinzelman, J. Kulik and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," In proc. of the 5th ACM/IEEE Mobicom Conference (MobiCom '99), Seattle, WA, August, 1999. pp. 174-85.

[2] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion:a scalable and robust communication paradigm for sensor networks", In proc. of the ACM MobiCom '00, Boston, MA, 2000, pp. 56-67.Y.

[3] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks", in ACM SIGMOD Record, Volume 31 Issue 3, September 2002.

[4] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", In proc. of the 33rd Hawaii International Conference on System Sciences (HICSS '00), January 2000.

[5] S. Lindsey, C. Raghavendra, "PEGASIS: Power-Efficient Gathering in Sensor Information Systems", In proc. of the IEEE Aerospace Conference, 2002, Vol. 3, pp. 1125-1130.

[6] T. Winter, P. Thubert, A. Brandt, T. Clausen, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik and J.Vasseur, "RPL : IPv6 Routing Protocol for Low power and Lossy Networks", RFC 6550, IETF ROLL WG, March 2012.

[7] J.P. Vasseur, "Terminology in Low power And Lossy Networks", draft-ietf-roll-terminology-06.txt, September 2011.

[8] K. C. Lee, R. Sudhaakar, J. Ning, L. Dai, S. Addepalli, J. P. Vasseur and M. Gerla, "A Comprehensive Evaluation of RPL under Mobility", Hindawi Publishing Corporation International Journal of Vehicular Technology Volume 2012.

[9] K.C. Lee, R. Sudhaakar, L. Dai, S. Addepalli, and M. Gerla, "RPL under mobility", In proc. of 2012 IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, January, 2012.

[10] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne and T. Voigt, "Cross-Level Sensor Network Simulation with COOJA", In proc. of the 31st IEEE Conference on Local Computer Networks, LCN'06, Florida, USA, November 2006.

[11] C. Schurgers and M.B. Srivastava, "Energy efficient routing in wireless sensor networks", In proc. of the 2001 IEEE Military Communications Conference, MILCOM'01, Washington, USA, October 20012001.

[12] Z. Shelby and C. Bormann, "6LoWPAN: The Wireless Embedded Internet", John Wiley & Sons, year 2009.

[13] IEEE Standard for Information technology 802.15.4 - 2006, "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)", 2006.

[14] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle Algorithm", RFC 6206, (Proposed Standard), March 2011.

[15] M. Goyal, E. Baccelli, M. Philipp, A. Brandt and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks", draft-ietf-roll-p2p-rpl-13, November 2011.

[16] S.A. Munir, B. Ren, W. Jiao, B. Wang, D. Xie and J. Ma, "Mobile wireless sensor network: Architecture and enabling technologies for ubiquitous computing", In proc. of the 21st International Conference on Advanced Information Networking and Applications Workshops, AINAW (2007).

[17] P. Thubert, "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552 (Proposed Standard), March 2012.

[18] A. Dunkels, B. Gronvall and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors", In proc. of the First IEEE Workshop on Embedded Networked Sensors, Florida, USA, November 2004.