Integrating Mobility in RPL

Cosmin Cobârzan, Julien Montavont, and Thomas Noel

ICube laboratory (CNRS), University of Strasbourg, France
{cobarzan,montavont,noel}@unistra.fr

Abstract. In the last years the Low Power and Lossy Networks (LLNs), have become more and more popular. LLNs are inherently dynamic - nodes move, associate, disassociate or experience link perturbations. In order to meet the specific requirements for LLNs, the IETF has developed a new routing protocol - IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) that routes packets inside LLNs. RPL has to work in such dynamic environment and mechanisms that can mitigate such conditions are suggested in the standard such as Neighbor Unreachability Detection or Bidirectional Forwarding Detection. In this article, we show that such mechanisms fail to prevent serious node disconnection, which significantly increases the packet loss and leads to severe underachievements. To provide RPL the ability to mitigate network dynamics generated by node disconnection, we therefore propose a new cross-layer protocol operating at layers 2 and 3 known as Mobility-Triggered RPL (MT-RPL). MT-RPL benefits from the X-Machiavel MAC protocol that favors medium access to mobile devices. X-Machiavel has been extended to trigger RPL operations in order to maintain efficient connectivity with the network. MT-RPL is evaluated together with Neighbor Unreachability Detection and Bidirectional Forwarding Detection through an extensive simulation campaign. Results show that MT-RPL significantly reduces the disconnection time, which increases the packet delivery ratio and reduces energy consumption per data packet.

Keywords: Sensor networks, RPL, Network dynamics, Mobility.

1 Introduction

Smart objects, whether they are smart watches or intelligent home appliances, are surrounding us every day. What's "smart" is a sensor, which forms a network between it and others. Sensors can communicate wirelessly and form a class of networks called Low-Power and Lossy Networks (LLNs), where the routers and the devices they interconnect are constrained in terms of processing power, battery and communication range [5]. Interconnections between sensors are characterized by high loss rate, low data rates and instability [18].

Routing packets in LLN is done with a new protocol proposed by the IETF known as IPv6 routing protocol for Low-Power and Lossy Networks - RPL [18]. This protocol builds a Destination Orientated Directed Acyclic Graph (DODAG),

T. Abdelzaher et al. (Eds.): EWSN 2015, LNCS 8965, pp. 135–150, 2015.

which is shaped by a set of metrics/constraints. When a node connects to the graph, it chooses a parent (which will forward information to the root) and computes a rank (estimation of position in the graph). However, nodes can lose connectivity from the parent due to node actions (movement, association, disassociation or disappearance) or link perturbations (fading, shadowing or path loss). Such network dynamics have an impact on (re)organization, (re)configuration and routing protocol convergence that is likely to endanger network operations. RPL has been designed to cope with network dynamics and maintain network connectivity using external unreachability detection mechanisms.

There are three suggested unreachability detection mechanisms that help RPL to detect and repair communication problems: Neighbor Unreachability Detection (NUD) [17], Bidirectional Forwarding Detection [2] and hints from lower layers via Layer 2 (L2) triggers such as [6]. Those mechanisms act on different layers according to the needs of the application. In this article, we present a performance analysis of those three methods. To the best of our knowledge, they have not yet been evaluated side by side. Results presented in Sect. 5 show that those mechanisms fail to mitigate node mobility that make the network dynamic. As a result, nodes experience long disconnection time, increasing both packet loss and energy consumption. We therefore propose a new cross-layer protocol referred to as Mobility-Triggered RPL (MT-RPL). MT-RPL is a specific implementation of the generic L2 triggers with X-Machiavel [16] preamble sampling MAC protocol. X-Machiavel is part of our previous work and grants better access to transmission resources to mobile nodes. MT-RPL is further detailed in Sect. 4. The performance evaluation shows that MT-RPL shortens disconnection time and improves energy consumption and network usage. The main conclusion drawn from the work presented in this article is that LLNs require moving forward the layered protocol stack to achieve the best performance.

The rest of the paper is organized as follows. First, we present how RPL mitigates mobile nodes that make the network dynamic, without external unreachability detection mechanisms. Section 3 presents an overview of the mechanisms suggested by RPL to manage unreachability detection. Our proposal MT-RPL is described in Sect. 4. The simulation parameters and results of the performance evaluation are detailed in Sect. 5. The related work presented in Sect. 6 analyzes network dynamics in RPL. Finally, we give some concluding remarks along with future investigations in Sect. 7.

2 Problem Statement

RPL has been developed to enable IPv6 routing inside a LLN. It builds a Destination Orientated Directed Acyclic Graph (DODAG) toward the root, shaped by an objective function. The topology is built using new ICMPv6 messages: DODAG Information Object (DIO), DODAG Information Solicitation (DIS) and DODAG Destination Advertisement Object (DAO). The border router between the Internet and the LLN acts as the root for the graph. It starts building the graph by sending the first DIO. Nodes that receive DIO will build a parent

set (potential next hops toward the root) and select their preferred parent. The preferred parent is a member of the parent set that is the preferred next hop toward the root. Such selection is based on the rank advertised in DIO. Once a preferred parent is chosen, nodes are considered attached to the graph and will advertise DIO further. Nodes that are not connected can either wait for a DIO or send a DIS requesting information about existing DODAG. Nodes in the neighborhood transmit a DIO in response to a DIS. Finally DAO advertises destination information upward to the root, enabling point-to-point and point-to-multipoint communication. Nodes in a RPL network use these messages when they connect to the DODAG as well as each time when, after a disconnection, communication needs to be resumed.

Network dynamics is an integral part of LLNs as the links are lossy and nodes have limited transmission and energy capabilities. Adding mobility in such scenarios enables building of new applications that are impossible to have with static nodes, such as target tracking or surveillance applications ([8], [9], [15]). This in turn makes communicating in this environment more challenging: in addition to link perturbations, ongoing communications can suffer from either node movement or disappearance, leading to network partitions as parents in the graph might be no longer reachable. RPL mitigates such problems by allowing nodes to reconnect to the graph by changing their preferred parent. Such operation occurs when a node receives a DIO, advertising a better rank than the one of the preferred parent. However, there is a situation where the preferred parent of a node is no longer reachable (due to mobility, failure, etc.) and all received DIO advertise a higher (worse) rank. In this situation, the node is disconnected from the graph because its preferred parent remains the best candidate in the parent set. Such disconnection is likely to increase packet loss, contention on the medium and energy consumption. This scenario is illustrated in Fig. 1.

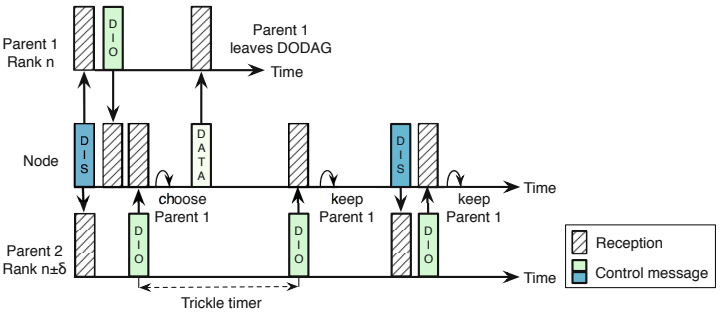


Fig. 1. RPL parent management

Furthermore, RPL does not specify how to manage the parent set, especially when and for what reason a node should be removed from a parent set. Nevertheless, RPL suggests the use of external mechanisms for unreachability detection such as Neighbor Unreachability Detection (NUD) [17], Bidirectional Forwarding Detection [2] and hints from lower layers via Layer 2 (L2) triggers [6]. When one of these mechanisms indicates that the preferred parent is unreachable, the

node will search for a new parent. First it will search in the parent set and, if no parent is available, through a local repair. Local repair means announcing infinite rank in a DIO (disconnecting from the DODAG), removing all parents from the parent set (to be able to accept parents regardless of their rank) and sending DIS periodically until new DIO are received. RPL, together with one of these mechanisms should enable continuous communication on transient and lossy links. However, to the best of our knowledge, those methods have not yet been evaluated side by side in RPL. In the next section, we present how all three mechanisms signal node unreachability.

3 Unreachability Detection in RPL

3.1 Neighbor Unreachability Detection

Neighbor Unreachability Detection (NUD) is part of Neighbor Discovery for IP version 6 (IPv6) [17]. It tracks all paths between active neighboring nodes and specifies when a neighbor is unreachable. The state of connectivity between neighbors is stored locally on each node in a structure called neighbor cache. When a path to a neighbor appears to be failing, NUD signals the need for a new next hop, by deleting the neighbor cache entry. At RPL layer, this will trigger the node to remove the parent and start searching for a new one, either in the parent set (if it is not empty) or through a local repair.

NUD enables neighbors to exchange Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages to confirm reachability. Each neighbor has an entry in the neighbor cache for all connections it has with other nodes in the same network. Cached values for nodes can be: REACHABLE - communication is granted between nodes, STALE - the neighbor is no longer known to be reachable but no action is taken until traffic is sent to this neighbor, DELAY - optimized state that delays sending probe for *DELAY_FIRST_PROBE_TIME* seconds (node waits for reachability confirmation from upper layers) and PROBE - NS are sent until reachability is confirmed or the maximum allowed number of probes (*MAX_UNICAST_SOLICIT*) are sent. Timers, which are illustrated in Fig. 2, manage the exchange of control messages and trigger the removal of the cache entry. Default values for timers give a 30 sec reachable time window.

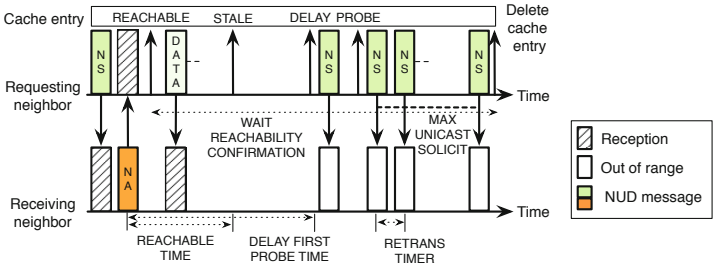


Fig. 2. NUD message exchange

After this time elapses, the first probe is sent with a default 5 sec delay (*DELAY_FIRST_PROBE_TIME*) and than each second until *MAX_UNICAST_SOLICIT* probes are sent (3 retransmissions by default). In the worst case and considering the timer default values, it takes 38 sec for NUD to detect the unreachability of a neighbor. We doubt that such delay is short enough to allow RPL nodes to change parent seamlessly and without experiencing packet loss.

3.2 Bidirectional Forwarding Detection

Bidirectional Forwarding Detection [2] is a simple Hello protocol that detects failures in communication with a forwarding plane next hop. A pair of nodes exchanges BFD messages encapsulated in UDP packets to maintain reachability information. The path between two nodes is declared operational when two-way communication can be established. When no messages are received for long enough, BFD considers that the neighboring system has failed. At RPL layer, this will trigger the node to remove the parent and start searching for a new one, either in the parent set (if it is not empty) or through a local repair. RPL is paired with BFD asynchronous mode. In this mode, messages are sent periodically between systems. If a number of packets in a row are not received, the session is declared down, as connectivity is lost. Operation of BFD, along with the state of each system, is presented in Fig. 3. BFD packets are sent by default every 20 sec and if one packet is lost, the systems declare the state *DOWN*.

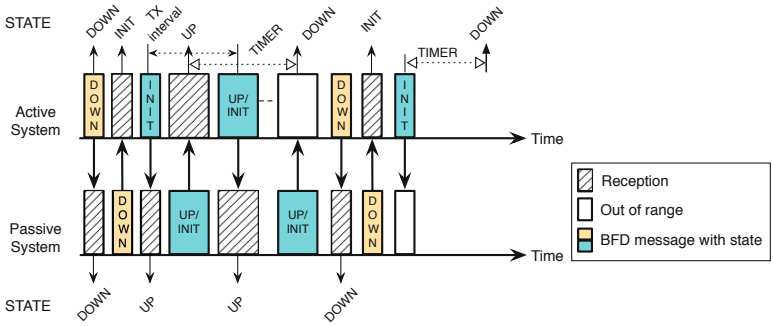


Fig. 3. BFD message exchange in asynchronous mode

3.3 Hints from Lower Layers via Layer 2 (L2) Triggers

Although specific to each MAC layer, the hints from lower layers via Layer 2 (L2) triggers [6] share common structure in the form of L2 Abstractions. Services between layers are provided in the form of primitives, which enable synchronous communication between layers. Two pairs of primitives are defined to be used when events occur: Request/Confirm and Indication/Response.

Primitives can be used in 3 different cases based on their types - *Type 1*: Provide information to upper layers; information is provided immediately to upper layer through a request-confirm message exchange; *Type 2*: Notify upper

layer of L2 events asynchronously, indicating each occurrence of registered events to upper layers; *Type 3*: Control L2 actions from upper layers; Request primitive is used to interact with lower layer which will reply with Ack or Nack in a Confirmation primitive.

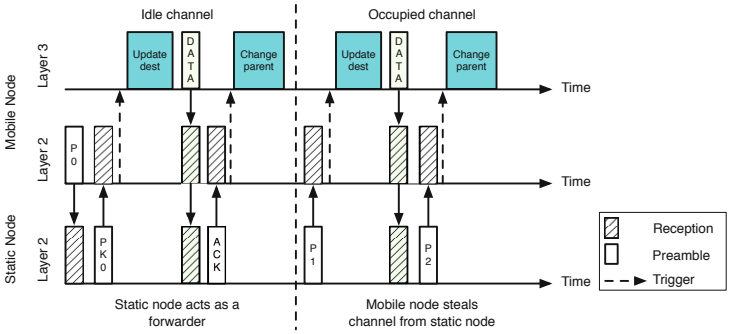
We are convinced that this solution will provide the best results. There are some MAC protocols like 802.14.5 in beacon mode that keep track of nodes associated to a PAN coordinator and detecting disconnection is already implemented in the protocol, but this is unavailable in most LLN MAC protocols. Using L2 triggers with any MAC layer allows events to be faster delivered to upper layer protocols. This is why we propose MT-RPL, a solution to communicate between MAC layer and RPL using *Type 2* primitives. In the next section, MT-RPL will be presented in more detail.

4 Mobility-Triggered RPL

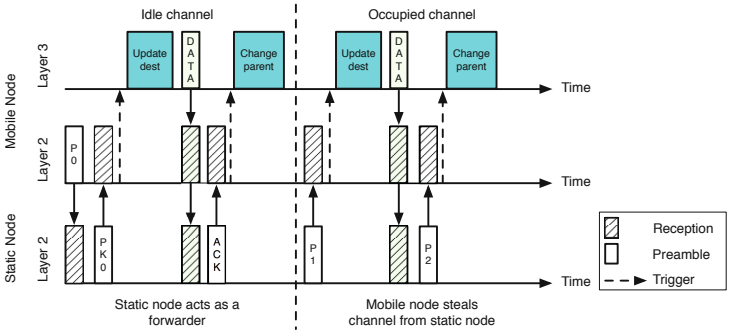
The mechanisms presented in the previous section alongside local repair should manage mobile nodes that generate dynamics in the network. However, we are convinced that they are not adapted to LLN specifics, especially considering the number of exchanged messages (BFD) or the suggested timer values (NUD). Only L2 triggers seems to cope with LLN constraints but this is a generic solution that should be adapted regarding the MAC protocol in operation. In this section, we propose a new cross layer protocol that manages network dynamics in LLN. Mobility-Triggered RPL (MT-RPL) is a specific implementation of L2 triggers linking RPL and X-Machiavel [16] preamble sampling MAC protocol.

X-Machiavel is a variation of the well known X-MAC [12] preamble sampling MAC protocol. With X-MAC, a node starts to send preamble strobes in order to synchronize the destination for the pending transmission. Once the destination receives a strobe, it sends back an ACK to notify the sender to stop the preamble and proceed with the data. Upon data reception, the destination sends a new ACK to the sender. X-Machiavel slightly modifies this behavior to favor mobile node transmissions. X-Machiavel assumes that the network is composed of static and mobile nodes. On an idle channel, packets sent by mobile nodes can be opportunistically forwarded by static nodes. On a busy channel, mobile nodes can steal the medium of an ongoing transmission to send their packets first. For this, X-Machiavel introduces two new fields in the packet header. The type field defines whether the packet is a preamble frame (P0, P1 or P2), a data packet (DATA), an acknowledgement for a preamble (PK0 or PK1) or an acknowledgement for a data packet (ACK). P0 preamble strobes are used by mobile nodes to forbid channel stealing and allow opportunistic nodes to accept the pending data on behalf of the destination. P1 preamble strobes are used by static nodes and enable mobile nodes to steal the channel. Finally, P2 preamble strobes are also used by static nodes to forbid channel stealing. Preamble strobes are acknowledged with type PK0 acknowledgement sent by static nodes to acknowledge a P0 preamble that was not initially intended for them. This informs the mobile node that its data can be handled by another static node

acting as an opportunistic forwarder. PK1 acknowledgement is sent in the other cases when nodes acknowledge preambles destined to them. In the flags field, a mobile node sets a M flag (on most significant bit - MSB) for data packets that is used to prioritize transmissions from mobile nodes. Fixed nodes that receive data with the M flag set forward it by using a P2 preamble so that other nodes cannot steal the medium and impair the transmission originating from the mobile node. For more information about how X-Machiavel works the reader can consult [16]. In the following, we present how X-Machiavel interacts with RPL to form MT-RPL.



(a) Preamble is acknowledged or overheard



(b) Preamble is not acknowledged

Fig. 4. MT-RPL

X-Machiavel prioritizes the transmission from mobile nodes, elements that generate great dynamics in the network. To take advantage of this in MT-RPL, RPL registers a L2 trigger to be informed asynchronously every time the mechanism of X-Machiavel is triggered (e.g. channel stealing or using an opportunistic forwarder). For this, MT-RPL includes the rank computed at RPL layer in the layer 2 header. By this means, nodes can decide in a distributed way whenever it is worthwhile to act as an opportunistic forwarder or to steal the medium from an ongoing communication. MT-RPL operational modes are presented in the following. If the preamble is acknowledged (Fig. 4a) on an idle channel, a node sends a P0 type preamble including its rank computed at the RPL layer.

If the destination is in the neighborhood, X-MAC principles apply: the destination sends a PK1 acknowledgement and claims the data from the mobile node. On the other hand, if the destination is not in the neighborhood and another static node receives the P0 preamble, it can decide to act as an opportunistic forwarder for the pending data. This decision is based on the RPL rank announced in the preamble. If the rank of the sender is greater than the one of the potential forwarder (i.e. the sender is located further in the graph than the potential forwarder), the potential forwarder can send back a PK0 acknowledgement. Upon reception, the mobile node changes the destination to the forwarder and sends the data. This data may now be routed to the root using P2 preambles so that no other mobile nodes can steal the channel. Forwarders with a rank equal or greater that the one of the sender simply discard the overheard preamble.

Transmitting data on an occupied channel requires the mobile node to seize the opportunity to transmit its data between strobed preamble frames that are destined to another node. X-MAC principles require that the destination of preamble strobes send back an ACK between two strobes to notify the sender to stop the preamble and proceed with the data. MT-RPL allows mobile nodes to send their own data before such ACK from the original destination. However, MT-RPL enables this behavior only if the rank of the sender of the preamble is lower than the rank of the mobile node, i.e. the mobile node's data will progress forward toward the root of the graph. As a result, channel stealing operates as follows. First, a mobile node should overheard a P1 preamble destined to another node and announcing a RPL rank lower than its own RPL rank. Then, the mobile node changes the destination of its data to this sender and transmits the resulting packet between two preamble strobes. After receiving such packet, the forwarder still needs to send its own data and does that by using P2 preambles. Further along nodes operate as in X-MAC. Regardless of how the static nodes received data from mobile nodes, they will forward it using P2 preambles until the final destination is reached. If the preamble sent by a mobile node is not acknowledged (Fig. 4b), the mobile node is in an area where all surrounding nodes have a rank higher than its own, so the mobile node will change its rank to infinite. At the next retransmission, any neighbor can acknowledge the preamble and the mobile nodes data packet will be forwarded to the root using P2 preamble.

MT-RPL manages the parent set regarding the information received from layer 2 through L2 triggers. When the mobile nodes benefits from an opportunistic forwarder (by receiving a PK0 acknowledgment) or steals the medium from another node (sending a data between two preamble strobes from an ongoing communication), if the transmission is successful, the layer 2 provides the rank and the address of the effective next hop to the RPL layer. Upon reception, RPL set this node as the new preferred parent, computes the related rank and proceed with RPL operations whenever necessary (send new DAO and/or DIO). As a result, MT-RPL should smooth network dynamics by enabling nodes to promptly react to network change without generating extra control traffic.

5 Simulation Setup and Results

5.1 Simulation Scenario

In order to evaluate the mitigation of network dynamics by RPL, we used the WSNet software [4]. WSNet is a discrete event simulator dedicated to the study of wireless sensor networks. WSNet already provides a basic RPL module that we extended to operate as presented in both Sect. 3 and 4.

Table 1. Simulation parameters

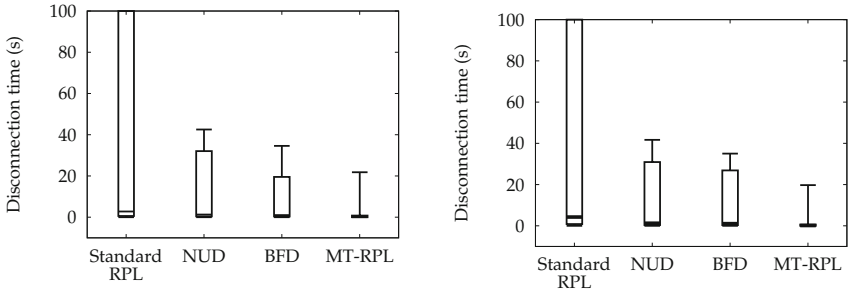
Simulation parameter	**Value**
Topology	
Random topology	1 root, 60 static nodes, 5 mobile nodes
Grid topology	1 root, 36 static nodes, 5 mobile nodes
Data collection scheme	Time driven 1 packet/30s static nodes → root 1 packet/5s mobile nodes → root and root → mobile nodes
Data packet size	127 bytes
Mobility model	Billiard, 1m/s random trajectory
Routing model	RPL in non-storing mode using MinHop
RPL default values	DIO - given by trickle timer algorithm [14] DIS - 2s if empty parent set, until attached to DODAG DAO - 60s from every node, or when needed
Values for parameters of unreachability detection mechanisms	
NUD (RFC 4861)	Maximum number of NS transmission - 3, Delay first probe - 5s, Reachable time - 30s, Retransmission time - 1s
BFD (RFC 5880)	Desired TX interval - 30s, Missed BFD packets that bring session DOWN - 1
MAC model	X-MAC (for standard RPL, NUD and BFD) and X-Machiavel (for MT-RPL) Maximum number of retransmissions - 4
Radio model	Half-duplex, Channel 0, Sensibility level: -92dBm, 15 kB/s bandwidth, 18m (60 feet) [10] unit disk range
Current consumption	TX: 31 mA, RX: 15.1 mA OFF: 400 nA (CC1100 chip)
Antenna model	Omnidirectional, modulation BPSK
Simulation setup	20 simulations/mechanism/topology, 4 mechanisms, 2 topologies, 1 hour/simulation

All simulation parameters are presented in Table 1. We deployed a random topology of 60 nodes on a 100x100 m area with the root in the middle and a grid topology with 36 nodes and the root in the middle. To generate dynamism, 5 mobile nodes are distributed and move following a simplified version of random direction model, used also in [19]. Such nodes are pre-configured with the status of mobile node, as they have physical capabilities to move (e.g. node is on a platform with wheels). Standard RPL, NUD and BFD are coupled with X-MAC because X-Machiavel favors transmissions of data packets from mobile nodes, but the node which acknowledges the preamble, or from which the channel is stolen, may not be the parent at RPL layer and the packet even though it is

sent, it is dropped by the receiving node. MT-RPL as it receives information from X-Machiavel takes advantage of this changes and adjusts accordingly the transmission of data packets. Only links between the mobile node and its parent are monitored using BFD, NUD or MT-RPL. On the rest of the path until the root, the packet is routed using standard RPL, as these links are not subject to network dynamics generated by the mobile node. With all methods, mobile nodes keep only the preferred parent in the parent list, which may change when DIO with a better rank is received or if the mobile node does a local repair. The path from the root to the mobile node is maintained up to date with DAO messages. Changes in topology are reported to the root in a timely fashion. Packets are delivered following source routing set by the root. In the analyzed scenarios, both mobile and static nodes send control messages as needed in order to maintain connectivity to the DODAG. The DODAG that RPL build needs a long time to stabilize [13]. Therefore, we started analyzing results only after 30 min from the start of the simulation, when the DODAG will be in a stable state and the mobile nodes start moving. After this time, the structure of the DODAG in the static part of the network will not change, in order to analyze only the changes induced by mobile nodes in the network. At the end of the simulation, packets were not sent for 15 min, so that all queues of packets from all nodes could be emptied.

5.2 Results

The results presented in this section were obtained after running 20 simulations of each scenario for each configuration for a total of 200 simulations. The presented results are the average of overall data collected from each set of simulations. The 95% confidence interval indicates the reliability of our measurements. We analyzed four parameters: disconnection time from the preferred parent, packet delivery ratio (PDR), overall number of control messages sent in the network and energy consumption.



(a) Grid topology (b) Random topology

Fig. 5. Average disconnection time from parent

Figure 5 illustrates the disconnection time for each scenario, i.e. the time between a mobile node going out of the radio range of its preferred parent and

enforcing a new preferred parent at the RPL layer. As we can see, standard RPL shows the longest disconnection time (up to 700 sec. in the worst cases) as changing the preferred parent is only done by receiving a new DIO with a better rank. Therefore, it is likely that a mobile node remains disconnected for a long period because all received DIO present a higher (worse) rank. An unreachability detection mechanism is therefore mandatory in order to avoid such situation that could lead to severe underachievements. By contrast, the disconnection time is drastically reduced using RPL coupled with NUD or BFD. In those cases, when a mobile node does not receive reachability confirmation from its preferred parent, RPL removes the preferred parent, reset the rank to infinite and starts sending DIS. BFD lowers the maximum disconnection time because it reacts quicker than NUD thanks to its slightly lower reachable time (30s versus 38s for NUD). Variations occur, as mobile nodes need sometimes to send several DIS messages before they can reconnect to the DODAG. Finally, MT-RPL presents the lowest disconnection times. Thanks to the interaction between the layers 2 and 3, a mobile node always regains connectivity when an opportunistic node acknowledges its preamble and successfully receives the effective data. In addition, a mobile node regains connectivity whenever it successfully steals the medium from a neighbor node with a better rank. In those situations, the disconnection time is bound to the sending frequency of data packets and the number of preamble strobes sent before stealing the medium or opportunistic node acknowledgment. This explains the low disconnection time observed for MT-RPL in Fig. 5. However, a mobile node may be in a situation in which it cannot steal the medium or opportunistic node cannot acknowledge its preamble strobes. Such situation occurs when the mobile node moves in an area where the rank of all neighbors is lower (worse) than the rank of the mobile node. Nevertheless, MT-RPL allows a mobile node to reset its rank and remove its preferred parent after sending a whole preamble without receiving an acknowledgment, either from its preferred parent or from an opportunistic forwarder (as in Fig. 4b). As a result, in an unfavorable environment, the disconnection time is only increased by the transmission duration of a whole layer 2 preamble.

Lowering the disconnection time should increase the packet delivery ratio (PDR) on the paths from mobile nodes to the root and from root to mobile nodes. Note that we implemented the solutions so that mobile nodes only try to send data packets if a preferred parent is set. As a result, all solutions do not necessarily send the same number of data packets. Table 2 present the PDR together with the number of data packet sent by each solution in the both scenarios. Standard RPL, as it cannot ensure continuous connectivity of mobile nodes to their parents, has the lowest PDR from mobile nodes to the root. In addition, this scheme sent the largest number of data packets because mobile nodes have no means to remove an out of range preferred parent. Therefore, they keep trying to send data packets while their preferred parents are no longer reachable, increasing the packet loss together with the medium contention due to retransmissions. Results for the path from the root to mobile nodes are not meaningful because only few packets are actually sent. Most of the time, the

Table 2. Nb. of sent data packets and PDR with 95% confidence intervals

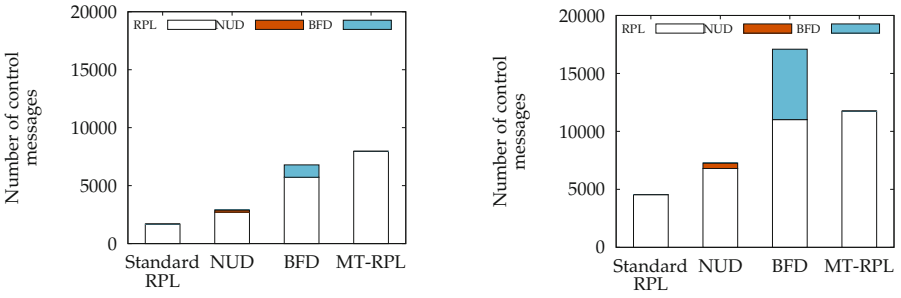
Grid topology			**Standard RPL**	**NUD**	**BFD**	**MT-RPL**
Mobile node to root	Packet delivery ratio	Avg. (%)	8.42	10.06	18.02	62.08
		± (%)	2.42	6.64	4.47	13.99
	Data packets sent	Avg.	666	184.61	501.84	410.15
		±	168.87	68.75	126.01	129.71
Root to mobile nodes	Packet delivery ratio	Avg. (%)	14.58	8.21	13.96	23.21
		± (%)	10.44	7.43	7.97	7.56
	Data packets sent	Avg.	23.95	22.46	47.42	64.60
		±	11.80	16.89	17.58	20.11
Random topology						
Mobile node to root	Packet delivery ratio	Avg. (%)	9.32	12.99	18.93	66.56
		± (%)	1.40	4.00	2.94	4.69
	Data packets sent	Avg.	895.36	210.87	482.78	757.17
		±	23.76	67.72	43.74	46.67
Root to mobile nodes	Packet delivery ratio	Avg. (%)	33.59	37.01	36.53	36.14
		± (%)	15.34	22.20	14.83	12.03
	Data packets sent	Avg.	34.00	22.25	40.57	81.17
		±	13.89	9.75	13.12	18.12

root has no route to mobile nodes (DAO cannot be sent from mobile nodes when they are disconnected) and therefore buffers the packets. When an unreachability mechanism is present at the mobile nodes, values of PDR improve. Thanks to BFD or NUD, mobile nodes change their preferred parents more often, resulting in longer connections to the graph. This allows mobile nodes to send more data packets that successfully arrive at the root. However, values of PDR are still low, as the disconnection from the preferred parent may be reported after long period of time (up to 30s for BFD and 38s for NUD). During this time, preferred parents are still considered as reachable, but all transmitted data packets are lost.

By contrast, lower disconnection times for MT-RPL seen in Fig. 5 are translated into the highest PDR for both mobile nodes and the root. Channel stealing and opportunistic forwarding allow mobile nodes to connect to a parent with a better rank whenever possible. Such reconnection occurs without triggering a local repair, reducing the disconnection time together with the signaling overhead as neighbor nodes can keep a low transmission rate of DIO. However, data packets are still lost with MT-RPL as congestion can form on the path towards the root. The same observation is achieved on the path from the root to the mobile nodes.

Fig. 6 presents the signaling overhead of each solution. The low number of control packets sent in standard RPL further supports the assumption that the

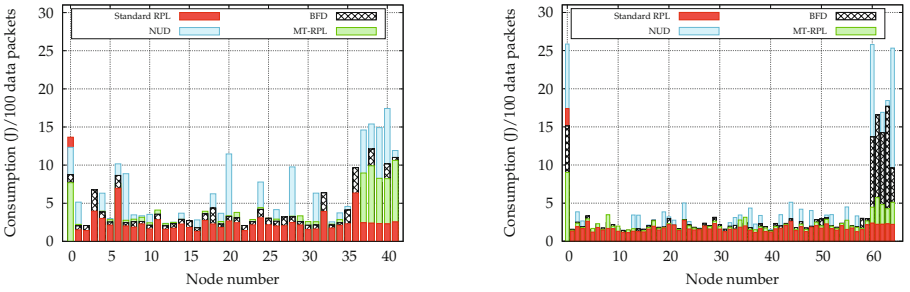
mobility of node are rarely reported with this solution. Furthermore, discovering and attaching to a new parent is done only with RPL control messages, which occur rarely. Adding unreachability mechanisms increases the signaling overhead in the network. Although BFD shows lower disconnection time and higher PDR than NUD, such results come with the expense of higher signaling overhead. BFD maintains sessions both ways between the mobile nodes and their parents by exchanging UDP packets every 30s (each entity manages its own timer). This explain the increased number of BFD control messages in both topologies. NUD on the other hand, relays more on messages sent by the mobile node, which has to check periodically (every 38s) the connectivity to its parent. Furthermore, both NUD and BFD trigger a local repair when the unreachability of the preferred parent is confirmed. Such procedure reset the trickle timer of all neighbor nodes. After a local repair, DIO are therefore sent at a high rate, increasing the signaling overhead reported at the RPL layer. By contrast, MT-RPL does not introduce new control messages. In addition, parent change is achieved without triggering local repair, thus reducing the overall signaling overhead. However, MT-RPL increases the number of reconnections, and therefore makes the use of a large number of DAO to report each parent change.



(a) Grid topology (b) Random topology

Fig. 6. Average number of control messages sent

Energy consumption being one of the crucial point in LLN, we also evaluated the energy depletion of each node in the network. Results are reported in Fig. 7. The Y-axis represents the energy needed to send 100 data packets in order to have an uniform representation for all methods. As a general remark, mobile nodes consume more energy than fixed nodes because they send 1 data packet every 5s whereas fixed nodes only send 1 data packet every 30s. With standard RPL, nodes try to send packets even if the parent is not in the neighborhood. If the preamble is not acknowledged and the retransmission number is reached, the data packet is dropped whiteout being sent on the medium from the mobile node. This is why even with a large number of packets sent by standard RPL, energy consumption remains low, as only a few packets manage to actually be sent between nodes. Once mobile nodes are longer connected to their parent the energy consumption for them and the root rises. NUD and BFD send additional



(a) Grid topology (b) Random topology

Fig. 7. Average energy consumption of nodes
Node 0 - root. Last 5 nodes - mobile nodes. Intermediate nodes - static nodes

control messages in the network. Given the low number of data packets sent by mobile nodes using NUD, the energy consumption to send 100 data packets is the highest of all. BFD, although it sends control packets both ways between the mobile node and parent, has lower consumption when we take into account the energy consumed for 100 data packets. Using only RPL control messages sent when changes occur in the network and are signaled by layer 2, MT-RPL achieves the lowest energy consumption from all unreachability detection mechanism.

6 Related Work

In the literature authors have until now focused on analyzing path quality, packet delivery ratio or route prevalence in a network with RPL. However, knowing when a node should search for a new parent (as communication is no longer possible with the current one) should improve network performance. Authors in [3] analyze the quality of routes in RPL. Some routes are longer than the optimal ones. In addition, dominant routes, the ones that are used primary by nodes, are remarkably prevalent and long lived. Changes of routes degrade the path in half route changes, so it is important to adapt to network dynamics in order to preserve the best path. Their analysis also points out to the low PDR offered by RPL. Loses occur especially when RPL chooses low quality links. We can conclude that mitigating low quality links and maintaining routes close to the optimal value by mitigating network dynamics will improve network performance. In [11], authors study the robustness of RPL. Their findings show that RPL loses many packets and that congestion around the sink has an important impact on performance, degrading the PDR when the sink's congestion increases. Pointing out to the high dynamics observed even in a static network, changes in the DODAG can occur even after the network stabilizes. A node that enters the network or nodes that change parents, change the topology, introduce instability and increase the number of control packets sent. According to the rank of these

nodes, their unreachability impacts RPL control message overhead greatly. Although the authors have drawn important conclusions, the mechanism to detect the node unreachability is not clearly presented. It is to our belief that knowing what mechanism is better suited to mitigate network dynamics improves performances. The article [7] makes an analysis of route change latency using RPL and 6LoWPAN Neighbor Discovery protocol. Their analysis is theoretical and does not take into account any network dynamics. On a perfect stable network, it would provide an insightful view of route change latency. But, as papers such as [11] show how unstable a network with RPL can be, we believe that the author's contribution to understanding the stability of routes using 6LoWPAN ND is limited.

All these papers address the problem of network dynamics in RPL, but until now, a clearer analysis of the core components that allow RPL to mitigate the dynamic situations has not been available. Our work makes a complete overview of the unreachability methods suggested by RPL and lifts the uncertainty on which one is better to use in LLNs with RPL.

7 Discussion and Perspectives

In this article, we analyzed how the IPv6 Routing Protocol for Low-power and Lossy Networks (RPL) manages network dynamics, especially the support of mobile nodes. From our simulation campaign, we showed that the mechanisms suggested in the standard to mitigate dynamicity fail to prevent serious node disconnection, which significantly increases the packet loss together with the energy consumption. To the best of our knowledge, this is the first time that such mechanisms have been evaluated side by side. Results presented here could therefore serve the research community to increase the efforts on novel proposals for supporting mobile nodes in RPL. Then, we proposed a new cross-layer protocol operating at layers 2 and 3 known as Mobility-Triggered RPL (MT-RPL) to support efficiently mobile nodes in RPL. MT-RPL favors medium access to mobile devices and triggers RPL operations in order to maintain efficient connectivity with the network. Results obtained from an extensive simulation campaign showed that MT-RPL significantly reduces the disconnection time, increases the packet delivery ratio while limiting the energy consumption. MT-RPL is therefore a serious solution.

Encouraged by the results here, our future work will focus on a more precise evaluation of our proposal through more realistic scenarios. Furthermore, MT-RPL suffers from a large number of parent changes, increasing the number of DAO sent to the root of the graph. We will first investigate methods to reduce the number of parent changes without affecting the overall performance of MT-RPL. Then, we will focus on extending MT-RPL to all nodes, being mobile or fixed. Currently, we are considering favoring neighbor nodes with a better rank to serve as an opportunistic forwarder by introducing a delay proportional to the rank before acknowledging preamble strobe on behalf of the preferred parent. Finally, we expect to benefit from the FIT IoT testbed [1] to extend our

performance studies to large-scale experiments involving multiple mobile nodes. Many of the reasons why long disconnection time occurs are closely related to implementation, platform or operating system specifics that are quite delicate to do so properly with simulations.

References

[1] Future Internet (FIT) - Internet of Things testbed, `http://fit-equipex.fr/`
[2] Katz, D., et al.: Bidirectional Forwarding Detection (BFD). IETF RFC 5880 (2010)
[3] Ancillotti, E., et al.: RPL routing protocol in advanced metering infrastructures: An analysis of the unreliability problems. In: Sustainable Internet and ICT for Sustainability, SustainIT (2012)
[4] Ben Hamida, E., et al.: On the Complexity of an Accurate and Precise Performance Evaluation of Wireless Networks using Simulations. In: ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM (2008)
[5] Lewis, F.: Wireless sensor networks. Smart Environments: Technologies, Protocols, and Applications (2004)
[6] Teraoka, F., et al.: Unified Layer 2 (L2) Abstractions for Layer 3 (L3)-Driven Fast Handover. IETF RFC 5184 (2008)
[7] Kermajani, H., et al.: Route change latency in low-power and lossy wireless networks using RPL and 6LoWPAN Neighbor Discovery. In: IEEE Symposium on Computers and Communications, ISCC (2011)
[8] Allred, J., et al.: Sensorflock: an airborne wireless sensor network of micro-air vehicles. In: ACM SenSys (2007)
[9] Leguay, J., et al.: An efficient service oriented architecture for heterogeneous and dynamic wireless sensor networks. In: IEEE International Workshop on Practial Issues in Building Sensor Network Applications (SensApp) (2008)
[10] Polastre, J., et al.: Telos: enabling ultra-low power wireless research. In: International Symposium on Information Processing in Sensor Networks (2005)
[11] Heurtefeux, K., et al.: Experimental evaluation of a Routing Protocol for WSNs: RPL robustness under study. In: IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) (2013)
[12] M. Buettner et al.: X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks. In: ACM International Conference on Embedded Networked Sensor Systems (SenSys) (2006)
[13] Iova, O., et al.: Stability and efficiency of RPL under realistic conditions in WSN. In: IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC) (2013)
[14] P. Levis et al.: The Trickle Algorithm. IETF RFC 6206 (2011)
[15] Sikka, P., et al.: Wireless ad hoc sensor and actuator networks on the farm. In: ACM International Conference on Information Processing in Sensor Networks (2006)
[16] Kuntz, R., et al.: Improving the medium access in highly mobile Wireless Sensor Networks. Telecommunication Systems (2013)
[17] Narten, T., et al.: Neighbor Discovery for IP version 6 (IPv6). IETF RFC 4861 (2007)
[18] Winter, T., et al.: RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. IETF RFC 6550
[19] Haas, Z.J., et al.: The performance of query control schemes for the zone routing protocol. IEEE/ACM Transactions on Networking (2001)