RPL Mobility Support for Point-to-Point Traffic Flows towards Mobile Nodes

David Carels, Eli De Poorter, Ingrid Moerman, and Piet Demeester

Department of Information Technology (INTEC), Ghent University and iMinds, Gaston Crommenlaan 8 Bus 201, 9050 Ghent, Belgium

Correspondence should be addressed to David Carels; david.carels@intec.ugent.be

Received 13 February 2015; Revised 4 June 2015; Accepted 10 June 2015

The "routing protocol for low-power and lossy networks" (RPL) from the IETF ROLL working group is a widely used standard to support routing in wireless sensor networks (WSNs). Although the RPL protocol was originally designed with static topologies in mind, recently a number of extensions have been proposed to support traffic flows from mobile nodes towards a static gateway. However, this paper demonstrates that these solutions do not support traffic flows going the other direction, for example, from the gateway towards mobile devices. To remedy this, the paper first analyses the problems that prevent reliable traffic flows towards mobile devices when using RPL. Afterwards, a new mechanism to improve downward route updating is proposed. Our new approach minimizes the probability of connectivity loss by ensuring that the internal state of the static network remains consistent. Our solution is implemented and evaluated using both simulation tools and experimental facilities and it is shown that it improves the end-to-end packet delivery ratio to mobile nodes from 20–30% up to 80% while reducing the overall RPL signalling overhead without the use of location information.

1. Introduction

The IETF IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [1] is widely used to support routing between sensor nodes. In most scenarios, a backbone network of intermediary nodes is installed, which is assumed to be static. Although handling mobility is not an explicit design criteria for RPL [2], a number of use cases have been proposed in which also mobile nodes could be added to this static network [3–6]. Examples include mobile nodes in intelligent transportation systems (ITS) for the monitoring of air quality, mobile robots collecting information, and location tracking of occupants and assets. For most of these use cases, communication flows go from the mobile node to the sink and vice versa. In addition, more advanced use cases, such as continuous monitoring of patients where patients (data sources) and doctors (data collectors) are mobile [3], also require direct communication flows between mobile nodes.

RPL is optimised for data traffic in which the sink is colo-located with the root of the RPL tree, but also other communication patterns are supported [1, 2]. The protocol is based on the exchange of DIO (DODAG information object) messages to construct a DODAG (destination-oriented directed acyclic graph) for routing of data towards a DODAG root. RPL uses DAO (destination advertisement object) messages to establish downward routes towards the network nodes. These downward routes also enable the point-to-point communication between network nodes. The establishment of downwards routes leads to higher control traffic overhead and increased memory and processing requirements for (nodes near) the DODAG root [2].

RPL uses a tree-based routing approach for the fixed backbone network. However, the main characteristic of mobility is a highly dynamic topology which results in frequent disconnections with neighbouring nodes. Due to these disconnections, packets that are routed towards a mobile device can be routed towards edges (parents) even when the mobile device is already out of reach of these parents.

The main contributions of this paper are as follows:

(i) The paper thoroughly analyses the conditions during which packet loss can occur when routing towards mobile nodes in a RPL network.

(ii) In contrast to previous papers about mobility in RPL, that focused mainly on efficient parent selection by mobile nodes, this paper proposes a solution to keep...
the internal routing state of the backbone network consistent.

(iii) An improved route-cancellation algorithm has been proposed, implemented, and evaluated in a simulator using different criteria and scenarios, including different movement speeds.

(iv) The results have been experimentally verified using large scale wireless sensor testbed.

The remainder of this paper is organized as follows. In Section 2 an overview of the current state-of-the-art and related work on RPL and mobility support within RPL is presented. The problems that can occur with mobile nodes due to the standard RPL downward path construction mechanism are analysed in Section 3. To remedy these problems, an alternative downward path construction mechanism towards mobile nodes is proposed in Section 4. The evaluation setup is described in Section 5 and the evaluation results are presented in Section 6. Section 7 presents the experimental validation of the results. The paper is concluded by the future work in Section 8 and the conclusions of this paper in Section 9.

2. Related Work

This related work section will focus on three relevant research aspects: (i) the selection of objective functions for RPL, (ii) RPL support for traffic flows towards one or more mobile sinks, and (iii) RPL support for traffic flows from mobile nodes towards a static gateway (sink) node.

2.1. Link Estimation in RPL. To adapt the RPL protocol to different network conditions link estimation algorithms are used for evaluation of the path cost and definition of the properties of the constructed tree in an objective function (OF).

The most popular link estimation algorithm for RPL is the estimated transmission count (ETX) metric. This metric estimates the amount of transmission needed to successfully deliver a packet to a neighbouring node. The combination of the different ETX values defines the rank of that node in the DODAG. The default objective function of RPL is the minimum rank with hysteresis objection function (MRHOF) which uses ETX by default.

In [7] the problems concerning link quality estimation and neighbour management policies for high-density networks are discussed and evaluated. Because the ETX calculation is implemented as an exponentially weighted moving average and only evaluates the links that are currently being used no alternatives are evaluated. This can result in suboptimal routing topology after some time. The proposed solution is passive probing mechanism for new discovered links. These new discovered links are assumed to be excellent (the best possible value) instead of bad (the worst value). This results in the evaluation of each newly discovered link. The evaluation of this technique shows that some kind of passive probing is possible to ensure the evaluation of the link with every possible parent. This technique however requires a cache management policy to ensure already evaluated bad links are not deleted out of the neighbour table and afterwards readded suppressing good but not perfect links.

An evaluation of the impact of different metrics on the stability and efficiency of RPL is made in [8]. The paper illustrates the instability problem for all evaluated routing metrics (MinHop, ETX, and LQI). Also the trade-off between stability and efficiency for the evaluated metrics is illustrated.

The use of the nodes next hop remaining energy is compared as routing metric with the ETX routing metric in [9]. The paper concludes that the proposed mechanism and implementation increase the network lifetime with 14% compared to the popular ETX-based scheme and evenly distributes the energy over all the nodes of the network. However, the more energy efficient approach however results in a slightly lower number of received packets by the sink.

In RPL the selection of an appropriate parent is essential for the efficiency of the routing structure. Because the objective function is responsible for the selection of the parent, the selection of the best objective function is of high importance.

2.2. Sink Mobility in RPL. Because sensor nodes around a sink are more likely to spend their energy faster (due to their forwarding task), a different category of research papers focuses on solutions to support moving sinks to extend the network lifetime. Examples of modifications to RPL for these purposes can be found in [10, 11]. However, since these solutions focus on traffic flows to mobile gateways without support for mobile end-devices, we consider these as out-of-scope for this paper.

2.3. Routing from Mobile Node towards Static Sink in RPL. A major challenge for optimizing traffic flows from a mobile node towards a static sink is the timely selection of a new optimal parent node for the mobile node during its movement.

In [12] a simulation performance study of RPL for vehicular networks is presented. The study focuses on 10 nodes, with an intermediate distance and transmission range of 250 meters, traversing a straight line of 5000 meters, with the sink being in the middle of the line. The paper evaluates the performance of RPL under these mobile circumstances and varied the DIO (DODAG information object) message period for all nodes. The simulations were performed for speeds of 25 mph (40,2 km/h), 45 mph (72,4 km/h), and 65 mph (104,6 km/h). The conclusion illustrates that an increase in DIO frequency improves the PDR (packet delivery ratio). The proposed adaptations of RPL are an illustration of techniques which make it possible to use RPL for mobile low-power devices. However, the scenario used is specific for car-to-car routing, where neighbouring nodes travel together on the same path.

The ME-RPL (mobility enhanced RPL) [13] protocol proposes optimizations for the RPL protocol to support routing from mobile nodes over a static network towards a static sink node. A first technique is the use of explicit advertisement of mobility status information in control messages. As a result, nodes are able to distinguish between fixed nodes and mobile nodes. Fixed nodes are preferred as parent and routes through mobile nodes are avoided. Finally, DIS (DODAG information solicitation) messages are used to monitor the environment
and a mechanism is introduced that adapts the RPL-DIS message interval depending on the stability of the neighbourhood. This DIS message is normally only used when constructing a new network to discover neighbouring nodes. The simulation results show that the proposed optimizations perform better than the standard RPL protocol in terms of packet delivery ratio and route stability, but no experimental results are available.

Co-RPL [14] proposes an extension to RPL to support mobility. To improve the network performance the extension keeps track of the mobile nodes’ positions while moving. To allow localization of RPL nodes in motion the extension relies on the corona mechanism. This mechanism divides the network into circular areas around multiple DAG roots which are called coronas. Based on the distance from the DAG roots, this technique allows the computation of the position of the node. The extension is evaluated against standard RPL via a simulation study using Contiki/COOJA simulator. In the study the impact of the node speed, packet transmission rate and the number of DAG roots on the network performance is evaluated. The results show that Co-RPL decreases the average energy consumption by 50%, the packet loss ratio by 45%, and the end-to-end delay by 2.5 seconds.

Cobarzan et al. [15] have shown that RPL and standard unreachability detection systems with RPL fail in preventing node disconnection for mobile nodes. Therefore a new cross-layer protocol (MT-RPL) is proposed. This MT-RPL protocol benefits from the X-Machiavel MAC protocol that favors mobile nodes which want to transmit data. The new protocol achieves a packet delivery ratio of 62% to 66% for traffic from a mobile node towards a sink node. The paper also analyses the packet delivery ratio for traffic from root to mobile node which results in percentages of 23% to 36%.

In our research group we defined a standard compliant optimization of RPL. RPL with enhanced neighbour discovery optimizes the routing from a mobile node towards a static sink. This solution uses the DIS mechanism, an adapted link quality estimation function, an alternative parent, and detection of connection loss to obtain an end-to-end packet delivery ratio of more than 80% over a static “access” network from communication from a mobile node towards a static sink.

Due to the movement of mobile nodes, their neighbouring nodes and the link quality with these nodes constantly change. Because the mobile node has to rediscover its environment every time and this situation is similar to the discovery of the network at start up, the same discovery mechanism (DIS broadcast mechanism) to solicit for DIO messages from neighbouring nodes is used.

The solution also uses an alternative parent which, together with the preferred parent, is monitored via DIS unicasts.

To prevent the selection of a neighbour which is out of reach, old parents in the neighbour list are deleted if they do not respond with a DIO after a few DIS broadcast messages.

For the selection of a parent for a mobile node this paper will use our optimization of RPL because it is the most relevant to our scenario and it is standard compliant.

3. Downward Routing in RPL

To send packets to any node that is part of a RPL network, a downward route must exist towards this node. When a mobile device is part of the network, this downward route can fail. As such, before proposing optimizations in Section 4, this section first describes the existing mechanism for creating and maintaining downward routes.

3.1. Downward Route Construction. Downward routes are created from “bottom to top,” meaning that the downward route construction process is initiated by the node that should be reached. The node, to which a downward route has to be set up, sends a destination advertisement object (DAO) message (with the lifetime of the route as an option field) via its preferred parent towards the root. The preferred parent will then forward the DAO message via its own preferred parent towards the root of the DODAG. Using this mechanism, the DAO message propagates the destination information upwards to the root via the DODAG.

The DAO mechanism can work in two modes: storing mode and nonstoring mode. (i) In nonstoring mode, intermediary nodes on the path forward the DAO message towards the sink but do not store the destination information in their routing table. Only the sink node has knowledge about the exact topology. Downward messages are routed downward using IP source routing: the sink includes the full route information in the header of the packets. (ii) In storing mode intermediate nodes on the path between the root and the node store information about downwards nodes in their routing table. The routing table is used for routing packets downward using their IPv6 destination address. In this paper we will focus on the storing mode mechanism.

3.2. Downward Route Maintenance and Update. When a node receives a DAO message, the node computes if the DAO changes the set of prefixes that the node advertises. If so, it generates a new DAO message and transmits it to its parent. According to the RFC, the transmission of this DAO has to be delayed to aggregate information of other nodes. The default value suggested by the RFC for this delay is 1 second.

3.3. Downward Route Cancellation and Update. No-path DAO messages (DAO message with lifetime zero) are sent by nodes to their parent to remove a path towards a specific node (Figure 1). A node that receives a no-path DAO will send this packet further to its parent, to inform also its parent that it should cancel the downward route towards the original node. Route cancellations can be sent to indicate a forwarding error, indicate a neighbour unreachable error, or indicate when a mobile node loses reachability with its parent.

After the transmission of the no-path DAO, a node sends out a DAO message towards its new parent to establish a new downward path (Figure 1). The RFC specifying RPL mentions that, in order to aggregate DAO information of other nodes, the sending of a DAO message should be delayed.

3.4. Point-to-Point Routing in RPL. Communication between two nodes inside the network is based on the downward path.
4 International Journal of Distributed Sensor Networks

$$R \rightarrow R$$

$$R \rightarrow 11$$

$$R \rightarrow OP$$

C

$$21$$

$$NPOP$$

11

$$R$$

(2) DAO

$$R \rightarrow R$$

$$R \rightarrow NP$$

$$R \rightarrow 21$$

Figure 1: Downward route cancellation and setup mechanism of RPL.

$$R \rightarrow R$$

$$R \rightarrow OP$$

$$R \rightarrow NP$$

$$R \rightarrow R$$

Figure 2: Point-to-point routing in RPL.

Figure 3: Packet loss due to inconsistent subtrees in point-to-point traffic when old route is not cancelled.

mentioned previously, the mechanism for switching from a previous parent to a new preferred parent is described by the RFC as follows. (1) First, the node must transmit a no-path DAO towards the old preferred parent. (2) Next, the node must generate a DAO message to the new preferred parent to construct a new downward route from the root to the mobile node (Figure 1).

The mechanism described in the RFC can lead to inconsistent routing states when the no-path DAO does not arrive due to packet loss or when the mobile node is already out of reach of the old parent. In this case, the old downward route keeps on existing until the lifetime of the route expires (Figure 3).

The frequency with which this will occur depends mainly on the packet loss of the network and the movement speed of the mobile node and can potentially result in a large part of the tree with incorrect routing information.

A second inconsistency can occur when after deleting an old path (by a no-path DAO) a new parent is selected from the neighbour list that is no longer reachable (Figure 4). This will result in loss of connectivity because the old path is deleted and the new path is not set up since the DAO initiating the construction of a new path cannot be delivered to the new parent.

4.1. Route Maintenance Improvements. To remedy this situation, this paper proposes modifying the cancellation mechanism for downward routes. Currently when a DAO message arrives at a node which is a common ancestor for the old and the new route and the old route is not already cancelled, the old path is overwritten by the new path. However, the old path will still exist in the subtree of the old path until the lifetime of the path expires. Therefore, we propose to change the behaviour. Firstly, mobile nodes should not delay the transmission of the DAO, since this has a significant potential to lead to inconsistent states. Secondly, instead of generating a no-path DAO from the mobile node, the common ancestor should generate a no-path DAO upon receiving an updated

4. Downward Route to Mobile Nodes

This section will discuss the shortcomings of the RFC mechanism when used to communicate with mobile nodes. As construction for communication between sink and node. As stated in the RFC [1] defining RPL “A packet flows towards a root until it reaches an ancestor that has a known route to the destination.” If no common ancestor exists, that common ancestor may be the DODAG root. In other cases, it is a node closer to both the source and destination (Figure 2).

The mentioned issues can have a severe impact on point-to-point traffic. If the old path in the sub-DODAG of the common parent is not deleted, messages from all nodes in this sub-DODAG towards the destination node will be sent via the old path (Figure 3).
DAO. The no-path DAO created by the common ancestor should be propagated downward over the old route towards the old parent of the mobile node (Figure 5). (We assume that packet loss is most problematic for the mobile node, while the underlying static network is sufficiently reliable to deliver the different DAO messages. This assumption is also made in the RFC and is a requirement for the good functioning of the RPL protocol for static networks.)

This approach solves several of the identified shortcomings of the current behaviour. First, packets that are stored in a node between the common ancestor and the mobile node can still be delivered to the mobile node since the old downward route between the root and the destination can still be used until the new downward route is installed. Secondly, lost no-path DAO packets from the mobile node no longer result in an inconsistent state since the proposed mechanism will clean up the route in the sub-DODAG of the old common parent on the path to the destination node. As a result, point-to-point communication can no longer be disrupted due to the inconsistent state.

In addition to solving the inconsistencies, the proposed route update approach also has a lower total packet overhead. The packet overhead for the nodes on the old and new path between the common ancestor and the destination node remains the same (Figure 6). However, the destination node (R) and the nodes between the common ancestor (CA) and the DODAG root have to transmit only a single packet and the old parent (OP) has to transmit no DAO packet. The overall reduction of packets per parent switch can thus be expressed as \#hops(CA → root) + 1.

4.2. Alternative Solutions. For the sake of completeness, this section mentions several alternative mechanisms for route maintenance that we considered but did not implement since they have several other disadvantages:

(i) One alternative is to delay the transmission of no-path DAO from the mobile nodes until the DAO messages on the new path have propagated the new path. Delaying these no-path DAO messages ensures that at no point in time no downward route to the mobile node exists. However, this approach has the disadvantage that the probability increases that the previous preferred parent is already out of range and can no longer receive the no-path DAO, again resulting in inconsistencies in the tree.
(ii) A second alternative is to let the mobile node transmit a DAO with a limited lifetime towards the old preferred parent instead of a DAO with a lifetime of zero (no-path DAO). As a result, the old path will remain active only for a limited time. After the lifetime has expired, the path is deleted. This approach has two major disadvantages. (i) During the limited lifetime it is possible that the mobile node moves out of reach of the previous parent, which again makes the path incorrect. (ii) The new parent might be unreachable due to incorrect parent selection. In such a case the old path is cancelled on the moment the limited lifetime expires, resulting in a situation where no path is available any more to the mobile node (Figure 4).

The solution proposed before did not have either of these problems.

5. Evaluation Set-Up

The proposed enhancements to support communication towards a mobile node which moves inside a static grid of a wireless sensor network have been implemented in the open source Contiki operation system (version 2.7) [16]. To evaluate the performance of the improvements, experiments were performed with the COOJA simulator [17] with the configuration parameters of Table 1 and without the use of location information.

The simulations were performed for 4 topologies. The first two topologies use receiving ratio of 100%, a transmit range of 100 meters, and an interference range of 140 meters (Figure 7). In the first topology the sender will be the sink positioned in the upper left corner. For the second topology the senders were positioned in the lower left and upper right corner of the grid. With these topologies we simulate an almost ideal scenario in which every node (except the border nodes) has four ideal neighbours with perfect link quality.
6. Evaluation Results

Using simulations, we will compare the performance of (i) standard RPL, (ii) RPL with enhanced neighbour discovery for mobile nodes (see Section 2.3), and (iii) the proposed optimizations for routing towards mobile nodes. The first parameter for the evaluation is the average end-to-end packet delivery ratio which represents the percentage of received packets divided by the number of transmitted packets during the movement of the mobile node. A second parameter is the distribution of the delay of all the received packets during the whole simulation. In total 5010 packets are sent over the 30 simulation runs for each parameter.

6.1. Sink to Mobile without Packet Loss. The first scenario represents communication from a sink towards a mobile node in a lossless environment (Figure 7). In Figure 9(a) the average end-to-end packet delivery for the three evaluated protocols is presented, as well as the standard deviation. The results show that the standard RPL protocol is not suited at all for communication from sink towards a mobile device. The “mobile” solution, which was proposed to support communication from mobile nodes, slightly improves the performance, but the communication remains extremely unreliable. In contrast, the proposed solution drastically improves the end-to-end packet delivery ratio for all considered movement speeds.

In Figure 11(a) the delay of all the received packets is shown. The figure shows that almost all the received packets have a delay between 50 and 100 ms.

6.2. Sink to Mobile with Packet Loss. Similar experiments have been performed with packet loss (Figure 8); the results are shown in Figures 9(b) and 11(b). As expected, the average end-to-end packet delivery ratio is lower than in the previous simulations, but the optimizations still result in significant improvements (Figure 9(b)). Note that whereas in the previous simulations the packet loss was nonexistent, the packet loss used in the last simulations is higher than that observed in many (especially open air) environments. As such, most realistic deployments will probably exhibit behaviour that is intermediate between these scenarios.

If we compare the results of the average packet delivery ratio of standard RPL and the results of RPL with enhanced neighbour discovery for mobile nodes (Figure 9(b)) we ascertain a lower delivery ratio for RPL with enhanced neighbour discovery. This decrease can be explained by the loss of control messages to switch the downward path which lead to inconsistencies in the routing path towards the mobile node. Since for the standard protocol switching of parent occurs only rarely, the packet loss will only affect the actual data traffic and will not affect the consistency of the downward path.

The effect of packet loss on the average delay of all the received packets is shown in Figure 11(b). Due to retransmissions, the number of packets with a delay between 50 and 100 ms is drastically decreased for all scenarios.

6.3. Point-to-Point Sending towards Mobile without Packet Loss. Whereas the previous scenarios evaluated traffic from the sink to the mobile node, we now evaluate the performance of point-to-point communication from two infrastructure nodes, situated in the lower left and upper right corner of the grid (Figure 7) towards the mobile node. Again, the performance of the optimized solution improves the overall performance by remedying the internal inconsistencies (Figure 10).
The average delay of the received packets (Figures 12(a) and 12(b)) is lower than that in the previous scenario since the mobile node moves closer to the sender during its movement. However, it is worth noting that the delay of the unoptimized solutions is lower than the one of the optimized solution. This is because a faulty internal state is more likely to result in packet loss when packets have to traverse multiple hops. As a result, packet loss for the unoptimized solutions occurs mainly for those packets that have to traverse multiple hops, whereas the packets with a low number of hops (low delay) are still received. We illustrate the above explanation by analysing the number of packets that are received over time by each solution (see Figures 13(a), 13(b), and 14).

The main part of the received packets, from the upper right corner, has a delay between 100 and 200 ms for the standard and optimized scenario. For the protocol with only optimized neighbour discovery the main part has a delay lower than 50 ms.

When using the standard RPL, received packets arrive for both senders periodically at the same time period (Figure 13(a)). Because standard RPL fails in the timely selection of a new parent and in updating the downward path, the existing path stays in place and all packets are routed to the old parent of the mobile node. Every time the mobile node comes within reach of the old parent, the packets sent by the two senders arrive at the mobile node. This also explains why the delay for the sender in the upper right corner (Figure 12(a)) is higher than the delay for packets from the sender in the lower left corner (Figure 12(b)), which is the initial parent.

When considering the reception pattern for the RPL protocol with optimized neighbour discovery (Figure 13(b)) the packets also arrive periodically, but they are no longer synchronized. In this case, the mobile node selects new parent nodes, but the old path cannot be cancelled because the old parent is out of reach so the no-path DAO cannot be delivered. As a result, packets arrive mainly when the node is in reach of the sending node.

Finally, Figure 14 shows the reception pattern for the optimized protocol. Since the mobile node constantly evaluates its parents and old routes are cancelled by sending no-path DAO messages downwards via the old path, packet reception is not systematically interrupted.

6.4. Point-to-Point Sending towards Mobile with Packet Loss. Finally, the results obtained when two infrastructure nodes
are transmitting towards a mobile node in a lossy environment are similar to the previous observations. The overall end-to-end delivery ratio is lower due to internal packet loss, and as a result the average end-to-end delay increases.

7. Experimental Evaluation: iMinds wiLab.t Office Testbed

To validate the conclusions from the simulations, additional tests were performed on a real-life testbed (iMinds wiLab.t). The setup and results are discussed in this section.

7.1. Testbed Setup. The wiLab.t testbed [18, 19] is a testbed which consists of Tmote sky sensor nodes connected to an embedded PC installed at the ceiling on almost 200 locations on three floors in an office environment. The embedded PCs are interconnected via a switch over Ethernet. For the wireless communication in these tests, the IEEE 802.15.4 interface with channel 26 and a transmit power of −15 dBm (level 7) was used. The topology and selected nodes from the wiLab.t testbed are displayed in Figure 15.

The tests were performed on two floors of the testbed. The sink was positioned centrally on the second floor (node 107). The connection with the third floor is possible through nodes...
installed in the technical shaft (nodes 10-50-70-101) between the two floors. This results in a DODAG tree with two main branches, each of which connects via one of the technical shafts towards the third floor of the testbed. On the third floor a mobile node moves from one side of the floor towards the other side of the floor to force the switching of one branch towards the other branch of the DODAG.

Two nodes (nodes 57 and 95) on the second floor send data packets towards the mobile node. Since no automated mobility is supported in the used testbed, a mobility pattern was emulated by having a person walk from the left side of the floor (under node 200) towards the right side of the floor (under node 2) and back to the left side of the floor, with a stationary moment at each end point. The mobile node was attached to a laptop for logging purposes. Due to limitations of the office environment only slow mobility (walking speed) could be tested. Multiple repetitions of the same movement pattern were used.

7.2. Experiment Results. In Figure 16 the received packets by the mobile node in relation to the packets sent towards the mobile node (by the two sending nodes) is represented for the standard (Figure 16(a)) and the optimized protocol (Figure 16(b)). In this scenario there is no stationary moment between moving from the left to the right and from the right to the left on the third floor of the testbed.

In Figure 16(a) the connection is lost by the standard RPL protocol and is only reestablished when the mobile devices moves back to its original location. For the optimized protocol (Figure 16(b)) the packet reception is much more continuous, but during parent switches some packet loss occurs.

Figure 17 compares the packet reception pattern for the standard RPL protocol (Figure 17(a)) and the optimized RPL protocol (Figure 17(b)) in case of a stationary moment of 3 minutes (represented by a dotted line in the graph) before going back to the original location. The effects of this stationary moment can clearly be observed in the standard RPL protocol behaviour, illustrating the long period it takes before standard RPL detects inconsistencies in its routing tree. The graph shows that as long as the mobile node is in reach of its parent the packets from both senders, which are routed via the preferred parent, arrive at the mobile node. Once the mobile node is out of reach of the preferred parent, no packets arrive at the mobile node. When the mobile node starts to move back again, after some time the node is again in reach of its parent and the transmitted packets are received.
again by the mobile node. These shortcomings are remedied by the improved neighbour monitoring and the new path-update mechanism of the proposed solution.

In terms of packet delivery ratio, the end-to-end packet delivery ratio for two senders (on the second floor) towards a mobile node with stationary moments (on the third floor) is between 61% and 78% depending on the duration of the stationary period. The slight increase of the end-to-end packet delivery ratio for longer stationary moments can be explained because the link quality will be more stable resulting in less parent switches.

For the standard protocol the end-to-end packet delivery ratio is much lower: between 25% and 35% depending on the length of the stationary moment. In case no stationary moment is used, the delivery ratio is around 70%, which is comparable to the optimised scenario. This last percentage is however not representative because the mobile node is only a short period out of reach of its original parent.
7.3. Experiment Conclusions. Due to the use of a real life environment, the experimental tests are influenced by outside network conditions and as such are not fully repetitive. However, the experiments illustrate that the standard RPL protocol cannot handle fast parent change and fast downward route update for nodes moving out of reach. The obtained results validate the simulations and prove the need for modifying the deletion and updating mechanism for downward routes, especially for routing towards mobile nodes.

8. Future Work

Further research about the optimization of the parent selection mechanism can possibly lead to an increase of the end-to-end packet reception ratio and a decrease in terms of signalling overhead. A reduction of the total overhead can be obtained by the intelligent selection of alternative and preferred parent. If this selection, especially the switching of the preferred parent, is done intelligently, the number of downward path switches is minimised which reduces the packet loss and total overhead.

One issue which is not yet solved by the optimizations proposed in the paper is the fact that packet loss can occur when the old parent cannot reach the mobile node any more, but the old route is not yet cancelled by a new DAO. This point of failure also exists in the old protocol, where an old route cannot be cancelled when at the moment of the parent switch the old parent is no longer in reach.

In an ideal switching case the possibility of communication with the old parent still exists some time after the switching of parent to ensure that packets still travelling on the old downward path still can be delivered. If the connection with the old and the new preferred parent remains possible until the complete downward route is updated (and the old route is completely cancelled), this can possibly lead to a total elimination of packet loss due to the updating of the downward route.

The best moment for switching from one preferred parent towards another preferred parent happens when the link quality with the old parent is decreasing and the link quality with the new parent is increasing. If the location of the infrastructure nodes and the path of the mobile node are known, ideally the next selected parent is selected to ensure the longest connection time between the mobile node and the new parent. Since this information is typically unavailable, a more realistic switching scenario is that a mobile node selects a parent where it is still moving towards and switches when it detects that the link with the current preferred parent is degrading. Especially for the maintenance of the downward paths towards the mobile node, it is important that the number of neighbour switches is minimized. On the other hand the switching has to be finished before connection is lost between the mobile node and the current parent. Future work could include historical information to estimate the speed of the device and use this information to dynamically adapt the speed of neighbour switches.

9. Conclusion

This paper has shown that communication towards mobile nodes is not feasible in standard RPL protocols due to slow parent switching mechanism and the occurrence of faulty states in subtrees when mobile nodes move out of reach of a parent. Although previous RPL optimizations to support mobility have been proposed, these did not remedy the problem of incorrect downwards route maintenance. To this end, this paper (i) analysed the downward path mechanism and identified shortcomings for routing towards mobile nodes, (ii) proposed optimizations to enhance routing towards mobile nodes, and (iii) validated these optimizations in simulation with different scenarios, different movement speeds, and without the use of location information. The validation of the optimizations was also verified in real-life experiments.
This study is the first to enable communication to a mobile node using the RPL protocol, either from a sink (sink-to-point) or from another RPL node (point-to-point). The solution was shown to reduce the overhead for communicating route updates through the RPL DODAG and a mathematical formula for calculating the packet reduction per parent switch was given. Further reduction of the total overhead can be obtained by the intelligent selection of alternative and preferred parent. Simulations and experimental deployments showed improvements of the end-to-end packet delivery ratio by up to 40%, depending on the scenario. In realistic scenarios, the obtained improvements will depend mainly on the frequency with which the mobile node moves towards parts of the tree that are inconsistent.

After adding the proposed enhancements of this paper to RPL, when using one of the mobility mechanisms for upwards routes, the reliable communication from and towards mobile nodes via RPL becomes possible.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] T. Winter, P. Thubert, A. Brandt et al., RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, IETF ROLL Working Group, 2012, http://datatracker.ietf.org/doc/rfc6550/.

[2] T. Clausen, J. Yi, U. Herberg, and Y. Igarashi, Observations of RPL: IPv6 Routing Protocol for Low Power and Lossy Networks, Network Working Group, 2014, http://tools.ietf.org/html/draft-clausen-lln-rpl-experiences-08.

[3] J. Martocci, P. de Mil, N. Riou, and W. Vermeylen, Building Automation Routing Requirements in Low-Power and Lossy Networks, Internet Engineering Task Force (IETF), 2010, http://tools.ietf.org/html/rfc5867.

[4] A. Brandt, J. Buron, and G. Porcu, “Home automation routing requirements in low-power and lossy networks,” Internet Engineering Task Force (IETF), 2010, http://tools.ietf.org/html/rfc5826.

[5] K. Pister, P. Thubert, S. Dwars, and T. Phinney, Industrial Routing Requirements in Low-Power and Lossy Networks, Network Working Group, 2009, http://tools.ietf.org/html/rfc5673.

[6] M. Dohler, T. Watteyne, T. Winter, and D. Barthel, Routing Requirements for Urban Low-Power and Lossy Networks, Network Working Group, 2009, http://tools.ietf.org/html/rfc5548.

[7] S. Dawans, S. Duquennoy, and O. Bonaventure, “On link estimation in dense RPL deployments,” in Proceedings of the International Workshop on Practical Issues in Building Sensor Network Applications (IEEE SenseApp ’12), Clearwater, Fla, USA, 2012.

[8] O. Iova, F. Theoleyre, and T. Noel, “Stability and efficiency of rpl under realistic conditions in wireless sensor networks,” in Proceedings of the IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC ’13), pp. 2098–2032, September 2013.

[9] P. O. Kamgueu, E. Nataf, T. D. Ndie, and O. Festor, “Energy-based routing metric for RPL,” Rapport de Recherche RR-8208, INRIA, 2013.

[10] L. B. Saad and B. Tourancheau, “Sinks mobility strategy in IPv6-based WSNs for network lifetime improvement,” in Proceedings of the 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS ‘11), pp. 1–5, February 2011.

[11] B. Tian, K. M. Hou, H. Shi et al., “Application of modified RPL under VANET-WSN communication architecture,” in Proceedings of the 5th International Conference on Computational and Information Sciences (ICCIS ’13), pp. 1467–1470, June 2013.

[12] K. C. Lee, R. Sudhaakar, J. Ning et al., “A comprehensive evaluation of RPL under mobility,” International Journal of Vehicular Technology, vol. 2012, Article ID 904308, 10 pages, 2012.

[13] I. E. Korbi, M. Ben Brahim, C. Adjih, and L. A. Saidane, “Mobility enhanced RPL for wireless sensor networks,” in Proceedings of the 3rd International Conference on the Network of the Future (NOF ’12), pp. 63–70, November 2012.

[14] 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 Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES ’14), pp. 200–209, June 2014.

[15] C. Cobarzan, J. Montavont, and T. Noel, “Integrating Mobility in RPL,” in Wireless Sensor Networks, T. Abdelzaher, N. Pereira, and E. Tovar, Eds., vol. 8965 of Lecture Notes in Computer Science, pp. 135–150, Springer International Publishing, Cham, Switzerland, 2015.

[16] Contiki: the open source OS for the internet of things, http://www.contiki-os.org/.

[17] An introduction to cooja, 2015, https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja.

[18] S. Bouckaert, W. Vandenbergh, B. Jooris, I. Moerman, and P. Demeester, “The w-ilab.t testbed,” in Testbeds and Research Infrastructures. Development of Networks and Communities: 6th International ICST Conference, TridentCom 2010, Berlin, Germany, May 18–20, 2010, Revised Selected Papers, Institute for Computer Sciences-Social Informatics and Telecommunications Engineering (ICST), Ed., vol. 46 of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 145–154, Springer, Berlin, Germany, 2011.

[19] iMinds technical testing: ilab.t—w-ilab.t hardware and lay-out, 2015, http://ilabt.iminds.be/wilabt/hardwarelayout.
