Analysis and Performance Evaluation of RPL under Mobility

Cosmin Cobârzan, Julien Montavont and Thomas Noël
ICube laboratory (CNRS), University of Strasbourg, France
Email: {cobarzan,montavont,noel}@unistra.fr

***Abstract*—Wireless Sensor Networks (WSN) have become popular in the last years. Paths between sensors are computed by routing protocols, which are specifically designed to cope with the characteristics of WSN. Lately, the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) has been accepted as the Internet Engineering Task Force (IETF) standard for routing packets in WSN. However, this protocol does not pay particular attention to mobile nodes. Furthermore, mobility has been contemplated as a way to improve sensing coverage and connectivity in unattended WSN. In addition, WSN will account for an increasingly huge number of connections, from 1.9 billion devices today to 9 billion by 2018 among which most of them would be mobile (e.g. wearable devices). Using RPL to support the mobility of nodes is the main problem investigated in this article. We propose a new version of the trickle algorithm which allows mobile nodes to move seamlessly into a routing topology build by RPL together with limiting the signalling overhead. Our proposal is compared to two other schemes available in the literature. Results highlight that our proposal significantly reduces disconnection times and increases the packet delivery ratio while mitigating the extra control traffic.**

Keywords*—*Wireless Sensor Networks, RPL, Mobility

I. INTRODUCTION

Smart objects begin surrounding us from every direction. A short while ago they gave birth to what is now called the Internet of Things, a network of interconnected smart objects. This is a major departure in the history of Internet, as connections move beyond computing devices and begin to power billion of everyday devices, from household electrical goods to wearable devices. The necessity to provide the least intrusive experience or collect fine-grained information (e.g. using body sensors in a medical environment as depicted in [1]) will certainly make use of a huge number of mobile sensors or devices travelling across dense topologies of fixed nodes. Besides, mobile sensors enable building new applications that cannot be feasible with fixed nodes such as target tracking or surveillance applications as depicted in [2], [3], [4]. In that context, considering the envisioned growth of connected devices (9 billion by 2018 [5]), a particular attention should be provided to efficiently support mobile nodes. Lately, the IETF proposed an IPv6 routing protocol for low-power and lossy networks known as RPL [6]. This protocol builds a Destination Orientated Directed Acyclic Graph (DODAG) using an objective function along with a set of metrics/constraints. Once connected to the graph, a node is configured with a parent (the next hop to the root of the graph) and a rank (which estimate the location of the node in the graph). Until recently, RPL was mainly used in static networks [7]. Only a few attempts took mobile nodes into consideration [8], [9].

RPL has no restriction on the participation of mobile nodes in DODAGs. Once a mobile node moves in the graph, it is likely to be disconnected from its current parent. In order to re-attach itself to the graph, the mobile node should discover new parents available in its new surroundings. In short, sending DODAG Information Solicitation message (DIS) allows a (mobile) node to gather information about potential parents in the vicinity. However, the specifications of RPL [6] do not define how and when DIS should be sent. In addition, a node keeps its current parent until it receives announces (referred to as DODAG Information Object or DIO) from surrounding parents with a better rank or when a new version of the graph is built. As a result, a disconnected node may solicit new DIO by sending DIS but remains disconnected as all pending DIO include a higher rank than its current one. In such situation, exchanging DIS and DIO does not allow mobile or static nodes to re-attach to the graph. In this paper, we propose a new approach to support mobile nodes in RPL. Our proposal introduces local operations on nodes that act as mobile nodes' parents in the graph. By this means, only nodes located on the path of mobile nodes are involved, without endangering the configuration of the whole network. Furthermore, our new operations include a new trickle algorithm [10] that allows fast re-attachment to the graph. Our proposal is evaluated together with two solutions from the literature [8], [9] through 2 scenarios ranging in size and complexity from a theoretical case to a more realistic one. Our simulation results show that our proposal significantly reduces the disconnection times and increases the packet delivery ratio while mitigating the extra control traffic.

The rest of the paper is organised as follows. First, we briefly present RPL and pertinent contributions that tackle mobility in RPL. Section III makes an overview of our proposed solution to support mobile nodes in RPL. The simulation parameters and results of the performance evaluation are detailed in Section IV. Finally we give some concluding remarks along with future investigations in Section V.

II. RELATED WORK

A. RPL Basics

RPL is a distance vector IPv6 routing protocol. It builds a Destination Orientated Directed Acyclic Graph (DODAG) using an objective function along with a set of metrics/constraints. The root of the graph is situated at the border router between the Internet and the smart objects, which form a wireless sensor network. RPL introduces new messages known as DODAG Information Object (DIO), DODAG Information Solicitation (DIS) and DODAG Destination Advertisement

Object (DAO). DIO advertises the characteristics of the graph such as the objective function in use, the node rank and the graph version, and allows the creation and maintenance of DODAGs. Once nodes connect to the graph, they will relay further the DIO sent by the root in their own neighborhood. A node willing to connect to a graph can request the transmission of DIO from nodes attached to the graph by sending DIS. Finally, DAO enables the support of point-to-multipoint and point-to-point communication by propagating destination information upward along the graph.

The root is the only entity that can launch the construction of the DODAG by sending the first DIO in the neighborhood (cf. Figure 1). Govern by the trickle algorithm [10], the sending of DIO will shape the graph according to the objective function with minimal control traffic overhead. A node receiving DIO attaches to the graph, computes its rank according to the objective function and starts advertising the graph further using DIO. If a node does not want to wait for a DIO (e.g. Node 3 in Figure 1), it can send a multicast DIS which triggers the transmission of DIO and resets the trickle timer from nodes already attached to a DODAG. Attaching to the graph (by processing a DIO) ensures a route from the nodes to the root. To establish a downward route, a node advertises its address via the transmission of DAO. RPL supports two modes for managing downward routes - storing and non-storing. In the storing mode, nodes store downward routing tables for their sub-graph. The non-storing mode routes downward packets via a source routing mechanism. Regular messages have to be sent, as advertised addresses have a limited lifetime. Using these three control messages RPL builds a logical topology between nodes and enables bi-directional paths between nodes and root with minimal overhead. In the following, we only consider the non-storing mode of operation.

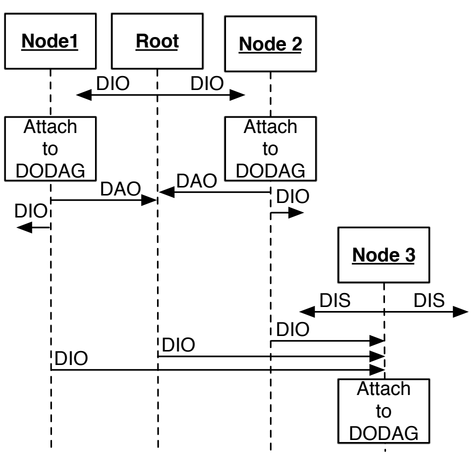

Fig. 1: DODAG construction

The construction of the DODAG is mostly governed by the metric advertised in the objective function. As RPL is a routing protocol, the metric is centred on expressing the quality of the link. A good estimation of the quality of the link is still an open issue in the community [11]. But several metrics which are widely used have been proposed such as MinHop [12], ETX [13] and ETOP [14]. Although ETOP is the most recent one, we particularly focus on ETX because it is the default metric used in the RPL implementation available in well-known operating systems for WSN such as TinyOS and Contiki. In addition, proposals presented in Section II-B also use ETX in their performance evaluation. ETX is defined as the predicted number of transmissions required to successfully send a packet over a link. The metric enables high throughout as it is based on the Packet Delivery Ratio (PDR) between the source and destination. The PDR is evaluated actively from every sent or received packet and gives an instantaneous quality evaluation of the link. Nevertheless, ETX can induce unwanted instability in the network due to its fast reaction to changes.

B. Mobility in RPL: problem statement and solutions

Due to the relatively short range of the wireless technology used in WSN (e.g. IEEE 802.15.4), it is likely that a mobile node moves out of the range of its current parent and therefore become disconnected from the graph built by RPL. In such situation, a mobile node can re-attach to the graph through the reception of new DIO followed by the transmission of a new DAO. However, a (mobile) node may wait for a long period before receiving the next scheduled DIO due to the trickle algorithm [10] (e.g. 2.3 hours calculated using default values from [6]). Nevertheless, a node has the option to send DIS which trigger the immediate transmission of DIO from surrounding nodes that are already attached to the graph. Still, the specifications of RPL do not define how and when DIS should be sent. In addition, a node only changes its current parent upon reception of DIO with better rank. As a result, a disconnected node can solicit new DIO but stays disconnected as all received DIO advertise a rank worse than its current one. In such situation, exchanging DIS and DIO does not allow mobile or static nodes to re-attach to the graph. This problem arises due to the management of the parent set. RPL does not specify how a parent (and particularly the preferred parent) that becomes unreachable can be removed from the parent set. Although RPL suggests the use of external mechanisms for unreachability detection, the proposed solutions seem too complex (Neighbor Discovery or Bidirectional Forwarding Detection) or dedicated to a specific MAC layer (hints from lower layers via Layer 2 (L2) triggers) to be practical. To the best of our knowledge, there is no such attempt in the literature. In conclusion, RPL already includes the mechanisms required to enable re-attachments to the graph but suffers from the poor support (or lack thereof) for rapid detection of parent unreachability in order to reset the rank of mobile nodes.

In literature, few attempts have been made to use RPL in mobile environments. The following proposals are evaluated together with our solution in Section IV. The authors of [8] propose to remove the trickle algorithm for DIO and replace it with a fixed value ranging from $2s$ to $10s$. Obviously, the DIO message overhead is proportional to the DIO period - lower the DIO period, the higher the DIO message overhead. But lower DIO periods allow the reduction of disconnection time, as new DIO is processed more frequently. However, a fixed sending period will generate constant DIO in the whole network, leading to extra control traffic even in areas not crossed by mobile nodes. Such growth in control traffic may significantly increase the energy consumption together with the contention on the wireless medium. In addition, this proposal relies on the method used by RPL to change parents. As presented previously, the reception of new DIO does not necessarily trigger a parent change, even if the current parent is unreachable. Nevertheless, the authors mitigate such situation by using ETX.

In [9] the authors propose a dynamic DIS management procedure. In a nutshell, they send DIS with different intervals to force the refresh of routing information. When a mobile node experiences high mobility (i.e. it changes several times its preferred parent), the interval between two consecutive DIS should be small in order to allow for fast routing information updates. By contrast, if a mobile node remains connected to the same parent, the interval between two consecutive DIS should be large because the node is not moving or moving slowly. For this, they define $DownDIS$ which is the number of parents change above which the inter-DIS interval should be divided by 2, $UpDIS$ which is the number of times a mobile node chooses the same parent above which the inter-DIS should be multiplied by 2, and I_{min} and I_{max} which are respectively the minimum and maximum time interval for the inter-DIS. This solution is interesting as it only involves nodes close to mobile nodes and not the whole network. However, the transmission of a DIS may unnecessarily reset the trickle timer of all nodes in range (i.e. starting again with frequent DIO) and trigger the transmission of multiple DIO while the mobile node is still connected. Furthermore, this proposal also relies on the metric used by RPL to change parents. Situations in which a mobile node does not process new DIO due to the rank value is also mitigated by the use of ETX. According to their results, they suggest to set $DownDIS = 1$, $UpDIS = 5$, $I_{min} = 3s$ and $I_{max} = 60s$. However, they do not specify the default value I_{init} to start the algorithm.

III. Proposed Mobility Management

In this section, we present our new approach to support mobile nodes in RPL. Our proposal only involves nodes interacting with mobile nodes, so it does not modify the configuration of the whole network. Furthermore, our new operations include a reverse trickle timer that allows fast re-attachment to the graph. Our proposal assumes that mobile nodes should have minimum impact on the connectivity or stability of the DODAG. Selecting a mobile node as a preferred parent can generate loops in the network [9]. We therefore advocate that mobile nodes only connect as leaves in the graph, i.e. they do not advertise further the graph with DIO. Such limitation is already included in the specifications of RPL. Our proposal also consider that, similarly to [9], the mobile nodes are configured a priori with mobility capabilities and advertise their mobility status in control message. For this, we choose to modify the DAO send by mobile nodes, by introducing a Mobility Flag (MF) in the flags field of DAO.

A. The reverse trickle timer

The standard trickle algorithm starts with a short interval between two consecutive DIO in order to quickly react upon the various topology changes that occur when the graph is building. When events that might trigger the reset of the trickle timer - a multicast DIS is received, new version of DODAG advertised, a loop is detected - do not occur, the interval between DIO increases in order to limit the control traffic overhead. To support mobile nodes, we propose a reverse trickle timer that starts from the maximum allowed value and halves the sending intervals after each new DIO. Let denote $Imin$ and $Imax$ the minimum and maximum interval between two consecutive DIO. When the mobile connects to a new parent, it is likely that it remains connected to this parent for a long time. The interval between DIO should be large as no change is expected in the next period, so it is set to a value close to $Imax$. Then, the more the mobile node spends time connected to the same parent, the more it is likely to move outside the coverage of the parent. So after each DIO, the reverse trickle timer will decrease the next DIO interval. Once the trickle timer reaches $Imin$, the parent requests new DAO from any connected node. If a new DAO from the mobile node is received, the algorithm starts again. A parent switches from the standard trickle timer to the reverse one whenever a mobile node connects to it. By contrast, when the last mobile node leaves a parent, this parent switches back to the standard trickle timer. Precisely, this procedure is illustrated by Figure 2.

```
if DAO_reception && MF = 1 then
    mobile_node ← TRUE
end if
while mobile_node = TRUE do
    Itmp ← Imax
    while Itmp/2 > Imin do
        I ← random(Itmp/2 , Itmp)
        Schedule next DIO in I
        Itmp ← Itmp/2
    end while
    I ← random(Imin , Itmp)
    Schedule next DIO in I with DTSN+1
    if !DAO_reception or DAO_reception && MF ≠ 1 then
        mobile_node ← FALSE
    end if
end while
Go back to standard trickle timer
```

Fig. 2: Reverse trickle timer algorithm

B. Protocol overview

When a mobile node enters a DODAG it is disconnected at first and will send DIS messages in multicast in order to discover any DODAG in its neighborhood. Once it receives DIO, it chooses a parent and attaches to the graph. The next step is to advertise the mobile node address to the root in order to calculate the downward route. We do this by sending DAO toward the root with MF set to 1. When the DAO reaches the parent, it will check MF before forwarding the message upward. A parent receiving a DAO with MF set to 0 (i.e. from a fixed node) does the default RPL action for DAO. If MF is set to 1, the parent registers the address of this mobile node and starts the reverse trickle timer. If no change appears in the network, the procedure continues until the minimum value for the trickle timer is reached. At that time, the parent will send a multicast DIO with an increased Destination Advertisement Trigger Sequence Number (DTSN), which triggers the sending of a DAO from any attached nodes. If the mobile node is still attached to the parent, it will respond with a new DAO with MF set to 1. Upon reception, the parent resets the reverse trickle timer and starts again the algorithm. If no DAO from a mobile node is received by the parent, it goes back to the standard trickle timer.

On the mobile node side, they monitor the time since the last received DIO from their parents. Let $Dthresh$ denote the number of summed DIO intervals, out of which the reverse trickle algorithm chooses a value, before a mobile node considers that it lost connectivity with the former parent. The maximum value for $Dthresh$ is the sum of the maximum number of DIO intervals that can be obtained between I_{max} and I_{min}. As the first DAO triggers the usage of the reverse trickle timer, they can bound the time at which they are supposed to received next DIO to $Dthresh$. So after each DIO, the mobile node set a timer to $Dthresh$. When $Dthresh$ is exceeded, the mobile node removes its former parent and set its rank to infinity. Then, it starts sending DIS in multicast to discover new parents through DIO reception. The infinite rank allows the processing of new DIO, even if those DIO show a higher rank than before the disconnection of the node. Upon reception of new DIO, the mobile node can select a new parent and sends back a DAO with MF set to 1. All operations are illustrated on Figure 3.

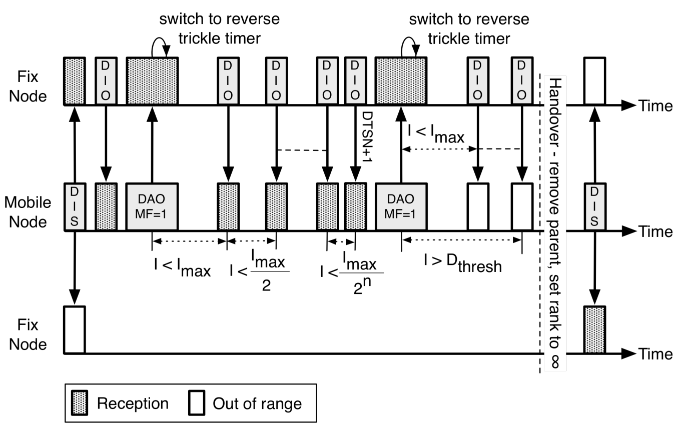

Fig. 3: Proposed algorithm

IV. Simulation Setup and Results

A. Simulation Scenario

In order to evaluate RPL's performance when operated with mobile nodes, we used the WSNet software [15]. 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 Section III. Furthermore, we implemented the two solutions from the literature referred to as PeriodicDIO [8] and DynamicDIS [9].

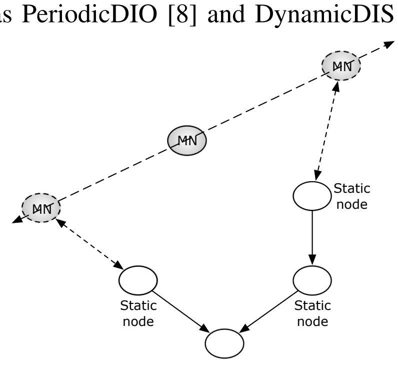

Fig. 4: Analysed academic scenario

All simulation parameters are presented in Table I. We deployed two topologies. The first scenario is an academic case scenario in which mobile nodes can only choose one parent. This scenario is presented in Figure 4. The second scenario aims to provide a real environment in which the DODAG root is located in the middle of a 425x425m area. 100 static nodes are positioned as a regular square grid, in a way that the whole simulation perimeter is covered at the radio level. Finally, 10 mobile nodes are initially distributed in a random fashion over the simulation grid and move randomly inside the area covered by the fixed nodes. When the mobile nodes move in the network, we do not change anymore the parents for the static nodes. We can thus better analyse only the effect of handovers between the mobile and the static nodes.

Simulation parameter	**Value**
Academic scenario	1 root, 3 static nodes, 1 mobile node
Realistic scenario	1 root, 100 static nodes, 10 mobile nodes
Data collection scheme	Time driven 1 packet/10s static nodes → root 1 packet/s mobile node(s) → root
Data payload size	56 bits/127 bytes
Mobility model	Billiard, 1m/s, linear trajectory (academic scenario) and random trajectory (realistic scenario)
Routing model	RPL in non-storing mode using ETX
RPL default values	DIO - given by trickle timer algorithm [10] DIS - 60s if no parent, until attached to DODAG DAO - 60s from every node, or when needed
Analysed schemes	
Reverse trickle	$Imin_1 = 2^{10}$ms and $Imax_1 = 2^{24}$ms $Imin_2 = 2^{11}$ms and $Imax_2 = 2^{25}$ms $Dthresh$ is set to 1, 3 and max
PeriodicDIO [8]	1 DIO/2s and 1 DIO/10s
DynamicDIS [9]	DownDIS = 1, UpDIS = 5, I_{min} = 3s, I_{max} = 60s I_{init} = 3s or 30s
MAC model	Unslotted CSMA 802.15.4 (802.15.4-2006)
Radio model	Half-duplex, Sensibility level: -92dBm, Channel 0, 250 kB/s bandwidth, 50m disk range
Antenna model	Omnidirectional, modulation OQPSK
Simulation setup	20 simulations/configuration 10 configurations (including PeriodicDIO and DynamicDIS schemes)
Academic scenario	1 handover/simulation
Realistic scenario	1 hour/simulation

TABLE I: Simulation parameters

B. Results

The results presented in this section were obtained after running 20 simulations of each scenario for each configuration. The presented results are the average of overall data collected from the set of simulations. The 95% confidence interval indicates the reliability of our measurements. We analysed three main parameters: overall number of DIO sent in the network, disconnection time and packet delivery ratio.

In the academic scenario, the mobile node changed only once the parent. The mobile node followed a path described in Figure 4. We analysed this simple scenario to better understand how a handover is performed. It is a good measure of how handovers will affect both the mobile node, on its path in the network, and the static nodes which it chooses as parents. It can be seen in Figure 5 that, if only the parent to which the mobile node connects send DIO, the overall number of control packets is reduced. In our solution and in DynamicDIS, DIO are sent only locally. The two solutions send less control packets than PeriodicDIO, in which all the network is flooded with DIO. Sending messages when needed, reduces the disconnection time of the mobile node from the graph. Parent change time, presented in Figure 7, shows that relying only on the RPL

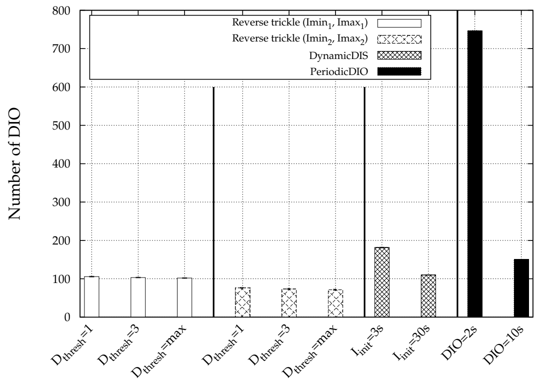

Fig. 5: Number of DIO sent - academic scenario

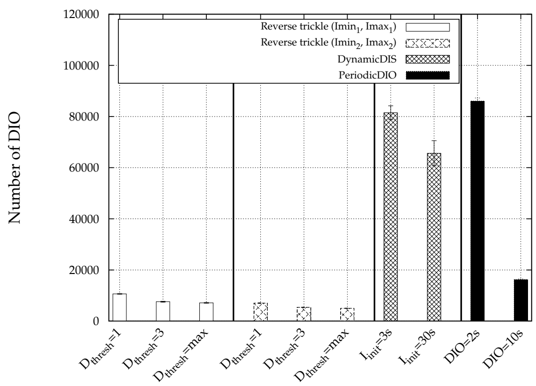

Fig. 6: Number of DIO sent - realistic scenario

mechanisms (as do DynamicDIS and PeriodicDIO) to change the parent can take a long time. As ETX needs traffic to be present in the network to be effective, the mobile node receives many DIO before changing its parent. In contrast, our solution changes the parent of the mobile node independent of metric. When *Dthresh* is exceeded, the mobile node will remove all parents. So, as soon as a new DIO is received, the mobile node will quickly choose a new parent, regardless of the advertised rank. Waiting for only *Dthresh* for a new DIO before considering that we are disconnected, limits the disconnection time to *Dthresh*. After *Dthresh*, the mobile node sends a new DIS in multicast to trigger new DIO from the neighborhood. Lower disconnection time translated to higher PDR values. Having a short disconnection time, as seen in Figure 7, enables our solution to have the highest PDR of all analysed solutions when using $Imin_1$ and $Imax_1$. This enables a quick parent change and low disconnection time from the graph. Because of the way ETX works, PeriodicDIO and DynamicDIS lose many packets as the time to change the parent is longer. Several data packets need to be exchanged by the static nodes. Our solution sends less control packets, but the mobile node remains connected longer to the graph and has higher PDR for intervals $Imin_1 - Imax_1$.

Moving on to the realistic scenario, before analysing the results, we present the assumptions made. The DODAG that RPL build needs a long time to stabilise [16]. Therefore, we started analysing our solution only after 30 min from the start of the simulation, when the DODAG will be in a stable state. When the mobile node enters the network it finds parents with paths toward the root. Control packets sent by all the nodes in the network, shown in Figure 6, give now a clearer view of how localised our solution is. The mobile node will determine an increase of the control traffic only on nodes that it connects to, not in the whole network. Using DynamicDIS, mobile nodes send DIS in multicast and reset the trickle timer for neighboring nodes, even when they are already connected to a parent. This will increase unnecessary the overall control traffic. PeriodicDIO, floods again all the network with control packets, even from nodes that will not be parents for the mobile node. As many data packets are exchanged between the static nodes and the root, ETX can easily choose a parent. Now, DynamicDIS and PeriodicDIO will allow the mobile node to change faster the parent, as seen in Figure 8. Still, as our solution does not depend on the metric to change the parent, the disconnection time in this scenario is comparable to the one obtained in the academic scenario. We show again that relying on the RPL mechanism to change the parent can take longer than our solution. In the realistic scenario we managed to have better PDR than the other two solutions. As just a few packets manage to arrive at the root from the mobile nodes, we analysed why packets are lost. We identified three main causes: no parent available for the mobile node due to handover, link state when packet is sent between the mobile node and the parent and link state on paths to the root. Our solution, for $Imin_1 - Imax_1$, loses 3.5% of packets due to handover, between 7.5-15% due to link state between the mobile nodes and the parent and 73-81% on the path from the parent to the root. When we change the values to $Imin_2 - Imax_2$, handover loss remains the same, but between the parent and the mobile nodes we loose 22-32% of packets and the rest (53-63%) on the path until the root. DynamicDIS and PeriodicDIO always provide a parent for the mobile nodes. Nevertheless, 48-52% of packets for PeriodicDIO and 55% for DynamicDIS are lost until the parent. The rest (41-44% for PeriodicDIO and 36% for DynamicDIS) are lost on the path until the root.

From the two scenarios, we have seen that our solution has better performances. We limit the disconnection time, lower the number of DIO sent, manage to achieve if not higher, than a comparable PDR, and lose less packets on the link between the mobile node and the parent. Choosing $Imin_1 - Imax_1$ and $Dthresh = 3$ give optimal performance in the network, if we consider all analysed parameters. Control traffic is limited while PDR remains high and loses are mitigated.

V. Discussion and Perspectives

In this paper, we analyse the way RPL handles mobility. Our first analysis showed that RPL already includes the mechanisms required to enable re-attachments to the graph but suffers from the poor support (or lack thereof) for rapid detection of parent unreachability in order to allow mobile nodes to reset their rank. We therefore proposed extensions to RPL in order to efficiently support mobile nodes. These extensions preserve the configuration of the network as those mechanisms are only enforced on nodes interacting with mobile nodes. Furthermore, we proposed a reverse trickle timer that allows mobile nodes to quickly detect the unreachability of their current parents in order to trigger a re-attachment to the graph.

Academic Scenario

	Reverse trickle (Imin1,Imax1)			Reverse trickle (Imin2,Imax2)			PeriodicDIO		DynamicDIS	
	Dthresh=1	Dthresh=3	Dthresh=max	Dthresh=1	Dthresh=3	Dthresh=max	DIO/2s	DIO/10s	I_{init} = 3s	I_{init} = 30s
Avg.	93.68	89.09	87.50	86.46	73.47	71.94	71.62	79.80	64.95	84.98
±	1.98	1.88	2.36	2.50	3.92	3.90	5.70	7.66	0.37	0.28
Realistic Scenario										
Avg.	8.92	9.22	8.90	9.05	9.81	8.75	8.12	6.73	8.33	8.23
±	2.38	2.79	2.12	2.58	3.38	2.66	1.60	1.15	1.70	1.73

TABLE II: Packet Delivery Ratio with 95% confidence intervals for academic and realistic scenarios

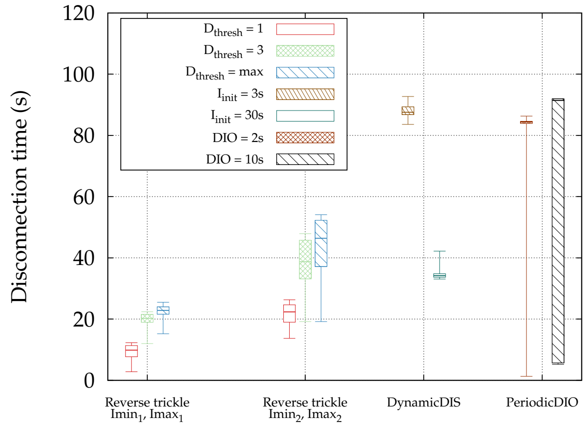

Fig. 7: Delay mobile node handover - academic scenario

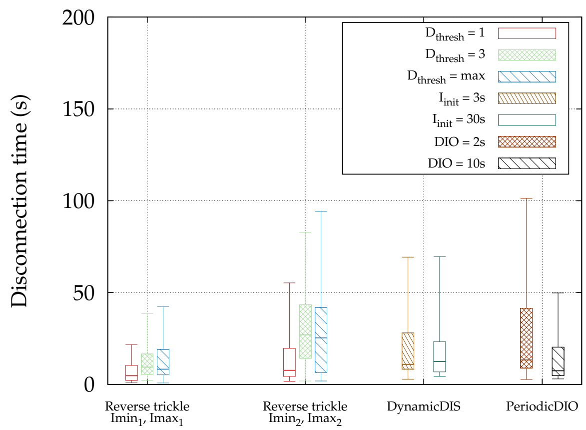

Fig. 8: Delay mobile node handover - realistic scenario

Simulations made using the popular ETX metric show that our proposal allows mobile nodes to reduce the disconnection times experienced while moving from one parent to another. Furthermore, our solution outperforms the two other proposals available in the literature in terms of traffic overhead and disconnection times thanks to our localised mechanisms and reverse trickle timer. Packet delivery ratio is higher in the majority of configurations. Packets are lost less on the link between the mobile node and the parent than in the other two proposals.

Encouraged by the results here, our future work will focus on a more precise evaluation of our proposal through more realistic scenarios. We take into consideration changing the MAC layer to diminish loss on links between nodes. We expect to benefit also from the FIT IoT testbed [17] to extend our performance studies to large-scale experiments involving multiple mobile nodes. In particular, the energy consumption will be precisely analysed, in order to represent per-node energy depletion during the whole lifetime of the network.

References

[1] V. Shnayder et al., "Sensor networks for medical care," in *Technical Report TR-08-05. Division of Engineering and Applied Sciences, Harvard University*, 2005.

[2] P. Sikka et al., "Wireless ad hoc sensor and actuator networks on the farm," in *proc. of the 5th ACM International Conference on Information Processing in Sensor Networks (IPSN)*, 2006.

[3] J. Leguay et al., "An efficient service oriented architecture for heterogeneous and dynamic wireless sensor networks," in *proc. of the 3rd International Workshop on Practial Issues in Building Sensor Network Applications (SensApp)*, 2008.

[4] J. Allred et al., "Sensorflock: an airborne wireless sensor network of micro-air vehicles," in *proc. of the 5th ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, 2007.

[5] Business Insider, "Here Comes The Internet of Things," in *BI Intelligence Report*, 2014.

[6] T. Winter et al., "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," IETF RFC 6550.

[7] J Tripathi et al., "A Performance Evaluation Study of RPL: Routing Protocol for Low power and Lossy Networks," in *proc. of the 44th IEEE Annual Conference on Information Sciences and Systems (CISS)*, 2010.

[8] K. C. Lee et al., "RPL Under Mobility," in *proc. of the IEEE Consumer Communications and Networking Conference (CCNC)*, 2012.

[9] I.E. Korbi et al., "Mobility Enhanced RPL for Wireless Sensor Networks," in *proc. of the 3rd International Conference on the Network of the Future (NOF)*, 2012.

[10] P. Levis et al., "The Trickle Algorithm," IETF RFC 6206, 2011.

[11] P. Karkazis et al., "Design of primary and composite routing metrics for RPL-compliant Wireless Sensor Networks," in *proc. of the IEEE International Conference on Telecomunications and Multimedia (TEMU)*, 2012.

[12] N. Ahmed et al., "Experimental Evaluation of Multi-Hop Routing Protocols for Wireless Sensor Networks," in *proc. of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2010.

[13] D.S.J De Couto et al., "A High-Throughput Path Metric for Multi-Hop Wireless Routing," in *proc. of the 9th ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2003.

[14] G. Jakllari et al., "Link positions matter: A noncommutative routing metric for wireless mesh networks," *proc. of IEEE Transactions on Mobile Computing*, 2012.

[15] E. Ben Hamida et al., "On the Complexity of an Accurate and Precise Performance Evaluation of Wireless Networks using Simulations," in *proc. of the 11th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2008.

[16] O. Iova et al., "Stability and efficiency of RPL under realistic conditions in WSN," in *proc. of the 24th IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, 2013.

[17] "Future Internet (FIT) - Internet of Things testbed," http://fit-equipex.fr/.