Dynamic RPL for Multi-hop Routing in IoT Applications

Harith Yahya, Hayder Al-Kashoash, Yaarob Al-Nidawi, Maria Quezada Mosquera and A.H. Kemp
School of Electronic and Electrical Engineering
University of Leeds, Leeds, UK
Email: {elhdy, ml14haak, elymna, el15mfqm, a.h.kemp}@leeds.ac.uk

***Abstract*—The Routing Protocol for Low Power and Lossy Networks (RPL) has become the standard routing protocol for the Internet of Things (IoT). This paper investigates the use of RPL in dynamic networks and presents an enhanced RPL for different applications with dynamic mobility and diverse network requirements. This implementation of RPL is designed with a new dynamic Objective-Function (D-OF) to improve the Packet Delivery Ratio (PDR), end-to-end delay and energy consumption while maintaining low packet overhead and loop-avoidance. We propose a controlled reverse-trickle timer based on received signal strength identification (RSSI) readings to maintain high responsiveness with minimum overhead and consult the objective function when a movement or inconsistency is detected to help nodes make an informed decision. Simulations are done using Cooja with different mobility scenarios for healthcare and animal tracking applications considering multi-hop routing. The results show that the proposed dynamic RPL (D-RPL) adapts to different mobility scenarios and has a higher PDR, slightly lower end-to-end delay and reasonable energy consumption compared to related existing protocols.**

I. Introduction

The Internet of things (IoT) is rapidly evolving with numerous applications, each application has its own characteristics and network requirements. Wireless Sensor Networks (WSNs) have a major role in the development of IoT and a number of technologies have already been standardized to support their integration. RPL[1] is standardized as the routing protocol of the IoT[2], it is a distance vector tree based routing protocol designed for IPv6 enabled networks, the routing tree is built as a number of Destination Oriented Directed Acyclic Graphs (DODAG) routed towards the DODAG root. Every DODAG is formed according to the defined Objective Function (OF) which determines the routing metrics that will be used for selecting the preferred parent. Many applications require some of the nodes to be mobile which creates an extra challenge to routing especially when nodes move at high speeds or in an unpredictable pattern [3][4][5]. RPL was originally designed for static networks but there are some efforts that proved it can be used for some mobile WSNs with a few alterations and enhancements[6].

Smart city applications are various and have dynamic mobility scenarios that include static and mobile nodes, some of these nodes can move in an unpredictable manor at different speeds, this type of mobility has a large impact on routing and it can significantly deteriorate the performance of the network. In order to satisfy the network requirements of applications with such a diverse mobility behaviour, it is imperative to have a dynamic routing protocol that can accommodate this kind of mobility and satisfy the demanding requirements of these applications.

To the best of our knowledge, none of the existing work on mobility enabled versions of RPL takes into account multi-hop routing through mobile nodes or the flexible interaction between the RPL timers and the objective function. Therefore, this paper is motivated by these considerations to propose an enhanced dynamic version of RPL called D-RPL with a dynamic objective function called D-OF.

In this paper, we provide realistic analysis for using RPL in mobile network based on extensive simulations for different mobility scenarios. We implement D-RPL that is an enhanced dynamic version of RPL with its own objective function (D-OF) designed for dynamic networks and compare it with existing related work taking into account different applications and mobility scenarios. The rest of the paper is organized as follows: Section 2 provides an overview of the native RPL and discusses its relevant design factors. Section 3 provides an overview of research related to using variations of RPL for mobile networks. Section 4 introduces the enhancements to RPL and the design of the D-OF using relevant metrics. Section 5 describes the simulation scenarios used to evaluate the proposed approach and provides results and analysis with regards to PDR, end-to-end delay, and energy efficiency. Section 6 presents the hardware implementation and testing for D-RPL using Tmote sky nodes MTM-CM5000-MSP. Finally, section 7 discusses conclusions and the direction of future work.

II. RPL OVERVIEW

RPL is an IPv6 routing protocol designed by the IETF ROLL working group for Low-power and Lossy Networks (LLN), it operates on the IEEE 802.15.4 standard using 6LoWPAN as an adaptation layer. RPL builds the topology of the network based on a Directed Acyclic Graph (DAG) with no outgoing edges so that no cycles can exist. Every DAG is routed towards one or more DAG roots forming a Destination-Oriented DAG (DODAG) and every DODAG has its own DODAG-ID. The DODAG is built using the predefined objective function which contains the metrics for route selection. RPL maintains connectivity using a number of control messages, The DODAG Information Object (DIO) carries information including the DODAG-ID and the rank to allow other nodes to discover the DODAG. The Destination Advertisement Object (DAO) contains the RPL instanced ID that was learnt from the DIO and it is sent from the child node to the parent node or the DODAG root. The DODAG

Information Solicitation (DIS) is used to request a DIO from an RPL node.

RPL uses the trickle timer [7] to control the frequency at which it sends DIO messages, this timer is responsible for setting the periodic timer that increases if the node's rank does not change over a threshold number of DIO transmissions, the rank of the node is depicted based on the objective function of RPL. This timer is reset if the nodes rank changes or if an inconsistency is found. The main parameters of the trickle timer are I_{min}, $I_{doubling}$ and I_{max}.

$$I_{min} = 2^n \tag{1}$$

$$Imax = 2^{n+Idoubling} \tag{2}$$

The DIO interval n produces I_{min} which is the initial and minimum interval size of the trickle timer in (ms) as shown in (1). $I_{doubling}$ decides I_{max} which is the maximum interval size of of the trickle timer in (ms) as shown in (2). The selection of these values is critical because the directly affect PDR, end-to-end delay and energy consumption of the network. High intervals lead to low responsiveness to network's inconsistencies including those caused by nodes' mobility, while low intervals mean higher overhead leading to shorter lifetime for the network.

There are currently two objective functions presented by the IETF, the first one is Objective Function zero (OF0) [8] which is a simple and basic objective function where nodes select a parent node based on its rank in the DODAG. It does not consider any other routing metrics and is designed as a general objective function that allows implementations of other objective functions. The second one is the MRHOF [9] which is based on routing metrics containers. It transmits the metric container in the DIO message and it uses the Expected Transmission Count (ETX) to calculate the rank of the node and it also supports residual energy as a routing metric.

III. Related Work

There are several efforts on investigating routing for mobile WSNs and within the IoT applications, most of the recent work is based on RPL since it became the standard routing protocol for the IoT. RPL is a flexible and scalable routing protocol and using it as a standard makes it easier to build an interoperable solution for any application making it a part of IoT.

The DAG-based Multipath Routing for mobile sensor networks (DMR)[10] was designed based on RPL with rank information and link quality identifier (LQI) as routing metrics, it uses a multipath approach with redundant routes and it has a DODAG maintenance and repair technique. However, RPL already covers these methods and while DMR outperforms the ad-hoc on-demand distance vector (AODV)[11] and the ad-hoc on demand multipath distance vector(AOMDV)[12] protocols which were not designed for LLNs and it wasn't compared to native RPL.

The authors in [13] evaluated the use of RPL in IPv6 WSNs through simulation of two case studies, the first case assumes two mobile sinks in a network of up to 40 nodes and the second case uses Power Line Communication (PLC) nodes which are not energy constrained to act as mobile sinks resulting in a better balance of the energy consumption throughout the network. Although this approach does improve the lifetime of the network, it does not add any improvement to RPL as a protocol and it does not consider other network metrics.

Similar to the last approach, the authors in [14] present a strategy for mobile sinks in IPv6 WSNs. In this strategy, every node calculates its weight based on three metrics: number of hops, residual energy and number of neighbour nodes. The sinks look for the node with highest weight and moves towards it. This approach considers only the lifetime of the network by balancing the energy consumption, it is also limited to certain applications.

A hybrid routing protocol for WSNs with mobile sinks [15] aimed to improve the parent selection in RPL by deploying one or more mobile sinks that move towards nodes with higher residual energy in a controlled manner to overcome the problem of depleting nodes closer to the sink. This protocol improves the lifetime of the network by balancing the energy usage among nodes. However, this approach does not consider metrics other than energy and it is only applicable in environments where it is feasible and efficient to have a controlled sink that moves in this manner. In addition to that, the authors do not provide simulation or practical results to validate this protocol.

In [16], the authors proposed a strategy to include the mobility status of each node in the DIO message, static nodes will be preferred in the parent selection process. This approach has a higher PDR and a better routes stability but as it includes the mobility status in the DIO message, it changes the standard and makes it no longer compatible with other versions or RPL. It is also limited in application to some mobility scenarios because it does not include any routing metrics in the parent selection process.

The authors in [17] proposed an enhanced version of RPL for vehicular ad-hoc networks VANETs. They included geographical information as a new metric in order to predict nodes in forward direction and select them as preferred parents to minimize the number of dissociations and reformation of DODAGs. They also modified the DIO timer to be addaptive to the speed of nodes in order to improve the handover time and thus improve the PDR and end-to-end delay. However, this protocol is tested only for data collection with only one cluster head that collects data from static road side nodes regardless of application network requirements and assuming the mobile node does not change direction. It is also aimed for VANET-WSNs and does not take into account a dynamic environment.

The authors in [18] proposed analysis of RPL under mobility using a reverse trickle algorithm. According to their proposal, mobile nodes are preconfigured with a mobility flag and are set to ac as leaf nodes to make sure they do not participate in the DODAG building process. When a mobile node connects to a DODAG, it sets the trickle timer to the maximum value and periodically decreases it until it reaches the minimum value or moves to another parent. Using the reverse trickle timer for mobile nodes reduces the disconnection time and improves the detection of an unreachable parent. However, this approach assumes that there is always a static node in range of any mobile node. It also requires using different settings for static and mobile nodes making it less flexible. In addition to that, this protocol has no mobility detection scheme and it rather uses different trickle settings for mobile nodes.

In [19], the authors introduced a mobility support layer called "MoMoRo" targeted at low-power WSN applications with human-scale mobility and low traffic, it allows the nodes to send probes as soon as they observe that they are disconnected from their parent node, it also introduce a destination searching scheme by sending adaptive flood messages to detect a missing node in the data collection tree. According to the simulation results, this protocol achieves similar PDR when compared to the native RPL and to the AODV, it has less packet overhead than AODV but slightly more than the native RPL. In an outdoor practical test using three mobile nodes and one collection node, the PDR is similar to that of AODV with less packet overhead. However, this protocol cannot accommodate nodes that moves at higher speeds or require high amounts of traffic. In addition to that, the practical experiment is done using only three mobile nodes which cannot effectively show realistic results in a general manner.

The authors in [20] introduced a corona mechanism with RPL (Co-RPL) for two main enhancements to the protocol, the first one is based on the corona principle in which the network is divided into circular coronas around the DODAG root, this principle allows the nodes to find an alternative parent in a faster manner without needing to reform the DODAG, the second enhancement is the fuzzy logic objective function FL-OF that uses end-to-end delay, hop count, link quality and residual energy as routing metrics. This protocol achieves higher PDR, less end-to-end delay and better energy than the native RPL. However, this protocol is designed for nodes moving at low speeds of up to 4 m/s and it does not address a hybrid network with a dynamic mobility model.

Another enhancement of RPL designed for healthcare and medical applications [21] presents an evaluation of RPL for hybrid networks with both mobile and static nodes within the applications of healthcare. The authors do not introduce any enhancement to the RPL itself but rather force mobile nodes to act as leaf nodes which according to the RPL specifications cannot advertise themselves as routers and do not send DIO messages with the objective function metrics. This approach improves the stability of the network by allowing the mobile nodes to connect to the DODAG but not to act as a parent node nor to participate in the formation of the DODAG. The problem with this approach is that it assumes that there is always a fixed node in range of any other node, it also does not add anything to the design of RPL but rather evaluates using it within the given scenario.

In [22] the authors propose a mobile version of RPL called mRPL to manage mobility in IoT environments. This protocol aims to improve the hand-off time for mobile nodes by adding four timers to the original trickle algorithm in order to detect disconnected nodes in a smart and fast approach. The connectivity timer is responsible for detecting a loss of connectivity to the parent node. The mobility detection timer uses the average received signal strength indication (ARSSI) to assess the reliability of the connection. The hand-off timer is responsible for allocating an adaptive short period that is sufficient for sending bursts of DIS and receiving DIO replies in order to reduce the hand-off delay. The reply timer is responsible for sending replies to the mobile nodes using an adaptive period to minimize collision. This protocol is compared with the native RPL considering different simulation scenarios and the results show that mRPL outperforms the native RPL in terms of PDR, packet overhead and hand-off delay. A practical test is also conducted using Tmote-Sky nodes and the results were similar to the simulation. However, mRPL relies heavily on ARSSI values and neglects other metrics resulting in unnecessary hand overs and sometimes unreliable links establishment. This protocol is tested for only one mobile node moving at a constant velocity (2m/s) near nine static nodes and does not consider more than one mobile node or nodes moving at higher speeds. It also does not discuss the objective function of RPL and its potential to improve mobility management.

More recently, a "Smarter-HOP" version of mRPL for optimizing mobility in RPL was introduced to improve the performance of mobility management. This protocol is named mRPL++ [23] and it includes the objective function in the parent selection process to make sure that nodes are aware of link metrics other than RSSI. This approach improves the decision making by using the product of ARSSI and the ratio between the metric costs in the objective function of the competing parent nodes as the basis for parent selection. However, this protocol still suffers from the weakness points of mRPL and is still dependant on RSSI so that it cannot be neglected regardless of the objective function.

The authors in [24] present a routing strategy called Kalman positioning RPL (KP-RPL), this protocol is based on RPL and it provides robust routing for WSNs with both static and mobile nodes. In KP-RPL, two modes of communication are defined, the anchor to anchor (two static nodes) and the mobile to anchor. The first mode uses the default RPL while the second one is managed by using Kalman filter and blacklisting. Each mobile node creates an initial list of the static nodes within its range and according to the Received Signal Strength Identifier (RSSI), it blacklists those of low ETX that are considered "potentially unreliable links". This approach improves the reliability of the network by 25% according to simulation results. However, it assumes only one mobile node is moving within range of a number of static nodes and does not take into account additional mobile nodes. It also relies on positioning to estimate the position of the mobile node and performs blacklisting based on that. Inaccurate positioning can result in severe network degradation because not only the routing decision will be affected but also reliable links might be blacklisted.

IV. D-RPL Description

The IoT covers a wide range of applications using different standards and technologies to serve a large number of applications. These applications have different network requirements, different node distribution and different mobility scenarios. D-RPL is designed for networks where nodes can be attached to people or objects building a dynamic mobility scenario in which the DODAG formation can involve multiple mobile nodes. In this paper, healthcare and animal tracking are presented as realistic IoT applications with dynamic mobility scenarios that require multi-hop routing to the root or gateway through mobile nodes.

The design of D-RPL includes improvements to the RPL trickle timer, a new objective function and the interaction between these two factors to manage mobile nodes in the network and improve the performance of RPL routing.

A. Timers

RPL relies on the trickle timer in sending DIO messages, if the network is stable this timer will increase exponentially to limit the number of control messages and keep a low overhead. When an inconsistency is discovered this timer is reset to I_{min} in order to recover and repair the lost links. In D-RPL we add a control mechanism for the interval of the trickle timer based on the reception of data packets and control packets.

```
Begin
Initialize trickle timer;
if Received a packet from node n then
    Read $RSSI_n$;
    if $RSSI_n$ + $K_{RSSI}$ <$lastRSSI_n$ then
        $Trickle_I$ = ($OldTrickle_I$ / 2);
        if $Trickle_I$ <$I_{min}$ then
            $Trickle_I$ = $I_{min}$;
        end
        Send DIS to all neighbours;
    else
        Resume normal trickle algorithm;
    end
end
```

Algorithm 1: Trickle Timer in D-RPL

Upon receiving a packet from node n, nodes read the $RRSI_n$ and compare it to the last reading from the same node $lastRSSI_n$. If the new reading is lower by a redundancy constant K_{rssi} it switches to the reverse-trickle setting and decreases the current interval to half until it reaches I_{min}. It also sends a DIS to all neighbours to assess the available options, otherwise it resumes the native RPL mechanism. This is based on the fact that a moving node is not necessarily going to leave its parent node and the decision on whether to switch to a new parent is left to the objective function. The trickle timer operation in D-RPL is defined in pseudo-code 1.

B. The Objective Function

The proposed dynamic objective function D-OF utilizes the Minimum Rank with Hysteresis Objective Function (MRHOF) that is already available in Contiki OS, and it adds other metrics in the calculation of the path cost to the destination. These metrics include ETX which is based on the expected number of transmissions required to send a packet from source to destination, the energy metric is used as the estimated energy required to send a packet to the destination and the link quality indicator (LQI) which is based on the RSSI. The MRHOF objective function defines a threshold for switching to a new parent, nodes only switch if the rank difference is more than 1. However, in D-OF more than one metric is used to produce the cost and changing the threshold is necessary to minimize the number of unwanted hand-overs and improve the routing performance.

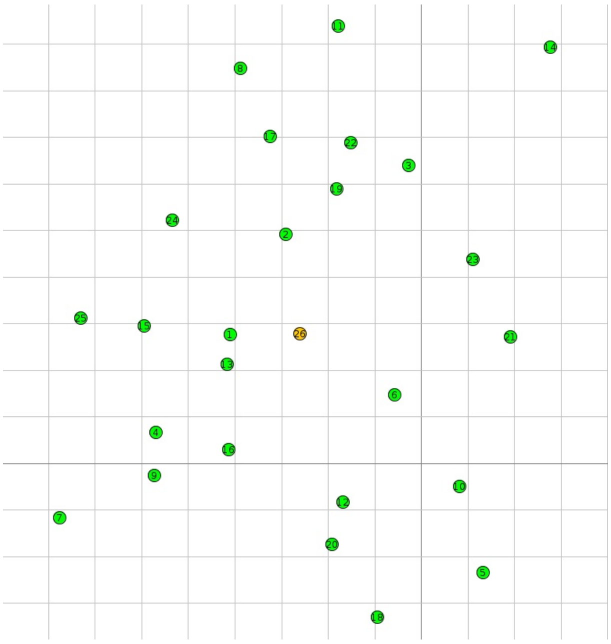

Fig. 1. Node Distribution

The proposed RSSI-based reverse-trickle timer mechanism in D-RPL aims to reduce the hand-over delay by sensing $RSSI$ values and detecting mobility or inconsistency while the proposed objective function D-OF which is responsible of parent selection aims to reduce the number of unnecessary hand-overs by comparing the calculated cost to the parent switching threshold. The integration of D-RPL and D-OF creates an optimization of these two crucial factors making it an adaptable solution for dynamic IoT applications.

V. Simulation Results and Analysis

A. Simulation Setup

The implementation and simulation of D-RPL has been done using Contiki operating system 3.0 [25], with COOJA [26] WSN simulator. Cooja has a mature and reliable implementation of RPL and although it does not normally support node mobility, it can import the coordinates of nodes through a mobility plug-in to represent mobile nodes. Mobility scenarios are generated using Bonnmotion [27], a free and widely used mobility scenario generation tool. Two different scenarios are generated to test the proposed D-RPL and compare it with relevant protocols.

We used 25 mobile nodes and 1 static sink node in a 150m x 150m simulation area as shown in Fig 1. These nodes move based on the random waypoint mobility model at 0-2 m/s and 0-5 m/s for Healthcare and Animal tracking scenarios respectively with a maximum pause of 30s. The values of I_{min} and $I_{doubling}$ are chosen to be 8 and 6 respectively giving a minimum interval of 256ms and a maximum interval of 16s as shown in Table I.

TABLE I. SIMULATION PARAMETERS

Parameter	Value
Simulation Area	150m x 150m
Number of Nodes	25 mobile nodes + 1 sink node
Transmission Range	50m
Healthcare Scenario	Random Waypoint, 0 to 2 m/s
Animal Tracking Scenario	Random Waypoint, 0 to 5 m/s
I_{min} / $I_{doubling}$	8 / 6
Simulation Time	1 hour
Radio	CC2420

B. Simulation Results

In order to test the performance of D-RPL, we chose three metrics that reflect the efficiency of the network. These metrics are end-to-end delay, energy consumption and PDR. The end-to-end delay represents the average time required for each node to successfully send a packet from source to destination. Energy consumption represents the average amount of energy consumed to successfully transmit a packet from source to destination at each node during 60 minutes of simulation. PDR shows the percentage of delivered packets from each node compared to the total number of packets sent by the same node.

1) Healthcare Application: Healthcare is one of the most important IoT applications because it aims to improve patients' experience and potentially save lives. In this application, we assume that low-powered mobile nodes are attached to people, objects and equipment in a healthcare establishment and thus we consider a maximum speed of 2m/s which corresponds to human speed and can also be applied for other IoT applications like smart cites and smart factory management.

Fig 2 shows the percentage of the number successfully transmitted packets compared to the total number of sent packets. mRPL has high responsiveness to mobility and simulation results show that it provides an average PDR of 75% which is much higher than the Native RPL but around 10% lower than D-RPL. This is because mRPL was designed on the assumption that there is always a static node in range of every mobile node, however in a dynamic scenario with multi-hop communication through mobile nodes, it performs some unnecessary hand-overs causing a loss of successfully delivered packets. D-RPL gives a PDR of around 84% using the adaptive trickle technique and its integration with the objective function.

RPL was originally designed for static networks and thus it has low responsiveness to topology changes and it has an average of 36% PDR in this scenario.

Fig 3 shows the average energy consumption per successfully transmitted packet at each node after 60 minutes of simulation, it shows that D-RPL performs better than mRPL and much better than the native RPL. This is due to multiple factors including the fact that mobility triggers the trickle timer to reset to its minimum value in both RPL and mRPL, while D-RPL detects mobility based on RSSI readings and this triggers a decrease in the DIO interval instead of resetting it to the minimum value and only resets it when the link is broken or no longer reliable according to the D-OF. Another factor is

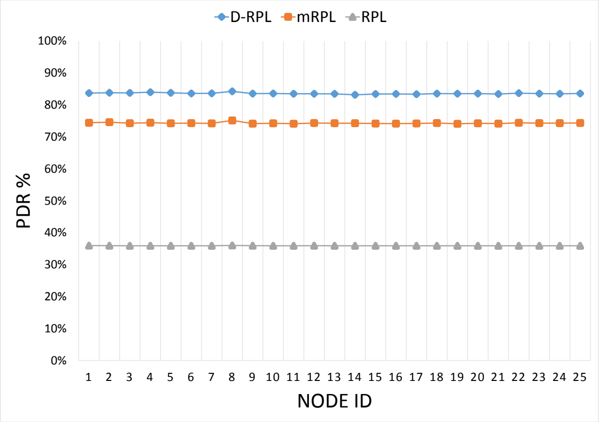

Fig. 2. PDR - Healthcare

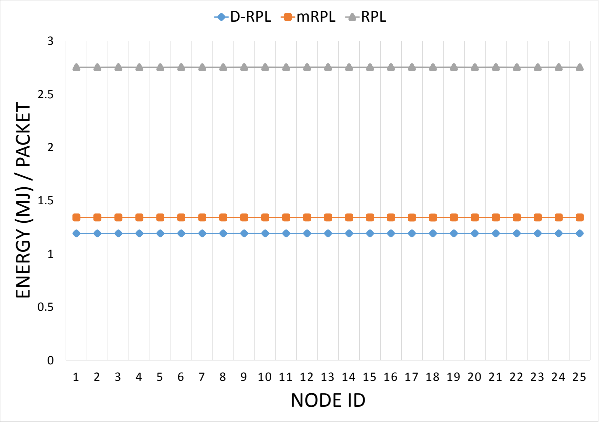

Fig. 3. Energy Consumption - Healthcare

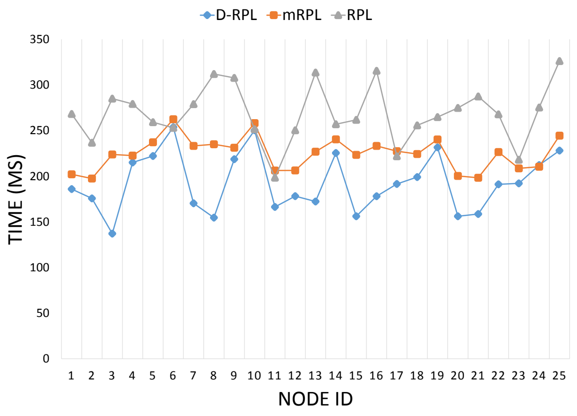

Fig. 4. End-to-End Delay - Healthcare

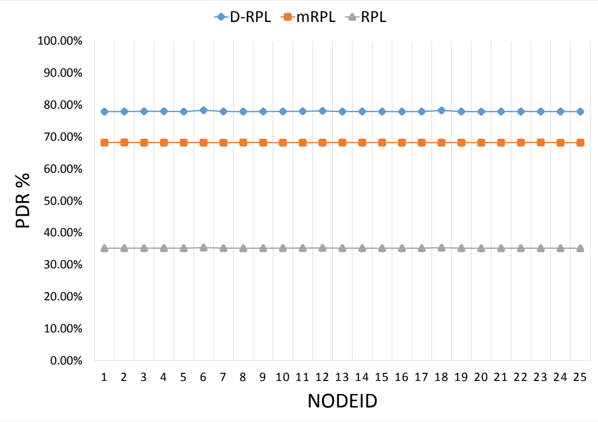

Fig. 5. PDR - Animal Tracking

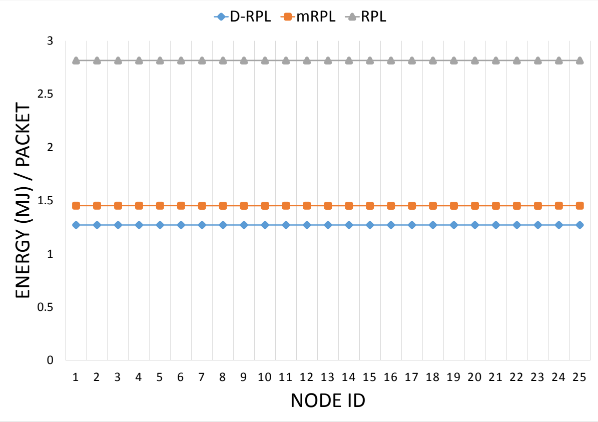

Fig. 6. Energy Consumption - Animal Tracking

the higher packet loss in mRPL and RPL that leads to more retransmissions and higher energy consumption.

The end-to-end delay in Fig 4 is similar in all three protocols with marginal difference. D-RPL performs slightly better for most of the nodes because of the better decisions in parent selection. Although all three protocols are using the same objective function, the operation of D-RPL is more compatible than mRPL and the native RPL leading to less delay from source to destination. This metric is based only on the successfully delivered packets and does not take into account the dropped packets and so it does not reflect the efficiency of routing unless incorporated with PDR.

2) Animal Tracking Application: This application is another IoT application that aims to track a herd or a pack of animals and provide information about not only the animals themselves but also their surrounding environment. Having nodes attached to animals can cover a larger area due to their mobility. This application can also be used to detect fires in a forest or a field. We chose this application because it has a dynamic mobility scenario with nodes moving at high speeds. It can also reflect the challenges of applications with similar mobility scenarios like sports monitoring.

Simulation results in Fig 5 shows that D-RPL and mRPL adapt to the high mobility and provide reasonable results of around 78% and 68% PDR respectively. While the native RPL fails to catch up and provide only 35% average PDR. Similar to the healthcare application, although mRPL responds to inconsistencies quicker than D-RPL it still relies on the presence of static nodes in range and thus generates extra overhead and makes unwanted hand-overs that lead to packet loss.

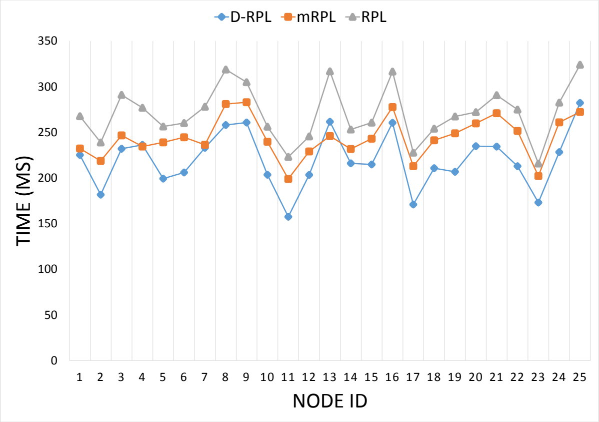

Fig. 7. End-to-End Delay - Animal Tracking

Fig 6 shows that RPL has the highest energy consumption per packet because of the very high packet loss caused by its low responsiveness to mobility. The performance of mRPL is much better than RPL but still fails to catch up with D-RPL because in addition to higher packet loss, the high mobility makes its trickle timer act as a periodic timer and generates high overhead. The trickle timer in D-RPL also acts more like as a periodic timer but at higher intervals that are adaptive to the speed of mobile nodes and thus has the lowest energy consumption.

The end-to-end delay in this scenario shows that RPL, mRPL and D-RPL have similar results for the successfully transmitted packets as shown in Fig 7. Taking PDR into account shows that D-RPL provides a higher routing efficiency and a more reliable solution.

VI. Practical Testing

In order to test the real performance of D-RPL, we conducted hardware testing using 10 Tmote sky nodes MTM-CM5000-MSP shown in Fig 8. The experiment was conducted in 2 environments, an obstacle-free open field and an indoor environment with obstacles. A simulation scenario is also created for comparison using a similar topology to the real hardware experiments.

The testing scenario involves 1 static sink node and 9 mobile nodes moving at 0 - 1.5 m/s. Mobile nodes are connected to people moving at normal human speeds and pausing for a maximum period of 30s. Nodes are placed with minimum overlapping to ensure multi-hop communication. The sink node with ID 1 as shown in Fig 9 is the only static node in the network, other nodes move randomly to force topology changes.

The results in Fig 10 show that RPL achieves around 42% PDR while mRPL and D-RPL achieve around 88% and 90% respectively in simulation and both practical tests. The

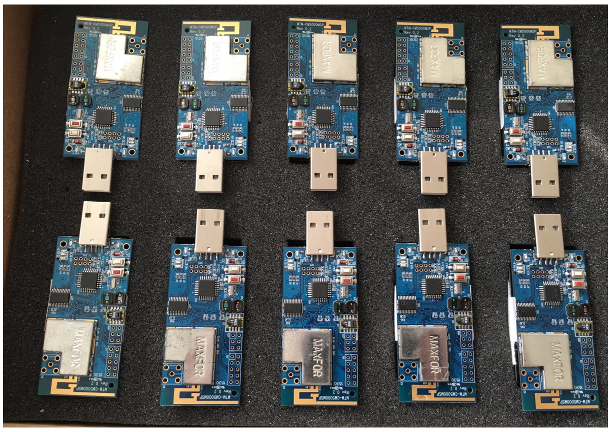

Fig. 8. Tmote sky nodes MTM-CM5000-MSP

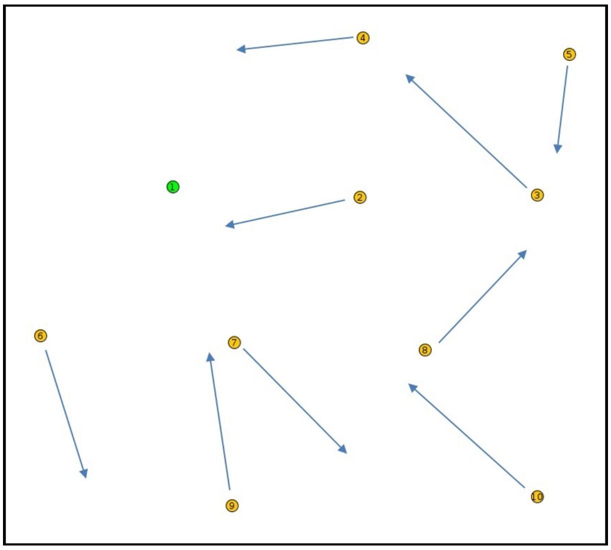

Fig. 9. Hardware Testing Scenario

lower density gives the objective function less options making the difference in performance of mRPL and D-RPL down to 2% only. Higher node density increases the chance of collisions and leads to higher packet loss due to interference and congestion [28].

D-RPL depends on data packets as well as control packets to manage mobility making it adapt to topology changes. It is also less prune to inaccurate RSSI readings because it involves the objective function metrics in the parent selection process.

The practical and simulation results are almost the same in spite of the external factors that are expected to affect practical testing. This confirms that Cooja is successful in emulating the actual hardware and providing a realistic channel model.

VII. Conclusion and Recommendations

In this paper, D-RPL is implemented for the dynamic applications of IoT to accommodate the network requirements and mobility demands of these applications, it is based on and compatible with RPL making it a flexible and scalable solution. Simulation results show that D-RPL improves the PDR, end-to-end delay, and energy efficiency of the network for different mobility scenarios.

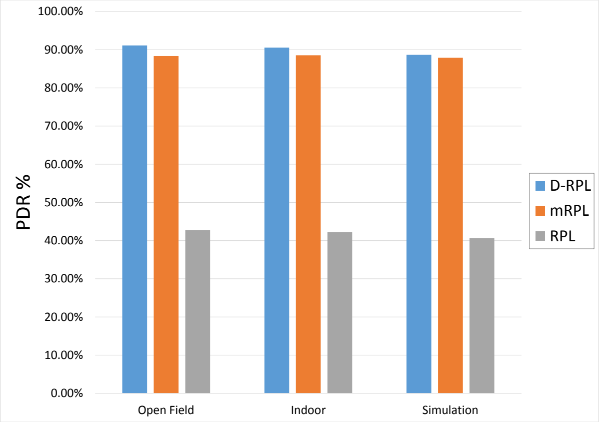

Fig. 10. Practical Test

D-RPL shows that it adapts to mobility changes better than relevant RPL-based protocols, achieving more than 10% improvement to PDR with better end-to-end delay and better energy consumption compared to mRPL. Simulation results also show the importance of the objective function and its impact on mobility management in RPL. The proposed objective function D-OF complements the operation of D-RPL giving reliable performance and efficient routing mechanism.

The design of D-RPL makes it adapt to other objective functions as well because it does not imply any metrics without consulting the objective function and uses RSSI only to detect mobility and not to make a final decision. Using the RSSI-based reverse-trickle algorithm in D-RPL leads to similar responsiveness to mRPL in low density networks. Including the objective function metrics improves the performance of D-RPL making it more efficient in highly dynamic scenarios. The optimization of the objective function to improve mobility management is essential to achieve higher network performance. It's worth mentioning that using mRPL or mRPL++ give better results in scenarios where it is feasible to have static nodes are in range of all mobile nodes. Further optimization and testing for the objective function parameters is still required to improve mobility management in dynamic networks.

References

[1] T. Winter, "Rpl: Ipv6 routing protocol for low-power and lossy networks," 2012.

[2] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, "Standardized protocol stack for the internet of (important) things," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 3, pp. 1389–1406, 2013.

[3] Y. Al-Nidawi, H. Yahya, and A. H. Kemp, "Tackling mobility in low latency deterministic multihop ieee 802.15. 4e sensor network," *IEEE Sensors Journal*, vol. 16, no. 5, pp. 1412–1427, 2016.

[4] H. Yahya, Y. Al-Nidawi, and A. H. Kemp, "A dynamic cluster head election protocol for mobile wireless sensor networks," in *2015 International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2015, pp. 356–360.

[5] Y. Al-Nidawi, H. Yahya, and A. H. Kemp, "Impact of mobility on the iot mac infrastructure: Ieee 802.15. 4e tsch and lldn platform," in *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*. IEEE, 2015, pp. 478–483.

[6] K. C. Lee, R. Sudhaakar, L. Dai, S. Addepalli, and M. Gerla, "Rpl under mobility," in *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*. IEEE, 2012, pp. 300–304.

[7] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The trickle algorithm," *Internet Engineering Task Force, RFC6206*, 2011.

[8] P. Thubert, "Objective function zero for the routing protocol for low-power and lossy networks (rpl)," 2012.

[9] O. Gnawali and P. Levis, "The etx objective function for rpl," 2010.

[10] K.-S. Hong and L. Choi, "Dag-based multipath routing for mobile sensor networks," in *ICT Convergence (ICTC), 2011 International Conference on*. IEEE, 2011, pp. 261–266.

[11] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing," Tech. Rep., 2003.

[12] M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing," *Wireless Communications and Mobile Computing*, vol. 6, no. 7, pp. 969–988, 2006.

[13] L. B. Saad, C. Chauvenet, and B. Tourancheau, "Simulation of the rpl routing protocol for ipv6 sensor networks: two cases studies," in *International Conference on Sensor Technologies and Applications SENSORCOMM 2011*. IARIA, 2011.

[14] L. B. Saad and B. Tourancheau, "Sinks mobility strategy in ipv6-based wsns for network lifetime improvement," in *New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on*. IEEE, 2011, pp. 1–5.

[15] V. Safdar, F. Bashir, Z. Hamid, H. Afzal, and J. Y. Pyun, "A hybrid routing protocol for wireless sensor networks with mobile sinks," in *Wireless and Pervasive Computing (ISWPC), 2012 7th International Symposium on*. IEEE, 2012, pp. 1–5.

[16] I. Korbi, M. Ben Brahim, C. Adjih, and L. A. Saidane, "Mobility enhanced rpl for wireless sensor networks," in *Network of the Future (NOF), 2012 Third International Conference on the*. IEEE, 2012, pp. 1–8.

[17] B. Tian, K. M. Hou, H. Shi, X. Liu, X. Diao, J. Li, Y. Chen, and J.-P. Chanet, "Application of modified rpl under vanet-wsn communication architecture," in *Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on*. IEEE, 2013, pp. 1467–1470.

[18] C. Cobarzan, J. Montavont, and T. Noel, "Analysis and performance evaluation of rpl under mobility," in *2014 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2014, pp. 1–6.

[19] J. Ko and M. Chang, "Momoro: Providing mobility support for low-power wireless applications," *Systems Journal, IEEE*, vol. 9, no. 2, pp. 585–594, 2015.

[20] O. Gaddour, A. Koubâa, and M. Abid, "Quality-of-service aware routing for static and mobile ipv6-based low-power and lossy sensor networks using rpl," *Ad Hoc Networks*, vol. 33, pp. 233–256, 2015.

[21] F. Gara, L. Ben Saad, R. Ben Ayed, and B. Tourancheau, "Rpl protocol adapted for healthcare and medical applications," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International*. IEEE, 2015, pp. 690–695.

[22] H. Fotouhi, D. Moreira, and M. Alves, "mrpl: Boosting mobility in the internet of things," *Ad Hoc Networks*, vol. 26, pp. 17–35, 2015.

[23] M. R. Anand and M. P. Tahiliani, "mrpl++: Smarter-hop for optimizing mobility in rpl," in *2016 IEEE Region 10 Symposium (TENSYMP)*. IEEE, 2016, pp. 36–41.

[24] M. Barcelo, A. Correa, J. L. Vicario, A. Morell, and X. Vilajosana, "Addressing mobility in rpl with position assisted metrics," *IEEE Sensors Journal*, vol. 16, no. 7, pp. 2151–2161, 2016.

[25] A. Dunkels *et al.*, "The contiki operating system," *Web page. Visited Oct*, vol. 24, 2006.

[26] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*. IEEE, 2006, pp. 641–648.

[27] C. de Waal and M. Gerharz, "Bonnmotion: A mobility scenario generation and analysis tool," *Communication Systems group, Institute of Computer Science IV, University of Bonn, Germany*, 2003.

[28] H. A. Al-Kashoash, Y. Al-Nidawi, and A. H. Kemp, "Congestion-aware rpl for 6l0wpan networks," in *2016 Wireless Telecommunications Symposium (WTS)*. IEEE, 2016, pp. 1–6.