

A RPL based adaptive and scalable data-collection protocol module for NS-3 simulation platform

Y. Chen, K.M. Hou, Jean-Pierre Chanet, Khalid El Gholami

▶ **To cite this version:**

Y. Chen, K.M. Hou, Jean-Pierre Chanet, Khalid El Gholami. A RPL based adaptive and scalable data-collection protocol module for NS-3 simulation platform. NICST 2103 New Information Communication Science and Technology for Sustainable Development: France-China International Workshop, Sep 2013, Clermont-Ferrand, France. 8 p. hal-00878089

HAL Id: hal-00878089
https://hal.science/hal-00878089

Submitted on 29 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A RPL based Adaptive and Scalable Data-collection Protocol module for NS-3 simulation platform

Yi-bo CHEN [1], Kun-Mean HOU [2], Jean-Pierre CHANET [1], Khalid El Gholami[2]

[1] *UR TSCF, IRSTEA*
Aubière, France
[1] yibo.chen@irstea.fr
[3] jean-pierre.chanet@irstea.fr

[2] *LIMOS Laboratory UMR 6158 CNRS, Blaise Pascal University Clermont-Ferrand II*
Clermont-Ferrand, France
[2] kun-mean.hou@isima.fr
[4] khalid.elgholami@isima.fr

Abstract—**This paper presents data-collection protocol framework based on RPL (IPv6 Routing Protocol for Low Power and Lossy Networks) for NS-3 (Network Simulator 3) simulation platform. Its design, implementation, simple examples of operations and evaluations will also be demonstrated. The conclusions and future developments are located in the final part of this paper.**

I. INTRODUCTION

The wireless infrastructure of Internet/Web of Things (I/WoT) is formed by hundreds or thousands of low energy wireless interconnected sensing devices. Although some concepts and terminology have changed, the constraints of traditional Wireless Sensor Networks (WSNs) still exist and bring challenges to solve the problems, such as designing routing protocol for an effective deployment of sensor nodes. In this context, a working group called Routing over Low Power and Lossy Networks (ROLL) was formed by the Internet Engineering Task Force (IEFT) to develop an adaptive and scalable routing solution. After 4 years of its growth, RPL became a standard and adopted by the IETF in 2012 [1].

RPL is a completely new routing protocol especially adapted and tailored for all Low Power Networks (The network based on IEEE 802.15.4 WPAN, Bluetooth, low power Wi-Fi or Power Line Communication) communicating in IPv6 [2]. We believe that this routing protocol could help I/WoT to become a reality since it is not only supporting WSN applications, but also the other specific requirements given by future customers and markets.

As we know, for a better understanding and evaluation for any routing algorithms, protocol simulation is a good start. However, it is not an easy task because the dynamic parameters like energy, packet delivery ratio and transmission delays will influence how the routing protocol optimizes the routes. These factors have to be taken into account. Otherwise, the simulation results will be far from the meaningful ones, and say nothing of the results equivalent with a real-world implementation. A routing protocol like RPL, which needs a full set of protocol stack supporting and combines a number of algorithms, the developers may find that it is not easy to design a suitable simulated environment for RPL and valid all of features from its standards [1]. We believe that this is the reason why the current implementations, such as those similar ones for Omnet++ Castalia [3], Contiki/Cooja [4] and TinyOS/TOSSIM [5], are quite limited and difficult to make the extensions at the moment. A RPL model for NS-3 is proposed in [6], however, this projects has not been released as open source, and cannot be accessed yet.

As per our best knowledge, NS-3 is a trustful and promising simulator, and it has equipped a full IPv6 supporting stack model, while we have a reliable and practicable WPAN model [7] with 6LoWPAN adaptation function to be a good foundation. Thus, we decide to propose our RPL-based data-collection routing protocol in this platform. This new model could allow us to achieve the following targets:

- A better understanding of RPL protocol standards and its basic operations.
- Making our efforts to define the advantage of RPL in the general data-collection application. By using the flexible multi-interface function of NS-3, a simulated edge router with Wi-Fi, Ethernet and 802.15.4 MAC interfaces could be more similar to the required date sink for the future LLNs application.
- Proposing a simplified and adapted RPL module but sufficient for the future applied requirements (e.g. in this paper, obviously is data-collection) could reduce its relative complexity.
- To make the experiments focused on a portion of RPL features instead of considering a whole standard.

The paper is organized as follows: in section two we introduce the RPL routing protocol in a brief way, in section three we discuss about the related work of this paper, in section four we describe the protocol model for NS-3 including its design and implementation, in section five we show the evaluation results, and final part covers the next steps work and a conclusion.

II. OVERVIEW OF RPL ROUTING PROTOCOL

As the core of this data-collection protocol module, the RPL will be presented in this section. RPL is a proactive routing protocol exclusively for IPv6 architecture, and can interact with topology changes to optimize the routes through a certain metric or a combination of different metrics. Moreover, RPL was designed and developed from four main foreseen applications of LLN and their requirements, such as Home automation, building automation, industrial and urban environments [5]. In the article [8], we also propose another use case for RPL in the area of precision agriculture. Although it is able to settle more various environments in addition to these documents have highlighted, the data gathering/collection is always the main use for any LLNs and the precondition for later processing or decision. The data distribution (or two way communication) is rarely to be used and is not often required by sensor networks, and it is often adopted only when some controlling messages need to be sent.

RPL supports three types of traffic: Point to Point (P2P), Point to Multipoint (P2MP) and Multipoint to Point (MP2P). Moreover, it has four Modes of Operation (MOP) for adapting more kinds of RPL nodes which might have specific constraints. The four MOPs are: 'No downward routes', Non Storing, Storing, Storing with Multicast support. The differences will be clear in the following [1].

A. *DODAG Building strategy*

The base concept proposed for RPL is the Destination Oriented Directed Acyclic Graph (DODAG). The idea of DODAG is simple and straightforward since if a DODAG has been built, all the data flow can attain the edge router, or spread to a specific leaf node.

ROLL working group maintains RPL keeps its simplicity and flexibility all the time. After RPL became a standard, the base operations and the messages format are settled. However, the DODAG building solution is still becoming more and more complex. Multi-DODAG with multi-roots is one issue. The link quality changes cause by mobility or dynamic environment is another issue.

The base operation of RPL is: when a node is not connected to any DODAG it will try to send DODAG Information Solicitation (DIS) to a special IPv6 multicast group address. Any other node which is already in a grounded DODAG can send back a DODAG Information Object (DIO). The node will choose the more suitable node as parent (next hop to root) and will join its parent's DODAG.

A DODAG needs to be determined by an Objective Functions (OF) [1, 9] and each OF will contain a metric [10]. The metric has diverse form like it can be used for additive, multiplicative, min or max. Furthermore, the number of hops, the link quality, the node's energy and so on can be involved even combined to a OF. This mechanism brings a very flexible solution to calculate the rank which is a new way to define the relative position of a node in the DODAG.

B. *RPL loop prevention and repair*

Loop will cause serious issue for any routing protocol. RPL offers a prevention mechanism during the DODAG building. More detail content is described in the RPL full specification [1]. From the defined rank for each node, a simple way is to ensure that every node has a rank greater than its preferred parent. However, there are some rare cases, such as those loops occur because of the delays in the DODAG update. RPL also has a loop detection mechanism to process this situation and to completely prevent loops. Sometimes, RPL will make a decision to rebuild the whole DODAG if a loop or broken upward link problems happen. RPL also provide an optional feature to repair a broken DODAG, which is to choose another parent from parent set. There is no perfect solution and the trade-off has to be made by developers.

C. *Downward routes*

Based on the specification, the DODAG can handle all the upward MP2P traffic because a node willing to send a packet 'up' has just to send it to its parent as a next hop. However, downward routes (support M2MP and P2P) are rather different RPL MOPs. Downward routes are maintained by Destination Advertisement Object (DAO) sent from each node to their parents. The DAO will contain all the IPs the node can route including its own IP and all the IPs of the nodes belonging to the sub-DODAG of this node.

- If the DODAG MOP is 0, ('No downward routes'), DAOs will not be sent, and the root will usually use a flooding in a scope address to find the downward routes and reach the node.
- If the DODAG MOP is 1 'Non Storing', meaning that the DAOs will be sent, but the nodes will not store these downward routes. The root will have to work like an edge router or a gateway, especially when the P2P traffic is required.
- If the DODAG MOP is 2 and 3, 'Storing' and 'Storing with multicast', mean that each node can act as a router, which is similar to a ZigBee Full Function Device (FFD). This will enable the packets transmission between two unconnected sub-DODAGS of each node.

In our implementation, we decided to focus on MOP 0, as it is the most simplified one. Moreover, the current hardware with constrained resource will prefer MOP 0, and we can utilize the extra computing power and storing ability for the upper layer protocol, such as CoAP and its applications [11].

III. RELATED WORK

In this section, we present our investigation work related to the RPL protocol.

A. *LLN and WSN*

From the conceptual views, a WSN is an LLN. However, an LLN may be not a common WSN because not all LLN nodes have to use wireless communication. Furthermore, the ROLL working group presents more distinguishing features for LLN in its charters, such as LLN should provide optimization for saving energy and it may be used over a

variety of link layers [12]. LLN's characteristics import more issues like energy starvation, very limited memory or CPU, high frequency of topology changes, large number of nodes, exclusive IPv6 supporting, which may not have in the traditional WSN. Under this contexts and requirements, ROLL working group proposed RPL and focused on routing issues for LLN.

B. Existing RPL simulations

Any routing protocols cannot leave simulation and experimental models, and several works and papers have been proposed for the experimenting with RPL and evaluating its performance.

The first official simulation model draft is built on OMNET++ [3]. This model can be used for basic performance evaluation study of RPL, such as the MP2P and P2P routing. The authors found the position of the root node will influence the simulation results. In addition, they demonstrated the efficiency of the trickle timer in controlling the packet overhead and stabilizing the network. In their small-scale network, the simulation shows the significant effect of control packet overhead caused by the global repair procedure.

The author of [13] tackled the issue of the possible appearance of routing loops which may happen during the procedure of nodes' parents changing and ranks increasing. Their team developed a simulation model of RPL under NS-2 simulator. This model can simulate a large-scale network composed of 1 to 1000 nodes deployed in a large space. In this paper, through the network convergence time and routing messages overhead performance evaluation, the author proved that the majority of the routing loops are resolved in a very short time and just lead to the generation of quite few DIO messages. But in some special cases, when one node carries out the rank increase operation, it may trigger and generate multiple routing loops so the network needs more time to converge to a stable loop-free state. The related node will of course produce a significant number of DIO messages. Thus, the authors found out the default loop avoidance mechanisms are not practical for large-scale RPL networks. This team also presented the RPL P2P routing algorithm in another paper with the same simulation platform.

The author of [14-15] adopted Cooja platform to evaluate the network overhead, the throughput and the end-to-end delays for various network sizes. The authors reported the network set-up time, the influence related to the DAO messages and the enhancement of control overhead of RPL, and also pointed out that RPL is still open to further improvement to optimize in facets of convergence time and control overhead in different applications.

After our initial simulation work in [8], we found that these simulation models can provide insights on the RPL protocol behavior appropriately. But due to their dependence of emulated channel models, they cannot report the exact performance of RPL because the environments are different from real channels, and their emulators of hardware resources are also not dependable. Thus, the experimental models are very important to assess the real protocol behavior and performance.

ContikiRPL is the first implementation of RPL for real-world devices. It is belonged to Contiki OS as a built-in component routing protocol. The platform of Contiki is quite comprehensive including simulation, experimentation, and evaluation of RPL's mechanisms and behaviors. It even provides a simple programming interface for designing and evaluating new Objective Functions for ContikiRPL. TinyRPL is another real-world implementation of RPL. It is based on TinyOS 2.x and BLIP. In paper [16], the authors evaluated the performance of RPL through real experimentation using TelosB motes and compared with the CTP protocol. These two RPL implementations have interoperability to some extent, and have to been tested although the results showed that there was still obvious limitation.

C. From 6LoWPAN routing protocol to RPL

In the article [17], the authors made a survey of the paradigm shift in WSN routing protocols in a chronological organization way to classify the protocols from early flooding-based and hierarchical protocols (standardized by IETF MANET) to geographic and self-organizing coordinate-based routing solutions (charted by IETF ROLL). However, we will only discuss the protocols for LLN in this sub-section.

Through the well-known routing protocols proposed for LLNs and related to standard protocols, we could realize the features of these routing protocols for comparative purpose with RPL. In fact, RPL can be seen as a conjoint outcome of preterit routing technologies, thus we summarized the below content to present an overview on how different routing protocols fit under different categories and also compare their characteristics.

Firstly, we cannot neglect ZigBee standard protocol which was always considered as the most prominent technology for LLNs since 2005. Its famous Cluster-Tree protocol uses link-state packets to form a two-tier cluster-based network and this is similar to the RPL as a hierarchical routing protocol. In 2007, with the emergence with ZigBee Pro 2007, which is adopted by so many industrial projects and also current iLive software system [18], the CTP was not supported and the new standard has adopted flat and mesh routing based on AODV [19].

Since the emergence of IPv6 network, its technologies started to be considered for the LLNs, and then the 6LoWPAN presented itself to the world. Hilow is one of the first routing protocols for 6LoWPAN networks. It is a hierarchical and on-demand routing protocol located on the 6LoWPAN adaptation layer, thus it take advantages of the 6LoWPAN capabilities in term of dynamic assignment of 16-bit short addresses. Its mechanisms are like any other tree routing, and the behaviours of any router are all based on the addresses [20]. To generalize, its addressing mechanism is very similar to ZigBee Cluster-Tree addressing scheme.

During the progress of 6LoWPAN technology, there was another flat on-demand routing protocol designed for it and

named LOAD. It is based on AODV, but without utilization of the destination sequence number, and it supports basic mesh network topologies. LOAD uses Link Quality Indicator and the number of hops as routing metrics for route selection. Excepts these, LOAD adopts the link layer acknowledgments instead of Hello messages used by AODV, and its evaluation experiments showed that this modification saves energy while keeping track of route connectivity. DYMO-low was proposed as a flat routing protocol besides LOAD in the same technical trend. This protocol operates on the link layer to create a mesh network topology of 6LoWPAN devices and has the basic route discovery and maintenance functions [21].

As one of the dependable 6LoWPAN implementation, Berkeley proposed the Hydro as its default routing protocol in BLIP protocol stack. A hybrid mechanism was represented in this protocol which contains both centralized control and local agility. It also uses DAG for routing data from router nodes to the border router that also means all the nodes forward packets to the border router which could forward them to the appropriate destinations. It was like a rudiment of RPL routing protocol but ultimately, it did not arrive the expected comprehensive functions.

After list and present these RPL's competitors, we could realize that RPL has become the exclusive routing protocol for IPv6-based LLNs. RPL provides more comprehensive features including the supports of various kinds of traffic (such as MP2P, P2MP and P2P) and its extraordinary ability to connect to any Internet nodes directly with global IPv6 addresses. Moreover, RPL's topology is built proactively, which is not like AODV-based protocols, RPL discards the energy consuming broadcast RREQ messages. The supporting of both hierarchical and flat topologies that means it provides the benefits of both mesh and tree routing protocols. In addition, RPL combines the paradigms of the distance-vector and the source routing, and supports local and global repair which are extremely suitable for fault-tolerant applications (for instance PA application). Furthermore, the objection function of RPL can contain various metrics which offers great flexibility for different application requirements and enables QoS-award routing. To sum up, RPL is very flexible and can be easily tuned for different domains' applications. However, its drawback of complexity is also caused by its flexibility.

IV. Simulation Module Design and Implementation for NS-3

This section will present our main work of design and implementation of the simulation module for NS-3. We will describe the routing control messages of RPL and approach which are adopted for the efficient data collection in LLNs. Furthermore, some choices we have made during the design and implementation will be also concluded in this section.

Figure 1 shows the module design in a simplified class diagram. The structure of IPv6 ns-3 stack is adopted and complied with, especially for its static routing and ICMPv6 parts. In our implementation, this simulation module in a specific namespace ns3::rpl4dc::RoutingProtocol in ns-3 by extending from the abstract base class ns3::Ipv6RoutingProtocol. The NetworkRouting list of this protocol will still follow the original one of IPv6 static routing, and the NeighborList is extended from IPv6 neighbour cache class which are used by ICMPv6 protocol implementation. The control messages are derived from ns3::Icmpv6Header and Icmpv6OptionHeader because all the RPL messages are just new types of ICMPv6 messages according its specification. The main class glues all these together and represents the all the routing logic.

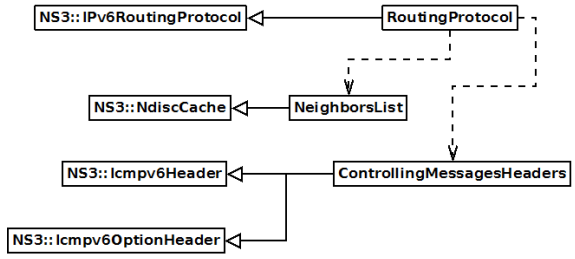


Fig. 1. Simplified class diagram of module

A. Routing Controlling Messages

In the previous section, we have introduced the DODAG building strategy and the IPv6 control messages (DIO, DIS, DAO) used by RPL. As the most important messages of RPL, DIO plays the main role to discover a RPL instance, learn its configuration parameters, and maintain the parent list as well as the DODAG through its carried information. In our simulation module, DIO also takes the cornerstone position since it carries all the indispensable elements related to instance and DODAG, and the sub-optional information, such as DODAG metric container, destination prefix and configuration sub-option used to advertise the parameter such as trickle timers. DIS message has a simple form without any additional messages body and is used to discover DODAGs in the neighbourhood and solicit DIOs from the neighbour RPL nodes. The third RPL controlling message is DAO which is designed for supporting P2MP and P2P traffic preferentially. Based on the reason that has been discussed in the previous section, this module doesn't include the mechanisms of DAO. Thus, we set the DODAG MOP is 0 since all the roots in the RPL instance of this simulation module will not try to find the downward routes using DAO messages.

The metric container option in the routing controlling messages will hold the information to do the effective calculation for the rank of a node compared to its neighbors, which means the selecting of best parent and optimize the DODAG eventually.

B. DODAG Building Process for an Efficient and adaptive Data Collection

In this module, DODAG is operated as a data collection topology and under a set of adaptive trickle timers' control. The DODAG building process is similar to description of RPL standard in the previous sections. The DIS and DIO messages

will be used when a node wants to join or disjoin a DODAG. In the most of cases, the DIO sending is under the control of an adaptive trickle timer [22], and it also can be summoned initiatively by DIS messages. The significant process of DODAG building is the DIO receiving. When this action happens, the node may have two states:

1) Not joined to any DODAG yet: The node will try to join this DODAG. Through processing the copied values hold by the received DIO, the node needs to identify the attributes of this DODAG and RPL Instance. Its rank needs to be calculated to determine a relative position in this DODAG for loop avoidance and launch its trickle timer. The IPv6 address of the sender will be stored in the neighbour list as a preferred parent selection and the Network Routing list will be updated.

2) Already joined one DODAG: If this DIO includes the poisoning information, the node will delete the sender node from its neighbour list and Network Routing list, and forge a new preferred parent from these two lists, or disjoin from the DODAG completely. If this is a periodic update message, the node will check the RPL Instance ID and DODAG version. If any of them has been changed, all the information lists and trickle timer will be reset. If there is no change, the node will carry out the loop avoidance mechanism like RPL specifications except without DTSN (destination advertise-ment trigger sequence number) checking. The neighbour and network routing lists will be updated no matter this node change the best parent selection or not.

To clarify the special features of this module and our efforts to adapt RPL protocol for the efficient simplified data collection application, several left issues will be explained and appended in the next sub-section.

C. Design and Implementation Choices

We made some choices to simplify this module and to adapt the requirement of data collection/gathering application. Except the MOP is set to 0, the important choices are related to the OF and metrics, and messages options.

The temporary removing the downward routing logic can reduce the complexity of this module, and shorten the debugging time. Moreover, the core part of RPL protocol still can be tested in our simulation. For the aspects of routing metrics, the RSSI, ETX, node energy and hop count are implemented, and OF0 (Objective Function 0) [9] and MRHOF [23] will be used to hold these metrics or constraints to organize the policies for the simulated DODAG. As the most important components of a data collection system, the root devices (or edge router) are usually well-preserved, active all the time, and have more resources to maintain a guaranteed connectivity to a specific set of nodes or to a larger routing infrastructure. Thus in this simulation module, all the root nodes are defined to be grounded and presented, which is a trade-off we made during the implementation. It will lose the flexibility of original RPL routing, but it brings the advantages below:

- Inner connectivity within the LLN is undesirable because the measurement data flow cannot be relayed or cached, which means these floating DAGs are redundant for this module.
- The rank computations for both of our OF implementations is simplified.
- The operations of data collection are less complicated since the upward traffic is guaranteed by the edge router as data sink, and also a backup feasible successor for all the nodes in the transmission range.
- This adjustment can be sure that the multiple interfaces policy of edge router is available.

Considering the messages options, the relevant DODAG metric container is designed for this module. Furthermore, the route information option is implemented to advertise the external networks reach-ability. This optional message will be used in large simulation scenarios with multiple roots as data sinks in the RPL instance. This optional message could extend the availability of our module. Especially when the network is over a hybrid link layer, this feature is adopted for the communication between the different DODAGs in the RPL instance.

V. Module Evaluation

This module has been evaluated by a set of simple network cases. These evaluations are performed under different links, such as the new WPAN module [7] and the existing ns-3 Wi-Fi and Ethernet modules. The demonstrations can present the routing logic and the data flow of simulation using a modified NetAnim module. During the evaluation work, the NetAnim GUI is adapted for a better show of network structure, such as the extension of IPv6 and RPL routing supporting.

Figure 2 depicts a sample hybrid network over three link layers. As the edge router can link these WPAN nodes to the Wi-Fi network (In Figure 2 showed in red color points) and transmit UDP data to the external Ethernet network. This network scenario is to simulate a common data collection application that is similar to most of current test-bed system.

Figure 3 shows a DODAG building sample in a Wi-Fi plus Ethernet multi-hop network topology. Note that, there is not a module called Ethernet in the source code repository of NS-3, but normally the CSMA module is adopted by the developers when the wired network simulation is a component of network and the packet is needed to be based on IEEE 802.3.

To evaluate the performance of our data collection protocol module implementation, we performed simulations using the latest NS-3-dev version. The main logic of routing referenced the open source ContikiRPL implementation. Note that ContikiRPL including its uIPv6 stack [24] is specially designed for low-power and memory-constrained devices. We made this design decision for this simulation module will increase the success rate of our future test-bed iLive-CLAS 'Close-loop Application-Simulation' because the similarity between the simulation and hardware program will be an important factor if we need to link the reality to NS-3.

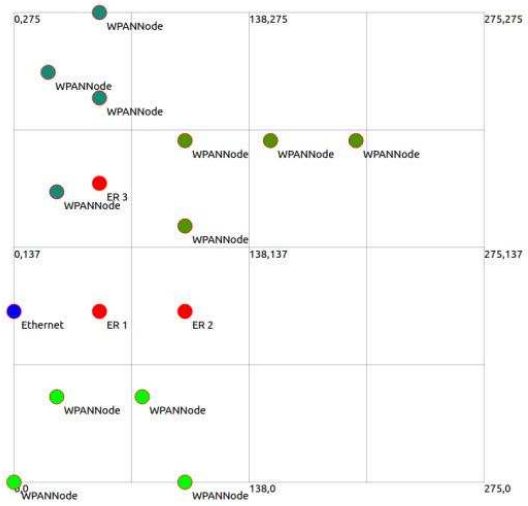


Fig. 2. Sample hybrid network with WPAN, Wi-Fi and CSMA modules

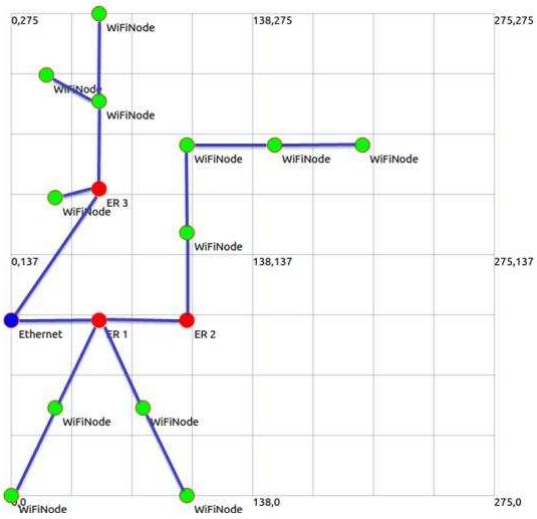


Fig. 3. Sample hybrid network and DODAG building

A. Performance Metrics

Due to the half-baked merging implementation for this module and WPAN module, the current performance evaluation is working under the sample topology depicted in Figure 3. The comparison to the other MANET routing protocol modules of NS-3 is in progress.

As the Figure 3 shows, the Wi-Fi network exports three DODAGs with OF0 and MRHOF, the latter hold a RSSI metric. In this topology, there are three edge routers (Red points), so the multiple roots policy of RPL instance organizes these three devices to relay the collected data to an end device (Blue point) through Ethernet network.

We tried to vary the transmission power level using the configuration set for YansWifiPhyHelper of ns-3 Wi-Fi module, and also the node density to evaluate the RPL behavior in terms of packet delivery ratio, overhead of control messages, path length and delay.

Table I sums up the essential experiment parameters used by this simulation module.

TABLE I
SUMMARY OF THE EXPERIMENT PARAMETERS

	NS-3 simulation platform
Sensor node position (3D)	Static or random in a defined limit area
Number of sensor nodes	15 to 50 nodes
Transmission power	1dBm to 16dBm
Energy detection threshold	-96dBm to -80dBm
Data packet size	64 bytes
Data packet period	15s
Experiment duration	300s
Wireless MAC Protocol	802.11b NS-3 Wi-Fi
Loss model	Friis propagation loss model
Delay model	Constant Speed Propagation Delay Model
Wired Network module	802.3/CSMA

Note that the transmission power and energy detection threshold have been set to achieve a controllable transmission range.

B. Simulation analysis

1) Data Packet Delivery Ratio: Poor packet delivery rate is an ingrained problem. It may be caused by many reasons, such as interferences, collisions, signal attenuation etc. The figure 4 represents the delivery ratio according to the distance with the default Wi-Fi module configuration. The 50 nodes are classified to their Euclidian distance to their sinks and simulated to set in a random way but in a defined limit area. The routs are based on the OF with RSSI metric. However, the result is not satisfying.

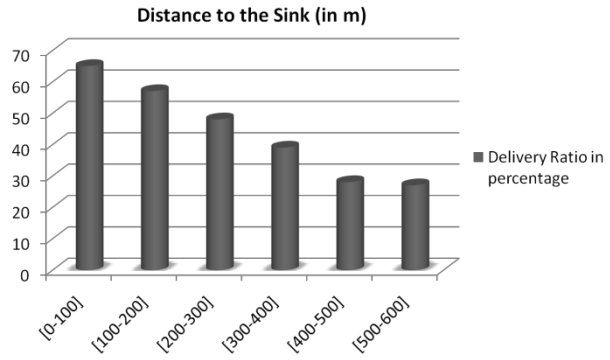


Fig. 4. Delivery ratio versus distance to the sink (or edge router)

In the article [25], the authors have explained the advantages to choose the Euclidian distance instead of hop counts. Due to the configuration of this module, we limit the packet queue like the ContikiRPL. Thus during the data forwarding process, it may meet the packet queue overflow or collisions, which can cause the data packet lost. On the other hand, if the distance between a node and its sink node, the multi-hop routing and the distrustful OF design could be reason to cause such results.

2) Control Packet Overhead: As we have discussed in the previous sections, RPL is a pro-active protocol, and its control

messages are adopted to build and maintain all the elements for the DODAGs. In our module, we have set the DAO messages sending to be suppressive (if not, it will cause more overhead [16, 25]), and in most of cases, the DIO are only control message. DIS messages are used only when a node waits a constant time period without any received DIO. However, as shown in Figure 5, we notice the percentage of data messages is lower than control messages. The big overhead is possibly caused by the ns-3 Wi-Fi module. During the Wi-Fi nodes in the ad-hoc mode, it is difficult to avoid generating large number of control messages from its MAC layer. However, based on the analysis of our module, the DIO and DIS of routing only represents around 17% of total amount of network control messages. We can estimate this module can work better when it merge to WPAN module.

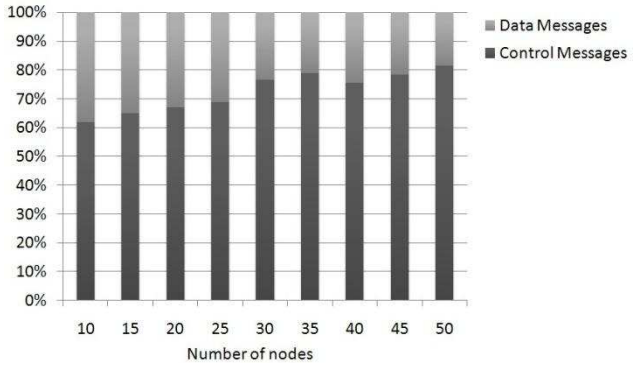


Fig. 5. Overhead ratio according to different number of nodes

3) Path length: In the wireless network, even this is one network without mobility, the topology is still not so stable. In RPL, it adopts rank to manage the dynamicity issue. In Figure 6, it shows the node distribution according to the rank when the 50 nodes-size network becomes stable. The results are based on the average values after 5 time simulations. It is clear to notice that the OF0 always contends for minimum hot count because of its rank calculation mechanism. But for MRHOF with RSSI metric, the nodes will choose the nearest neighbours since the signal attenuation is smaller.

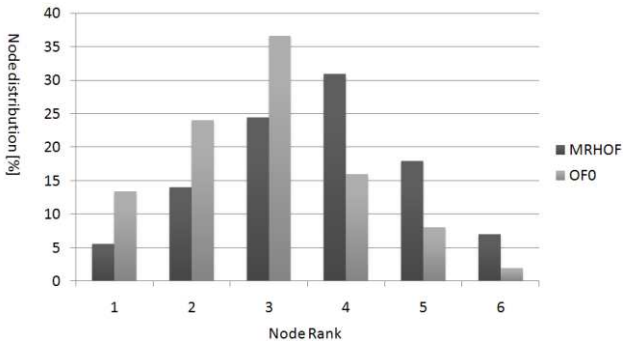


Fig. 6. Node ranks and its distribution

4) Delay: The figure 7 depicts a packet delay curve. In this case, when the number of nodes is increasing, the packet will be relayed by routers to arrive its edge router. This simulation result is obtained from the nodes installed MRHOF, the same reason lead to the poor packet delivery rate may also cause this kind of delay because the data packet need the retransmission.

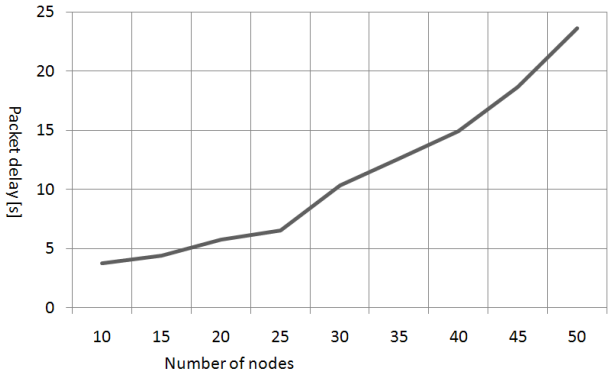


Fig. 7. Packet delay with varying node density

VI. Conclusion and Prospects

This module design and implementation can be used for the validation of upward traffic operation of RPL protocol. However, its real meaning is to offer a simplified routing protocol dedicated for future I/WoT data collection application. From the discussion of previous sections, we have found if it can achieve all the current requirements from this tailored module. Furthermore, the compliance with ContikiRPL's logic and features ensure its conformance of real implementation. On the other hand, this also provided us more understanding and idea about the differences between the hardware and simulation program, and more important is about how to link the reality to the simulation.

To sum up, this module is a part of iLive-CLAS system which can use simulation to do the forecast of the network dynamicity and to optimize the real-world deployment through the simulated rehearsal with varying OF and WPAN MAC layer parameters.

The very next steps is to continue to solve the potential bugs in this module, relate it to the WPAN module, and make a real preparation to integrate all the components of iLive-CLAS system.

Acknowledgment

The authors would like to thank the "Conseil Régional d'Auvergne" and the European Regional Development Funds (ERDF) which have financial supports for this research.

References

[1] T. Winter, *et al*., "RPL: IPv6 Routing Protocol for Low Power and Lossy Networks, RFC 6550," *IETF ROLL WG, Tech. Rep*, 2012.

[2] J. Vasseur, *et al*., "RPL: The IP routing protocol designed for low power and lossy networks," *Internet Protocol for Smart Objects (IPSO) Alliance*, 2011.

[3] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, 2008, p. 60.
[4] F. Österlind, *et al*., "Demo abstract: Cross-level simulation in cooja," in *Proceedings of the First IEEE International Workshop on Practical Issues in Building Sensor Network Applications*, 2006.
[5] O. Gaddour and A. Koubâa, "RPL in a nutshell: A survey," *Computer Networks*, 2012.
[6] L. Bartolozzi, *et al*., "ns-3 RPL module: IPv6 routing protocol for low power and lossy networks," in *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques*, 2012, pp. 359-366.
[7] K. El Gholami, *et al*., "FF-MAC: Fast Forward IEEE 802.15. 4 MAC Protocol for Real-Time Data Transmission," *International Journal*, vol. 4, 2013.
[8] Y. Chen, *et al*., "RPL Routing Protocol a case study: Precision agriculture," in *First China-France Workshop on Future Computing Technology (CF-WoFUCT 2012)*, 2012.
[9] P. Thubert, "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)," 2012.
[10] J. Vasseur, *et al*., "Routing metrics used for path calculation in low power and lossy networks," *draft-ietf-roll-routing-metrics-19 (work in progress)*, 2011.
[11] Z. Shelby, *et al*., "Constrained application protocol (coap)," *draft-ietf-corecoap-07*, 2011.
[12] C. Chauvenet, "RPL deployment experience in large scale networks draft-hui-vasseur-roll-rpl-deployment-02," 2012.
[13] J. Guo, *et al*., "Loop-Free Routing in Low-Power and Lossy Networks," in *SENSORCOMM 2012, The Sixth International Conference on Sensor Technologies and Applications*, 2012, pp. 59-66.
[14] S. Dawans, *et al*., "On Link Estimation in Dense RPL Deployments," 2012.
[15] P. O. Kamgueu, *et al*., "Energy-based routing metric for RPL," 2013.
[16] J. Tripathi, *et al*., "Performance evaluation of routing protocol for low power and lossy networks (RPL)," *IETF, Draft, January*, 2011.
[17] T. Watteyne, *et al*., "From MANET to IETF ROLL standardization: A paradigm shift in WSN routing protocols," *Communications Surveys & Tutorials, IEEE*, vol. 13, pp. 688-707, 2011.
[18] H.-L. Shi, *et al*., "Energy Efficient and Fault Tolerant Multicore Wireless Sensor Network: E MWSN," in *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, 2011, pp. 1-4.
[19] R. Fonseca, *et al*., "The collection tree protocol (CTP)," *TinyOS TEP*, vol. 123, p. 2, 2006.
[20] J. Hui, *et al*., "6LoWPAN: Incorporating IEEE 802.15. 4 into the IP architecture," *Internet Protocol for Smart Objects Alliance, white paper*, vol. 3, 2009.
[21] V. Kumar and S. Tiwari, "Routing in IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN): A Survey," *Journal of Computer Networks and Communications*, vol. 2012, 2012.
[22] J. Tripathi and J. C. de Oliveira, "On adaptive timers for improved RPL operation in low-power and lossy sensor networks," in *Communication Systems and Networks (COMSNETS), 2013 Fifth International Conference on*, 2013, pp. 1-10.
[23] O. Gnawali, "The Minimum Rank with Hysteresis Objective Function," 2012.
[24] N. Tsiftes, *et al*., "Low-power wireless IPv6 routing with ContikiRPL," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010, pp. 406-407.
[25] K. Heurtefeux and H. Menouar, "Experimental Evaluation of a Routing Protocol for Wireless Sensor Networks: RPL under study."