Middle-East Journal of Scientific Research 23 (8): 1639-1651, 2015
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2015.23.08.22418

New Real Evaluation Study of Rpl Routing Protocol Based on Cortex M3 Nodes of Iot-Lab Test Bed

Shimaa A. Abdel Hakeem, Tamer M. Barakat and Rania A. Abul Seoud

Department of Electronics and Communication Engineering, Fayoum University, Fayoum, Egypt

Abstract: RPL is a routing protocol for low power and lossy networks. It is an extensible proactive IPv6 distance vector protocol, which is a brand new proposed standard for including low power and lossy networks in the global realm of IPv6 networks. RPL has been implemented and tested in wireless sensor networks (WSNs) simulations with up to a thousand nodes. Recently, some of the research process has studied the operation of the protocol experimentally in small networks, which is not enough to illustrate the whole features of RPL. So, it is important to implement real large scenarios. In this paper, a new real evaluation study conducted in a large intensive test bed composed of more than 2728 wireless sensors nodes. Around 150 nodes tested the RPL characteristics and behavior on cortex M3-nodes giving a variety and intensive analysis. The results confirmed the stability of path length in dense networks with important control packet overhead while reduced packet delivery ratio. This work is a part of RPLstandardization process.

Key words: Performance evaluation · Testbeds · Wireless sensors · Routing protocols · IPV6 standard

INTRODUCTION

Wireless Sensor Networks (WSNs) are randomly distributed in any environment and composed of a large number of small sensor nodes with low energy and wirelessly interconnected. With these resources constraints, designing routing protocols are quite challenging. Routing is a crucial factor influencing connectivity and performance of information exchange. The general performance of Low Power and Lossy Network (LLN) is highly dependent on the choice of routing protocol and quality of its implementation. Recently, WSNs have been brought into reality with the effective deployment of sensor nodes. Routing in LLNs should be able to self-manage to a large extent and to be able to heal itself without requiring manual intervention. The Internet Engineering Task Force (IETF) [1] formalizes a new working group in order to standardize IPv6-based routing solution that led to creation of a new working group called ROLL (Routing Over Low power and Lossy) networks in 2008 [2]. The ROLL found a very important analysis of the routing requirements focusing on various applications. The result of this Working was the “Ripple” routing protocol (RPL) specification, along with many supporting specifications on routing objective functions, metrics and security mechanism. After some years, RPL was announced by the IETF in March 2012.

Related Work: Recently, simulations and implementations of RPL have been provided. For this internet draft [3], the evaluation of RPL performance depends on several routing metrics. Using OMNET++/Castalia simulators, which is one of the discrete event simulators [4], RPL protocol was exprinced using real data files and different topologies that were developed to assess the protocol behavior. In addition to some suggested broadcast mechanisms using NS2 simulator, the multipoint-to-point performance of RPL was investigated [5-7]. At Johns Hopkins and Berkeley universities, an implementation of RPL for TinyOS 2.x is under development [8]. The RPL performance was evaluated using simulations based on topology and link quality data extracted from a real network [8,9]. It was observed that the number of control packet was firstly increased and then decreased during the stabilization of the DODAG. RPL was found to allow a fast network

Corresponding Author: Shimaa A. Abdel Hakeem, Department of Electronics and Communication Engineering, Fayoum University, Fayoum, Egypt. E-mail: sa114@fayoum.edu.eg .

set-up and also to be important for reducing control packet overhead [9]. Recently, experimental evaluation for RPL become possible and has already been done with small scenarios moving towards using test beds to find RPL behavior for large scale networks. RPL implementation was ported to two typical hardware platforms, namely the MSB430 Scatter web mote and the WSN430 Sensor Board [10]. The core mechanisms of the protocol were implemented in C on the Contiki OS software platform. RPL performed equally well independently from the different network topologies it was running on. Another evaluation over test bed was done [11] using 100 nodes on Lille SensLAB platform, in which the nodes are randomly deployed in an indoor environment. It was proven that RPL could be efficient to build shortest path even in a very large scale network and is considered as a smart routing protocol. Therefore, the understanding of RPL behavior in realistic scenarios and environment is worth to be investigated.

This paper introduces a new evaluation study of RPL behavior in terms of new configuration parameters with new hardware specification to give a wide range of information concerning its stability over different physical networks. The performance of the network is also studied in terms of these metrics: control traffic overhead, packet loss, routing table size, average hop distance and convergence time .This study is based on IoT-LAB [12] testbed for implementing wireless sensor networks. The implementation was done using around 150 nodes at Inria Lille – Nord Europe area in France with cortex M3 platform running operating system (Contiki). The UIPv6 stack of Contiki was used [13], which provides IPv6 networking and contains RPL routing protocol. The foren6 diagnostic tool was used to analyze the network behavior [14].

RPL Operation and Messages: RPL is a routing protocol responsible for forwarding the packets between nodes and making intelligent routing decisions. RPL finds the best route towards destination, before it is actually needed. It distributes messages across the entire network for sharing the topology related information among all the nodes in the network. RPL uses several factors while computing the best paths. RPL creates a tree-like topology with a root at the top and leaves at the edges using "up" and "down" directions. The topology information is maintained in Destination Oriented Directed Acyclic Graph (DODAG). The DODAG contains the paths from the leaves to the root. The root is also called an LLN (Low Power and Lossy Network border router) (LBR). Each node has a rank, which defines its position relative to other nodes with respect to the DODAG root. The node ranking is strictly increased from the root towards the leaf. The rank is computed depending on the DODAG's Objective Function (OF) [15] that defines the optimization objectives, routing metrics and other related functions. Moreover, the OF determines the DODAG formation and how DODAG parents are selected. The RPL protocol populates DODAG with the parent information. The DODAG uses control packets to build and maintain its logical topology e.g. (route, parents' neighbors and table). RPL uses an IPv6 control message which consists of [16]:

- DODAG information object (DIO): DIO messages are used by RPL to form, maintain and discover the DODAG. When a RPL network starts, the nodes start exchanging the information about the DODAG using DIO messages, which contain information about the DODAG configuration.
- DODAG Information Solicitations (DIS): The DIS is used by any node to explicitly solicit the DIO messages from the neighbor nodes. It is triggered by the node in case when it could not receive a DIO after a predefined time interval.
- DODAG Destination Advertisement Object (DAO): The DAO messages are used by RPL to propagate a node prefix to the predecessor nodes that support the downward route traffic. At first, the (DAG) root initiates emmiting the DIO messages while the DODAG is progressively simply constructed. With respect to the timing process, the trickle algorithm timer is used by the protocol to find the minimum time period, in which the nodes start to resend the DAOs. This interval time is defined by the trickle [17].

Experimental Framework: In this study, IOT_LAB test bed is used to conduct the experiments. A subset of 150 cortex M3 nodes was chosen to adapt the experiment framework. The position of nodes is an effective parameter that affects the convergence time of the experiment, so that nodes deployed carefully and the border router is chosen to be in the middle of all nodes. Also, the Contiki operating system for low power and lossy network devices represent the firmware that nodes need to be flashed up. The binary images of the firmware is compiled to adapt the required one for each node to be updated. The experiment was completely monitored.

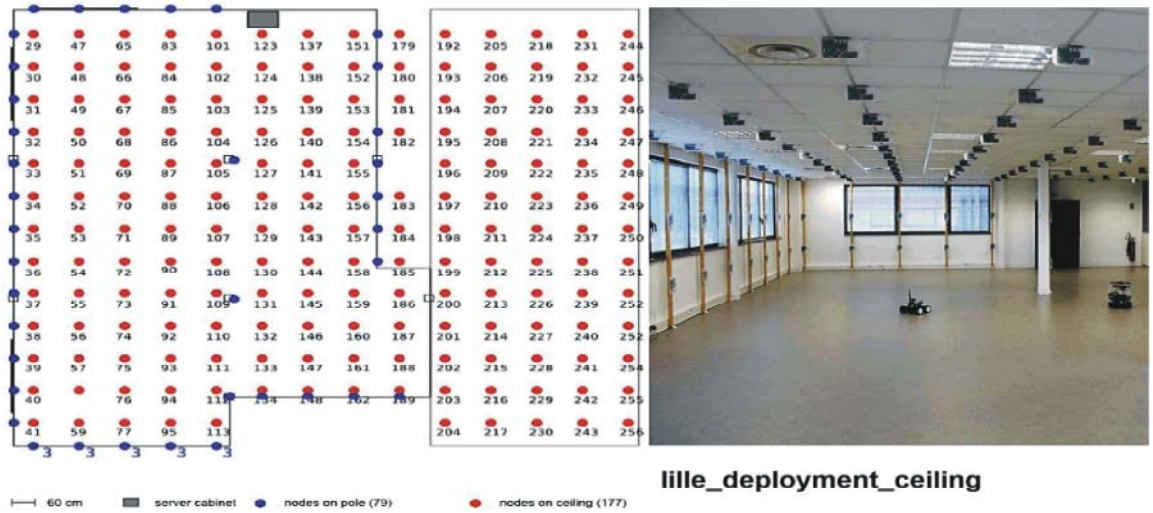

Fig. 1: Lille -deployment -ceiling around 177 nodes over a 1.20 m x 1.20 m grid, at 2.50 m high

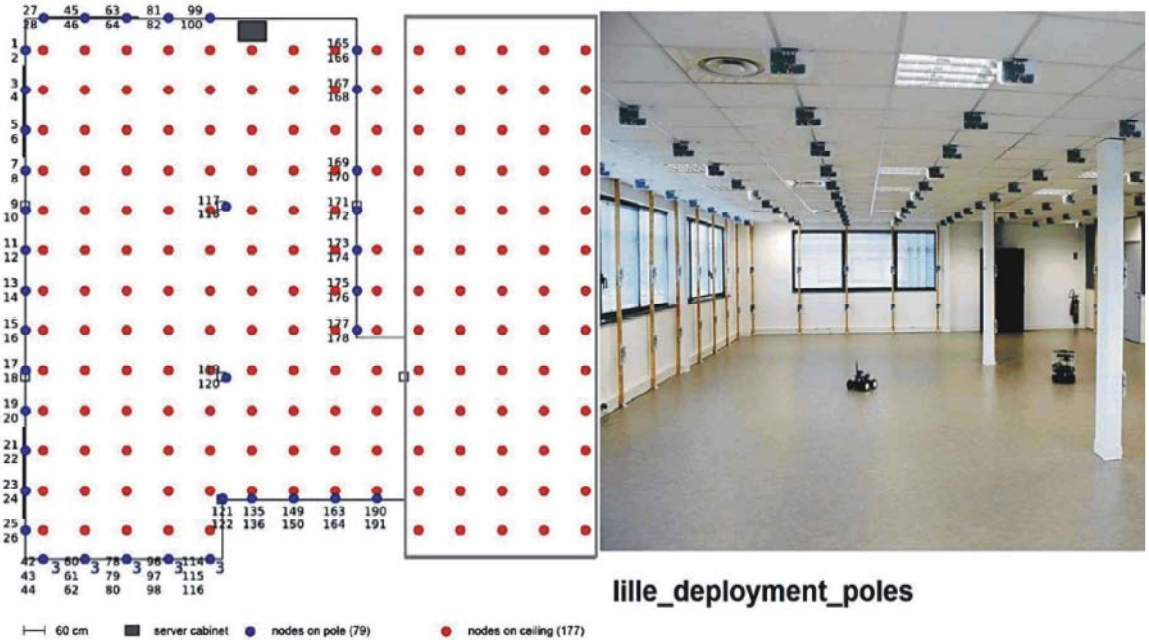

Fig. 2: Lille -deployment –poles around 79 nodes vertically hanged at 2.40 m, 1.50 m and 0.60 m high

Description of Iot-Lab Test Bed: Iot-LAB is a very large scale infrastructure suitable for testing small wireless sensor devices and communicating heterogeneous objects. IoT-LAB test beds are located at six different sites across France, which gives forward access and control of 2728 wireless sensor nodes. This study used a subset of 150 nodes of the Lille _Nord Europe resources [18] that is composed of 320 M3 open nodes. Lille testbed topology was deployed over a 225 m^2 area. Nodes were dispatched over the wooden poles and ceiling, situated at the borders of big rooms. Lille physical deployment is expressed over ceiling as shown in Figure 1, however, Lille physical deployment is expressed over poles as shown in Figure 2. Nodes on the ceiling were deployed over a 1.20 m x 1.20 m grid, at 2.50 m high. Nodes on the poles were vertically hanged at 2.40 m, 1.50 m and 0.60 m high [19].

Any IoT-LAB node consists of the three main components: open node, control node and gateway node. Both of gateway and control nodes offer a connection to control and monitor the open node. The control node was always used to interact actively or passively with the

Table 1: M3 open node structure

M3 Open Node Components	Description
Microcontroller	(ST2M32F103REY): ARM Cortex M3 microcontroller with 32 bits and 72 MHz offering 64 KB RAM.
Radio communication chip	(AT86RF231): A single-chip with 2.4 GHz radio transceiver and Programmable transmitted Power from -17 dBm up to +3 dBm with maximum transmit ion rate of 250 kbps that targeted for IEEE 802.15.4 [19] PHY standard.
Power	(LiPo battery):3,7V, 650 mAh

Table 2: Host node structure

Host Node Components	Description
The main processor board	Avariscite VAR-SOM-AM35 CPU. The main processor controls the "co-processor"
Radio communication chip	(AT86RF231): A single-chip with 2.4 GHz radio transceiver
Power	The board can be powered either over Ethernet (PoE) or by an external power supply (jack connector).
The co-processor	(ST2M32F103REY): ARM Cortex M3 microcontroller with 32 bits and 72 MHz offering 64 KB RAM.
Connectivity	The board offers three USB ports, three Ethernet ports and the dedicated open node connector.

open node, to monitor the consumption and sensor values during experiments and to select the power supply. While an open node wass totally open, the user is granted full access to hardware and memory. Meaning that, any operating system can be loaded, run and debugged. IoT-LABexposes the device serial line and provides start, reboot and stop. Finally, the gateway node is the most important one, as it handles the open node serial link, while the node is set to be a sink node. This study was done over a new generation of M3 open node, which were notably composed of different components. Table 1 summarizes M3 components and for more details consult M3 datasheet [20].

For the new generation of M3 nodes, the gateway and the control node are on a same board called host node. Each node of an experiment is managed and controlled by its host Node, while users have no access to it, for more details consult IoT-LAB host node datasheet [21]. Table 2 summarizes the host node components.

Description of Operating System Contiki: The open source operating system Contiki is the default sensor nodes operating system. It is designed especially for low-power and memory constrained devices. Virtual Linux environment is the basic requirement of Contiki. Contiki provides three network mechanisms: the Rime stack [22], which is a set of lightweight networking protocols review, the uIP TCP/IP stack providing IPv4 networking and the uIPv6 stack providing IPv6 networking. The UIPv6 stack was used, which provided IPv6 networking and contained the RPL routing protocol. The RPL process needs compilation of three types of firm wares, RPL border router firmware, http client firmware and foren6 sniffer firmware. Concerning the used MAC layer, the sensor nodes used a simple MAC layer called Contiki MAC [23] for packaging radio packets into 802.15.4 frames.

MATERIALS AND METHODS

Sensor nodes periodically send data packets to the root node; using the RPL to reach their destination. The network is able to run concurrently multiple instances of RPL. This study is limited to use only fixed nodes to run one instance of RPL with only one root node. The main Iot-Lab test bed experiment steps are summarized as shown in Figure 3. Experiment is initially started up by choosing the nodes in Inria Lille – Nord Europe. The nodes are then compiled and flashed by the required binary images (binary files are the result of compilation there is no need to compile). The Interaction with experiment objects is done through nodes serial lines. The main steps to prepare the experimental frame work are illustrated as follows:

- Programming of nodes: using the Contiki open source operating system which is flexible and lightweight operating system for tiny networked, constrained memory devices. In this study, three types of firm wars have been compiled to flash the nodes up.
- Positioning of Nodes: it means how to choose the required nodes with best positioned. As border router node must be in the middle of the experiment area so M3-140 is the root. Collector nodes also need good positioning to sniff all network traffic and monitor other nodes.
- Accessing of nodes: node setup is carried out using the three provided firmware using command line interface (CLI) tools to create a simple IPv6 network. These nodes are accessed directly using the HTTP browser over IPv6 from the local machine by using a local Contiki Tunslip6. Tunneling process makes accessing process easy to be done as IOT_LAB infrastructure constructed depending on

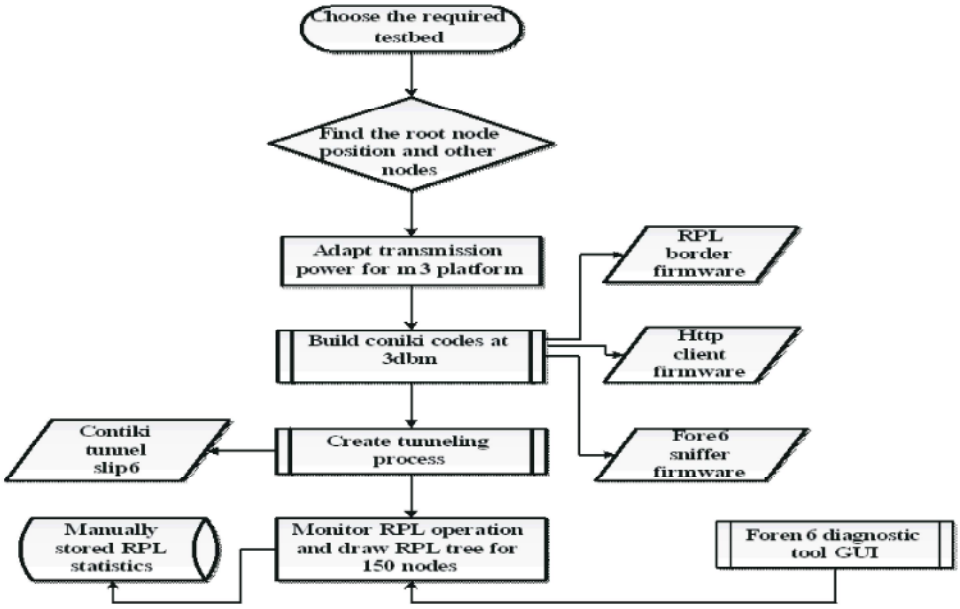

Fig. 3: Iot-Lab test bed experiment steps

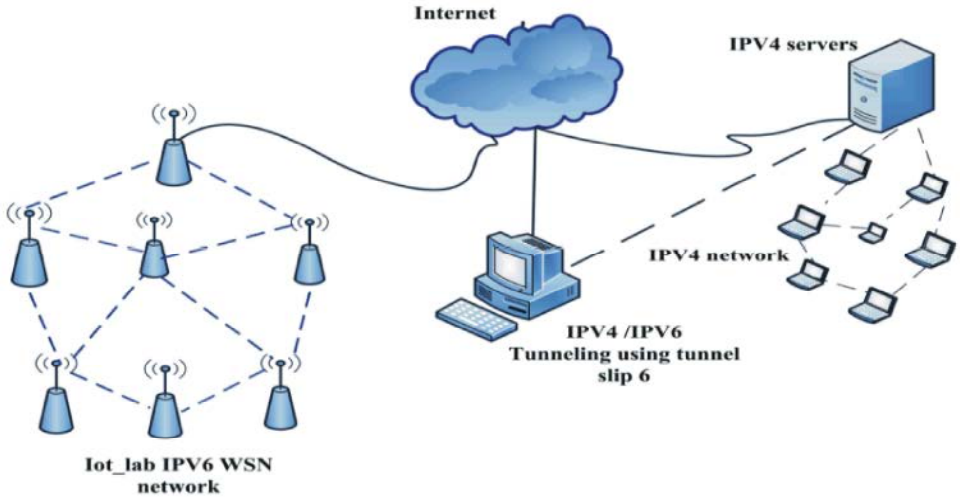

Fig. 4: Contiki Tunslip6 process, a tool used to bridge the wireless IPv6 network traffic onto a PC connected to IPV4 network (ipv4/ipv6 tunneling).

IPV6 framework while internet and other networks build with IPV4 mechanism so accessing and monitoring of the nodes could not achieve without tunneling. The Contiki tunneling process used Contiki Tunslip6 tool to bridge the wireless IPv6 network traffic onto a PC. Tunslip creates a virtual network interface (TUN) on the host side and uses serial line internet protocol (SLIP) to encapsulate and pass the IP traffic to the other side of the serial line as shown in Figure 4.

- Monitoring of nodes: IOT_LAB infrastructure allows users to monitor their running experiment using serial lines of each node with the help of the new diagnostic tool Foren6. Integrate Foren6 with IOT_LAB nodes needs a new compiled firmware to make some nodes sniffers.Foren6 is a non-intrusive 6LowPan network analysis tool provided by the Belgian research center CETIC [24]. IOT_ LAB provides a mote firmware whose purpose is to capture the radio traffic around a node and stream it on its serial line. IOT_LAB uses a sniffer node that is flashed with foren6_sniffer.elf firmware to capture the RPL traffic and render the network state in a graphical user interface.

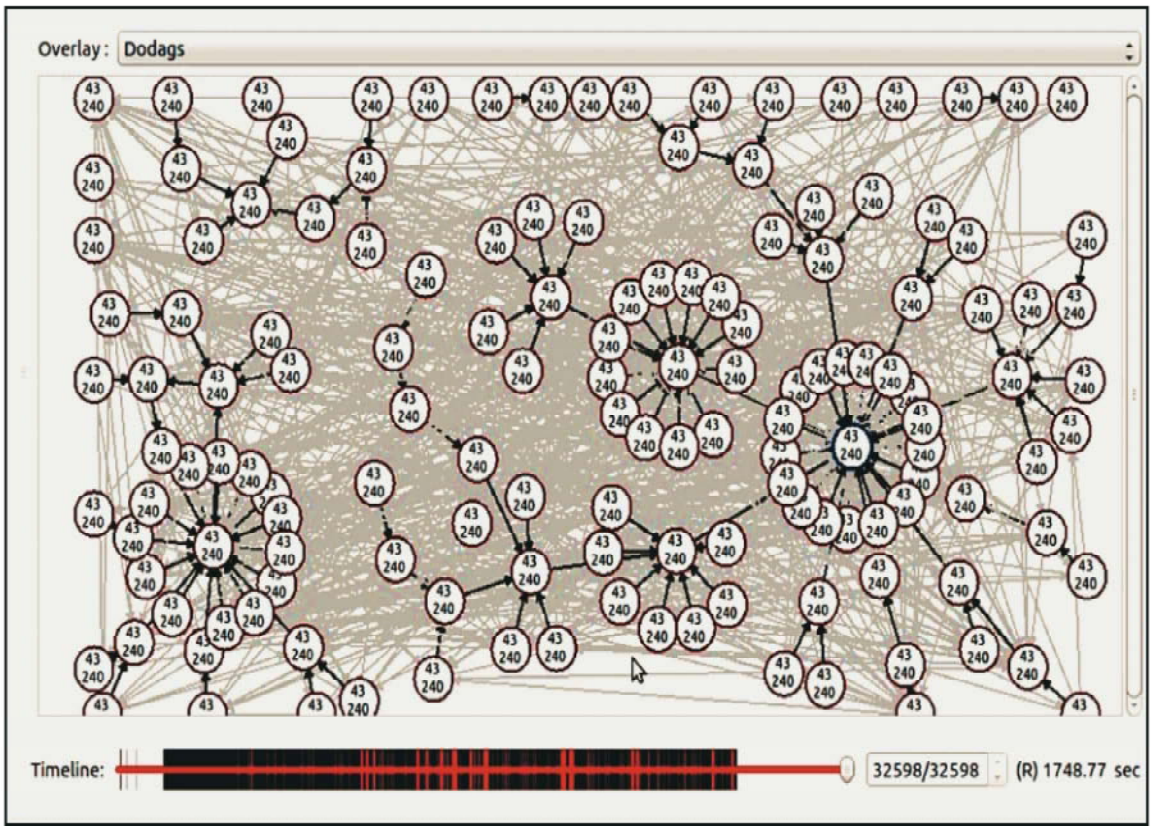

Fig. 5: The experimental multi-hop logical topology. Snapshot of foren6 GUI for the RPL operation during the experiment when all nodes joined the DODAG and form their routing tables

Experiment Parameters and Network Operation: This study, based on an experiment which has been repeated about 70 times to a maximum of 2 hours for each one with more than 53000 packets exchanged between nodes. In the beginning of the RPL border process, all nodes with each provided firmware is firstly updated and then each client node tries to join the DODAG. The DAG root broadcasts RPL DIO messages that used to announce the DODAG ID and the DODAG root rank value. All nodes in the network receive this DIO message and start to calculate their rank value then add the DIO sender as a parent in their parent list. When a node calculates the rank depending on the objective function and finds that it is greater than the DIO sender node, it updates the DIO message with the new rank and forwards it again .RPL tree is completely constructed after experiment status reached stable case and nodes build their routing table entries. Figure 5 shows the total RPL tree while all nodes are joined to the DODAG and construct their routing table's entries. All details about the RPL protocol parameters were presented [16]. Table 3 summarizes all the experiment parameters.

Performance Evaluation Results: According to previous work; RPL was evaluated against different parameters, the author experimented RPL in Contiki [25]. It is a Cooja Simulation based, that studies the impact of different RPL parameters on its performance. In this work, RPL configuration parameters that lead to efficient evaluation of RPL are fixed, we do not need to study the impact of their variation [26]. Table 4 summarizes all parameters.

The performance metrics that are studied here; Control traffic overhead, packet loss, routing table size, average hop distance and convergence time [3]. In this paper, all of these metrics are investigated while the adapted transmitted power for all nodes is 3 dBm. RPL different metrics are investigated in the following section.

Routing Table Size: The routing table size represents the number of routing entries; that stored during the network formation process and measured in terms of the number of entries for each evaluated node. The routing table memory requirement changes in accordance to the position of each node in the DODAG . The closer to the DAG root, the more entries needed to be stored. Keeping in mind

Table 3: The experiment parameters

Parameters	Value
Total Number of sensor nodes	150 nodes
Sensor nodes position (3D)	Uniform distribution of ceiling nodes and poles nodes (grid topology over 225 m2)
Test bed site	Inria Lille_Nord Europe site with cortex m3 platform
Experiment duration	120 minutes
Operating system	Contiki OS version 2.7
Frequency	2.4 GHZ divided into channels starting from 11 to 26, here the default contiki channel is (channel 11) while through experiment (channel 22) is used
Mac protocol	Contiki MAC protocol(it is very simple asynchronous mechanism with no signaling messages)
Transmit ion power	-17 dBm to +3 dBm while all experiments conducted at 3dBm transmitted power.
Startup delay	Around 65 s
Data rate transfer	250 kbps
The number of collectors	7 flashed up with foren6 sniffer firmware
The number of border router	Only one node flashed up with border router firmware

Table 4: RPL configuration parameters

RPL configuration parameters	Values through experiments
DIO Interval Minimum(ms)	12
DIO Interval Doublings(ms)	8
RPL Mode Of Operation	2 (storing mode with no multicast support)
Use Authentication	No
Path Control Size	0
DIO Redundancy Constant	10
Max Rank Increase	1792
Minimum Hop Rank Increase	256 (the root rank is 256)

that the worst assumption is made so that all nodes have equal memory capacity for storing the routing entries and there is no route aggregation in the network. The objective of this metric is to observe the distribution of the number of routing table entries per node. Figure 6 shows the routing entries for 146 nodes join the DODAG. More than 90 % of nodes have less than 20 entries; this contrast between results was obtained [3]. The authors of [3] proved that 90 % of nodes have less than 10 entries with case study of 45 nodes outdoor topology. Also, the root node does not have the same memory or power constraints like the other nodes do [3]. Nevertheless, in this study we have a large network deployment consists of 151 nodes; all nodes have equal memory and power. According to these two different results we can prove that the routing table size depends on the size and density of the network; the higher the number of neighbors, the more entries the routing table contains. Figure 7 shows the routing table entries for only one node when access it's IPV6 address using HTTP browser.

Control Traffic over Head: In this section, the ratio between data packets and control Packets is studied. For any routing protocol it is important to have negligible control traffic. The control traffic is used to update and maintain routing table, build the DODAG and share the protocol configuration between nodes. Figure 8 shows the distribution of different RPL control messages (DIO, DIS and DAO). The large part of control messages is composed of DIO messages that represent around 90 %, while DAO and DIS Messages represent only 10% and less than 1%, respectively [27].

In contrast to the related work results were obtained [10, 3].Author, *et al*., [3],observed that for both the small 45 node network and for the large scale scenario deployments, the amount of the control traffic will be negligible compared to the amount of data packets. The results showed that RPL performance was independent of the network's size and work equally well for all various deployments [10], however, the expectation of a uniform control traffic distribution was not proved. In this work, control traffic cannot be negligible; it is a bit less than half of the total node traffic reported [11]. This means that control traffic represents an overhead for large network scenarios but still has a steady state relation with data traffic that they seems to be equal. Figure 9 shows the relation between control traffic and total node traffic which illustrate high control traffic.

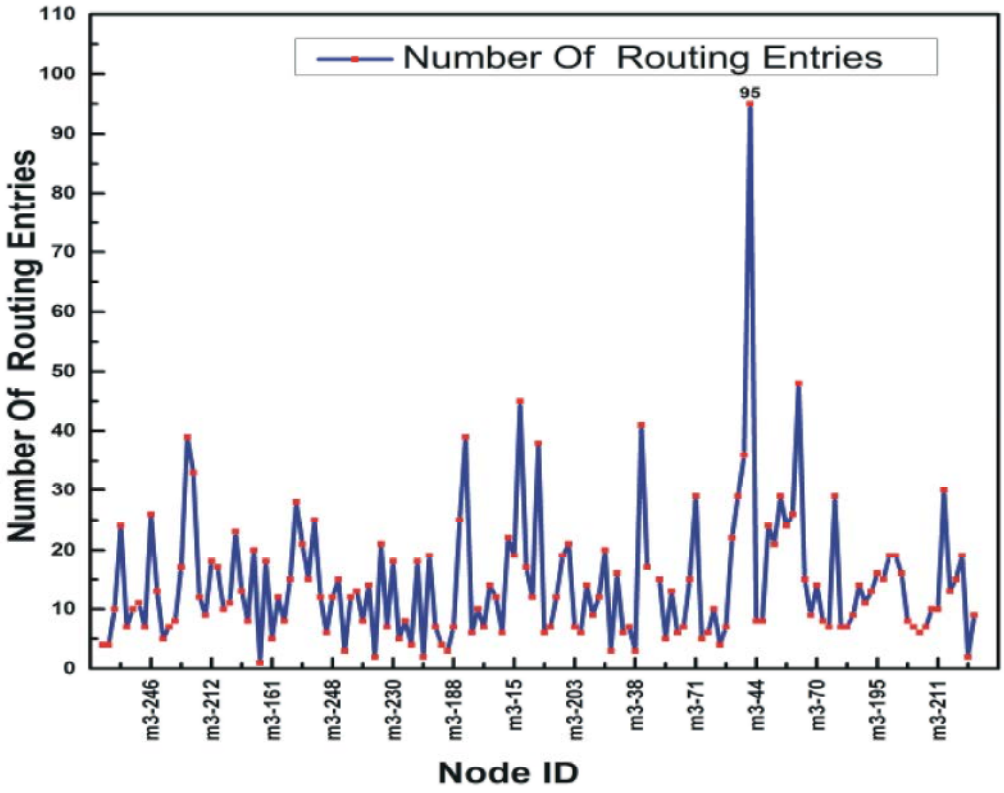

Fig. 6: Routing table entries per node extracted manually from each node through experiment

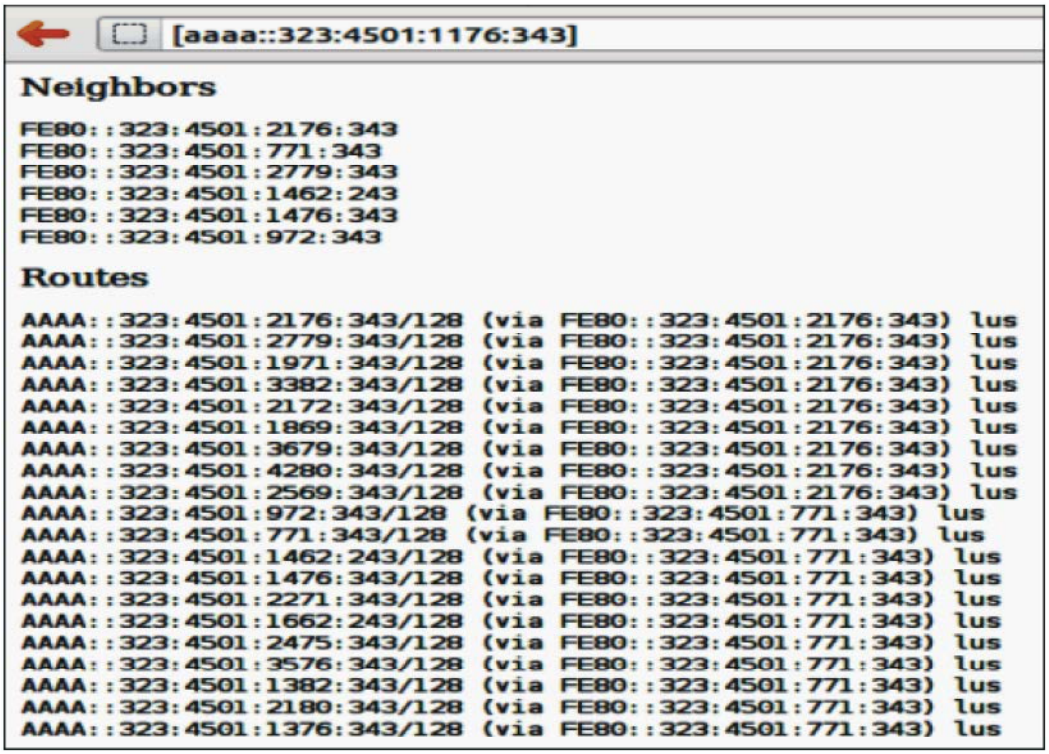

Fig. 7: The routing table entries and neighbors list of only one node when accessing its IPV6 address via http browser.

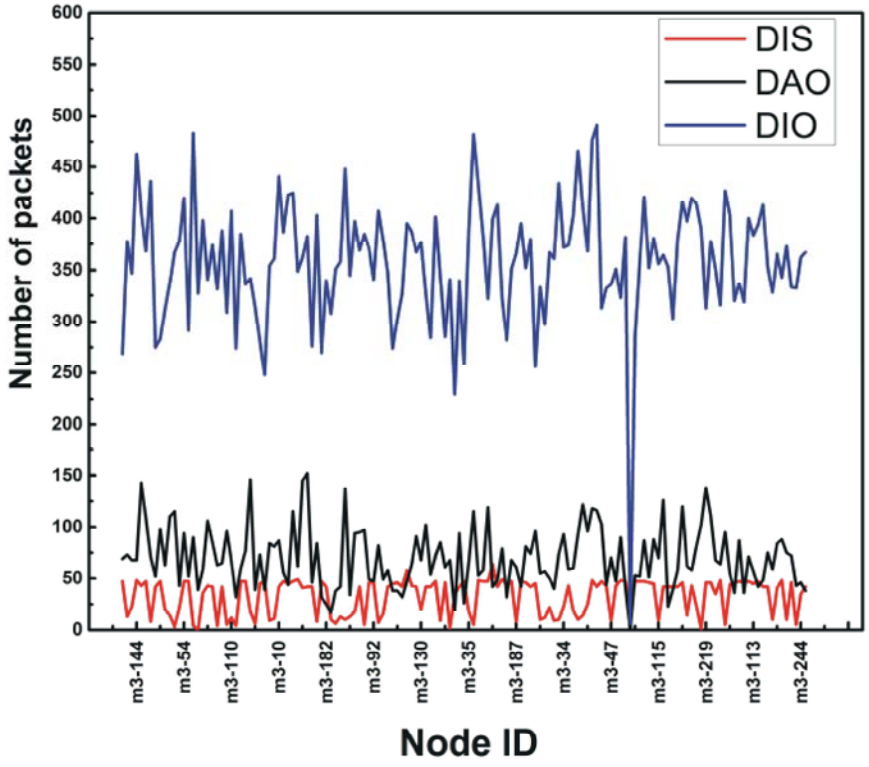

Fig. 8: Control messages of RPL per each node (DIO, DIS and DAO)

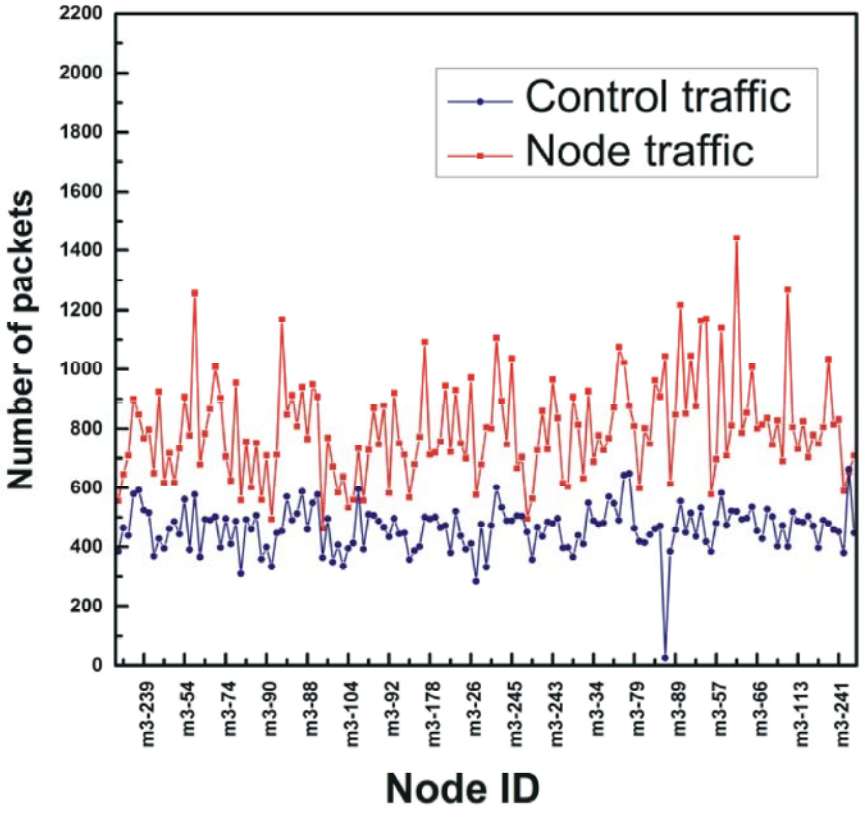

Fig. 9: Comparison between total node traffic and control traffic per each node

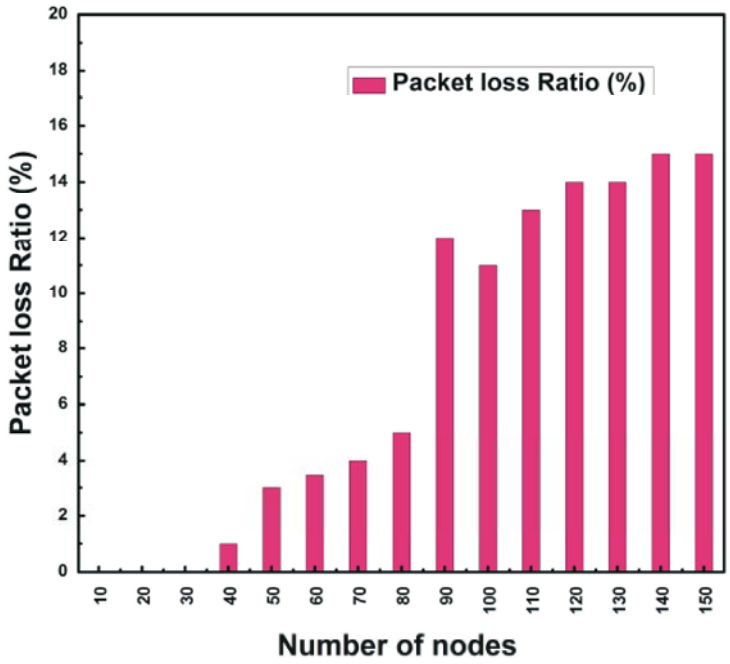

Fig. 10: Packet loss ratio

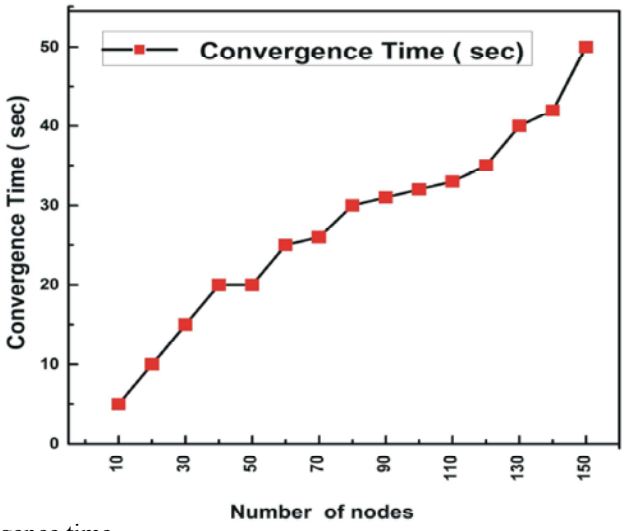

Fig. 11: DODAG convergence time

Acket Loss: Packet loss occurs when packets traveling across the DODAG fail to reach their destination. The packet loss always occurs for many reasons described [28]. In this study, our topology is deployed over a 225 m^2 area distributed uniformly at a distance of 1.2 m so all nearby nodes have no link attenuation or fading, therefore, packet loss is negligible. Although the RPL has been implemented especially to adapt lossy networks constraints, the results obtained from [10] and [11] show an important packet loss of RPL for various large scenarios deployments. Figure 10 shows the calculated packet loss percentage for different number of nodes which is more than 10 % for the total of 150 nodes deployment during two hours of experiment.

DODAG Convergence Time: DODAG convergence time is the setup time of network determined by the time needed by the last node in the network to join a DODAG and get its IP address. The convergence time noticed to be a short time that needed by the DODAG to be completely constructed with all nodes joined and have the same DODAG version number. It is clear that the

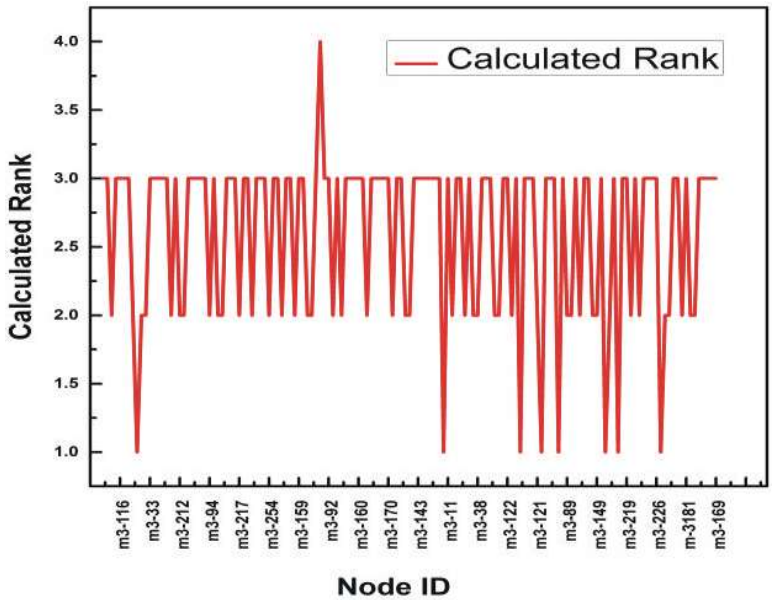

Fig. 12: Calculated rank values per each node depending on ETX metric and hop count

convergence time increases proportionally with the network size, it decrease when the root node position is chosen carefully to be near from all nodes (at the middle of network such as our case) to forward the DIOs messages rapidly. As soon as the root node is updated by the border router firmware, all client nodes start joining the DODAG and start calculating it's rank with respect to the root node, this process take a few seconds to take place. Figure 11 shows the convergence time while increasing network size up to reach 150 nodes [28].

Average Hop Distance: Average hop distance is defined as the number of hops between each node and the DAG root. The number of hops primarily depends on the metric that is specified by the objective function. As mentioned before, our objective function value through experiment equals 1, thus according to Contiki files, ETX is the default objective function. While calculating the rank, 256 are used as MinHopRankIncrease which is the minimum increase in the Rank between a node and any of its DODAG parents. When an Objective Function computes the Rank, it uses this relation: DAGRank (rank) = floor (rank/MinHopRankIncrease) where floor(x) is the function that evaluates to the greatest integer less than or equal to x. Actual rank depth values are extracted from foren6 files and the rank value is calculated with respect to the root rank (256) using rank formula. The average rank value is a constant which means that the shortest path using the RPL protocol is preserved. Figure 12 shows the calculated rank depending on (ETX and hope count).

CONCLUSION

In this paper, we investigated the RPL behavior in a large real network by performing experiments and tests on IOT_LAB test bed to explore RPL behavior on cortex M3 platform. The main contribution of this paper in RPL studies depending on large and dense real networks will be underlined as follows:

- Stability by finding the shortest path to destination with quiet packet loss and low packet delivery ratio.
- RPL protocol works well in dense networks and minimizes the required memory capacity to store routing entries.
- Around 50% of node total traffic represents control traffic while the main purpose of any routing protocol is to minimize its control messages.
- RPL convergence time depends mainly on network size and position of the border router (DODAG Root).
- Finally RPL protocol works well in large and dense networks and can be adopted to be the first wireless sensor routing protocol. In future works, RPL behavior will be investigated at various transmission power levels to find the best level of transmission to make it as standard for M3 platform.

REFERENCES

1. 'Internet Engineering Task Force (IETF)'. http://www.ietf.org/,accessed January 2015
2. 'Routing Over Low power and Lossy networks (roll) – Documents' ttp://datatracker.ietf.org/ wg/roll/documents/',accessed January 2015.
3. Tripathi, J., J. Fe Oliveira and J. Vasseur, 2012. Performance Evaluation of Routing Protocol for Low Power and Lossy Networks (RPL). IETF RFC 6687.
4. Varga, A. and R. Hornig, 2008. An overview of the OMNeT++ simulation environment. In Simutools '08 Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp: 1-10.
5. Clausen, T. and U. Herberg, 2010. Comparative study of RPL-enabled optimized broadcast in Wireless Sensor Networks. In Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2010 Sixth International Conference on, IEEE, pp: 7-12.
6. Afanasyev, M., D. O'Rourke and B. Kusy, 2010. Heterogeneous traffic performance comparison for 6LoWPAN enabled low-power transceivers. In Workshop on Hot Topics in Embedded Networked Sensors (HotEMNETS' 10), pp: 1-5.
7. 'The Network Simulator - ns-2'. http://www.isi.edu/nsnam/ns/. accessed January 2015
8. Tripathi, J., J. De Oliveira and J. Vasseur, 2010. A performance evaluation study of RPL: Routing Protocol for Low power and Lossy Networks. In 2010 44th Annual Conference on, IEEE: Information Sciences and Systems (CISS), pp: 1-6.
9. Accettura, N., L. Grieco and G. Boggia, 2011. Performance analysis of the RPL Routing Protocol. In Mechatronics (ICM), 2011 IEEE International Conference on, IEEE, pp: 767-772.
10. Maria, B.V., 2010. Routing in Wireless Sensor Networks: An Experimental Evaluation of RPL. M.S. thesis, Universität Augsburg.
11. Heurtefeux, K. and H. Menouar, 2013. Experimental evaluation of a routing protocol for wireless sensor networks: RPL under study. In Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP, IEEE, pp: 1-4.
12. 'FIT/IoT-LAB, Very large scale open wireless sensor network testbed'. https://www.iot-lab.info,accessed January 2015.
13. Durvy, M., J. Abeille and P. Wetterwald, 2008. Making sensor networks ipv6 ready'. In SenSys '08 Proceedings of the 6th ACM conference on Embedded Network Sensor Systems, pp: 421-422.
14. 'Foren6: A 6LoWPAN Diagnosis Tool'. http://cetic.github.io/foren6/index,accessed January 2015.
15. Brachman, A., 2013. RPL Objective Function Impact on LLNs Topology and Performance'. In Internet of Things, Smart Spaces and Next Generation Networking, Balandin S andreev S, Koucheryavy Y (ed), 1st ed. Springer Berlin Heidelberg, pp: 340-351.
16. Winter, T., P. Thubert and A. Brandt, 2011. RPL IPv6 Routing Protocol for Low power and Lossy Networks, IETF draft-dt-roll-rpl.
17. Kermajani, H. and C. Gomez, 2014. On the Network Convergence Process in RPL over IEEE 802.15.4 Multihop Networks: Improvement and Trade-Offs'. Sensors2014; 14(7): 11993-12022.
18. 'Lille • FIT/IoT-LAB'. https://www.iot-lab.info/deployment/lille/#lille, accessed January 2015.
19. 'Draft Amendment to IEEE Standard for Information technology-Telecommunications and information exchange between systems-PART 15.4:Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs): Amendment to add alternate PHY (Amendment of IEEE Std 802.15.4)'. IEEE2006.
20. 'M3 open node • FIT/IoT-LAB'. https://www.iot-lab.info/hardware/m3/,accessed January 2015.
21. 'iot-lab/iot-lab'. https://github.com/iot-lab/iot-lab/wiki/Hardware_Iotlab-gateway, accessed January 2015
22. Dunkels, A., 2007. Rime - a lightweight layered communication stack for sensor networks' in the In the Proceedings of Short paper 4th European Conference on Wireless Sensor Networks (EWSN).
23. Despaux, F., Y. Song and A. Lahmadi, 2014. Modelling and Performance Analysis of Wireless Sensor Networks Using Process Mining Techniques: ContikiMAC Use Case'. In DCOSS '14 Proceedings of the 2014 IEEE International Conference on Distributed Computing in Sensor Systems, IEEE Computer Society Washington, Marina Del Rey, pp: 225-232.

24. 'IPv6 over Low power WPAN (6lowpan) – Charter'. http://datatracker.ietf.org/wg/6lowpan/charter/., accessed January 2015.
25. Ali, H., 2012. A Performance Evaluation of RPL in Contiki: A Cooja Simulation based study.M.S. thesis, Blekinge Institute of Technology.
26. Gnawali, O. and P. Levis, 2011. Recommendations for Efficient Implementation of RPL. IEFT draft-gnawali-roll-rpl-recommendations-05.
27. 'AT86RF231'. http://www.atmel.com/devices/at86rf231.aspx?tab=documents., accessed January 2015.
28. Gollakota, S., 2014. Embracing Interference in Wireless Systems'. Morgan & Claypool Publishers: San Rafael, pp: 17.