Research Article

QoS Routing RPL for Low Power and Lossy Networks

Belghachi Mohamed¹ and Feham Mohamed²

¹Faculty of Science and Technology, University of Bechar, 08000 Bechar, Algeria
²STIC Laboratory, Faculty of Technology, University of Tlemcen, 13000 Tlemcen, Algeria

Correspondence should be addressed to Belghachi Mohamed; m.belghachi@yahoo.fr

Received 4 May 2015; Revised 8 October 2015; Accepted 12 October 2015

Academic Editor: Anfeng Liu

Energy conservation, while ensuring an adequate level of service, is a major concern in Low power and Lossy Networks (LLNs), because the nodes are typically deployed and are not replaced in case of failure. Several efforts have recently led to the standardization of a routing protocol for LLNs. The standard provides several criteria that can be used as a routing metric. The working group RoLL of the IETF developed a routing protocol for 6LoWPAN sensor network (IPv6 over IEEE 802.15.4) (Ko et al., 2011), RPL, recently standardized. Using this protocol could become common and standard in IPv6 sensor networks in the future. Most implementation of the protocol makes use of the transmission rate successfully (ETX) as metric and focuses on the reliability of links. In this paper we present the use of the residual energy and the transmission delay as a routing metric in the next hop selection process for the RPL protocol. We design an objective function for this metric based on ant colony optimization (ACO), and then we compare the results of experiments realized with the RPL based on ETX.

1. Introduction

During the last couple of years wireless communication has been one of the fastest growing technologies. It has been envisioned that, in the future, small wireless devices will be abundantly scattered everywhere, autonomously exchanging relevant information and creating the so-called “smart dust” [1]. In this context, wireless sensor networks have become a hot topic for research. These small network devices usually have a very limited memory and their practical usage requires minimal power consumption. Energy conservation, while ensuring an adequate level of service, is a major concern in LLNs, because the nodes are typically deployed and are not replaced in case of failure. Several efforts have recently led to the standardization of a routing protocol for low consumption lossy networks [2, 3]. Over the past four years, the working group RoLL IETF has developed a routing protocol for sensor networks 6LoWPAN (IPv6 over IEEE 802.15.4), RPL (routing protocol for Low power and Lossy Networks), recently standardized. The use of this protocol could become common and standard in IPv6 sensor networks in the future, despite some obstacles that slow adoption today. RPL is a flexible protocol, which in theory allows the interconnection of large-scale communicating objects of high heterogeneity, with strong constraints in memory, processing power, and energy resources. This flexibility is paid in complexity: RPL specifications are complex, and existing implementation is generally not complete. The standard provides several criteria that can be used as a routing metric. Most implementation of the protocol makes use of the rate of successful transmission (ETX) as a metric, thus emphasizing the reliability of the links. In this paper we present the use of the residual energy and the transmission delay as a routing metric in the next hop selection process. We design an objective function (QoSRPL) for this metric based on ant colony optimization (ACO), and then we compare the results of experiments performed with RPL based on ETX. The remainder of the paper is organised as follows. In Section 2 we present in detail the functioning of RPL protocol. Section 3 presents the implementation of the protocol under Contiki OS (ContikiRPL). Section 4 presents ACOs and WSN routing. Section 5 presents the design of the proposed objective function, and then we compare the results of experiments performed with the RPL based on...
2. RPL Presentation

RPL is a protocol for routing distance vector for LLNs which describes a method of constructing a logical topology called DODAG (Destination Oriented Directed Acyclic Graph) (see Figure 1) [4] using an objective function and a set of metrics and constraints. The objective function is based on a combination of metrics and constraints to calculate the “best acceptable path” [5]. There could be several objective functions considered by the same node or the same network. The objective function does not need to specify the metrics/constraints, but it dictates rules that are consistent of DODAG formation (e.g., the number of secondary parents and the use of load balancing). The graph built by RPL is a logical topology built on a physical topology to meet specific criteria [6]. A network can have multiple routing topologies (multiple graphs) active at the same time. These topologies are used to carry traffic with different metrics and different constraints [7, 8].

2.1. DODAG Construction. The process of building the DODAG structure begins at the root node, which is usually the data collection node (the Sink) [9]. There could be multiple roots in the configured network. RPL routing protocol specifies a set of new ICMPv6 control messages to exchange information related to the construction of the DODAG structure. The root starts the dissemination of information concerning the structure using the DIO (DODAG Information Object) message. Nodes within communication range of the root will receive and process the DIO message, and then they make the decision to join the structure or not, based on certain rules (depending on the objective function, the characteristics of the DAG, and the cost of the path announced). Once the node is attached to the structure, it has a route to the root of the DODAG structure (see Figure 2(a)). The root is called the parent node. The node calculates its position in the graph, which represents the position of the node in the DODAG structure. If this node is configured to act as a router in the network (which is generally true in WSNs), it starts broadcasting in the vicinity the new information of structure that has joined through DIOs messages. If the node is not configured to be a router then it just joined the DODAG structure and does not send a DIO message. Neighbor nodes receiving this announcement will repeat this process for selecting parent, adding routes, and announcing new information of the DODAG structure using DIOs messages (see Figures 2(b) and 2(c)). This process continues until it covers all network nodes. Each node of the DODAG structure has routing entry to its parent (or several parents according to the objective function) through which this node can reach the root of the DODAG structure. Each node in the graph has a rank that represents the relative position of the node to the root of the DODAG structure (see Figure 2(d)). The notion of rank is used by RPL for various purposes, including the loops avoidance. The different phases of graphical construction are shown in Figure 2.

The DAO (Destination Advertisement Object) messages aim to keep down routes, they are used only for applications requiring traffic point-to-multipoint and point-to-point. After the construction of the logical DODAG structure, when a node has data to send to the root, it sends them to a parent (called the preferred parent). This data structure will rise until it reaches the final destination. This model is the model MP2P (multipoint-to-point) traffic. Messages flow leaf node to root node. Other applications require the presence of traffic in the opposite direction. Trafficking P2MP (point-to-multipoint), the information flows to the leaf nodes. This traffic can come from outside the network, from node root. All this requires a routing table to be constructed at each node and a mechanism to fill these routes. The DAO messages are used to announce the accessibility to nodes that can be potential destinations. A node belonging to the DODAG ETX. Finally, we emphasise the contribution of this paper and future work.
structure sends a DAO message to all parents. Upon receipt of this DAO message, a parent node adds an entry in the routing table and sends a DAO message to all parents (aggregation of information received may be considered). This process continues until the information reaches the root DODAG. It should be noted that this method is called the “operating mode with storage” where all intermediate nodes have a memory for storing routing tables. The RPL protocol also supports another mode called “operation without storage” where no intermediate node stores the routes to the nodes that come to announce with DAOs, and the Sink node then uses a source routing. The RPL protocol also supports traffic point-to-point (P2P) (traffic from any node to another node in the DODAG structure). When a node sends a message to another node in the DODAG structure this message travels towards the root of DODAG to achieve a common ancestor node with knowledge of the route, at which the message is transmitted to the final destination.

2.2. Objective Function (OF). Routes to the DODAGs roots are optimized by an objective function. OF defines how RPL nodes select and optimize routes in a RPL instance [10]. The OF is identified by OCP (objective code point) in the DIO configuration option. The OF also defines how nodes should reflect one or more metrics and constraints in a value called rank, which is approximately the distance from the node to the DODAG root. It also defines how to select parent nodes. The RPL information contained in the data packets includes

Figure 2: Example illustrating the construction of a DODAG.
the rank of the sender to detect inconsistencies and routing loops.

2.3. The Trickle Algorithm. The RPL messages generation is based on a timer. Additionally, nodes configure transmission intervals messages using the Trickle algorithm. Trickle is a timer that allows nodes sharing the transmission medium with high rates of loss as the case of LLNs, exchanging information and data, scalable, saving energy, and ensuring successful transmission. This timer reduces the frequency of control messages transmission, while maintaining coherent information within the network. The basic idea of a Trickle timer is to send frequently more DIOs message when detecting inconsistency in a DODAG. As long as node receives consistent information, it will increase exponentially its transmission interval until a predetermined minimum frequency. For RPL, “consistent” means that DIO message from a sender with a lower rank does not cause any change for the following parameters: parents, the preferred parent, or the rank. Consisting event can occur when there is an event that changes the structure of the DAG; for example, routing loop detection, receiving DIO message indicating a new DODAG version or a new RPL instance, new parent selection or parental relationship has become inaccessible. In the case of such event, the Trickle timer will be reset and the transmission interval is set to a predefined minimum.

The Trickle algorithm runs for a defined interval and has three configuration variables [10]:

(a) The minimum size of the interval, $I_{\text{min}}$: this is a determined value for sending DIOs messages ($I_{\text{min}} = 2I_{\text{DIOIntervalMin}}$), where $I_{\text{DIOIntervalMin}}$ is a value announced by the DODAG root in DIOs messages. This value indicates the smallest interval between two DIOs messages.

(b) The maximum size of the interval, $I_{\text{max}}$: this value describes a doubling of the minimum interval size ($I_{\text{max}} = I_{\text{min}} \times 2^{I_{\text{doubling}}}$) where $I_{\text{doubling}}$ is a value announced by the DODAG root in DIOs messages, indicating how often the interval $I$ may be doubled.

(c) The redundancy constant, $K$: this is a value determined by the DODAG root.

3. RPL in Contiki OS

Contiki [11] is an operating system for wireless sensor networks, dedicated to the Internet of Things. Being able to run on microcontrollers with 64 to 256 KB Flash and 16 to 32 KB of memory RAM, the software loads a network stack IPv6/6LoWPAN and the RPL routing protocol, two technologies for networks, and low-flow devices with limited hardware resources. This operating system has the ContikiRPL protocol which is a reliable implementation of RPL by the Swedish Institute of Computer Science (SICS) which also has developed and implemented the ContikiMAC protocol. The implementation of ContikiRPL consists of several submodules; see Figure 3 [12, 13].

4. ACOs and Routing in WSN

A new approach for routing in WSNs has emerged: this approach is based on algorithms inspired from the ant colonies. These algorithms are based on the ability of simple ants to solve complex problems through cooperation. All methods using this paradigm are now called ACOs. Indeed, the collective intelligence in social insects results in the emergence of intelligent collective behavior due to macroscopic simple interactions at the microscopic level. The operation of ant colonies is the best example. Ant behavior is a collective and collaborative behavior. Each ant is a priority for the welfare of the community. Each individual in the colony is independent and is not supervised (completely distributed system). The colony is therefore autocontrolled by means of mechanisms relatively simple to study. By projection of the behavior of these insects on the characteristics of WSNs, we note that the behavior of ants is well suited to this type of networks, especially when calculating routes [14, 15].

5. Description of the Proposed Approach

Fully distributed nature of such networks is the problem of performance (due to the calculation of routes) as well as problems related to the security of exchanges between nodes. In regard to performance due to packet routing, we distinguish protocols inspired by ant colonies that give in most cases better results. We present a protocol based on an ant colony to calculate dynamic routes and a cooperation
mechanism which allows better quality of service (QoS) management problem in LLNs. The idea is to design an algorithm based on decentralized operation of ants, using their natural ability to find shortest paths between source and destination by moving through the network. We consider the implementation of an objective function specifically optimized for WSNs. We propose an approach that allows the use of the residual energy and the transmission delay as a routing metric of the nodes in the preferred parent selection process for the RPL protocol to build the DODAG structure.

5.1. The Proposed Objective Function. After receiving a DIO from a neighbor, any non-root nodes compute the path cost through this neighbor. The objective function of the proposed work is to find a path from source to destination through a parent with the high probability of transition. The transition probability from source $i$ to destination $d$ across the parent $j$ of $i$ is calculated as follows:

$$P_{ijd} = \frac{[\tau_{ijd}]^\alpha [D_{ijd}]^\beta [MBR_{ijd}]^\delta}{\sum_{l \in N_i} [\tau_{ild}]^\alpha [D_{ild}]^\beta [MBR_{ild}]^\delta},$$

where $\alpha$, $\beta$, and $\delta \geq 0$ are parameters that denote pheromone, delay, and residual energy.

(i) $\tau_{ijd}$ is relative weight of pheromone trail.

(ii) $D_{ijd}$ is the value of the heuristic related to the end-to-end delay.

(iii) MBR$_{ijd}$ is the value of the heuristic related to the residual energy.

Also, $N_i$ is the set of parents of $i$ and $l$ is parent of node $i$ through which a route goes to destination $d$.

While traveling in network the DIO message collects transmission delay of each link and the residual energy at each node. After a node has calculated path cost for all its neighbors and chose best parent in regard to the relation order for the selected metric, node updates its metric container (by computing its path through that parent) and starts to send its own DIOs.

5.2. Calculation of Relative Metrics. For calculating relative metrics, the end-to-end delay and residual energy are considered. The end-to-end delay is additive metric (the delay of a path is the sum of the delay of every hop), and residual energy is concave metric (the residual energy of a path is determined by the link with the minimum available residual energy). Since additive metrics have to be minimized for shortest paths, concave metric is used to maximize residual energy. It is desirable to avoid selecting a node with low energy to increase the lifetime of the network, because the power and energy are considered critical resources in most LLNs. Reciprocal values are used while calculating relative metrics.

5.2.1. The Delay. The delay between the source and the destination is calculated by

$$D_{ijd} = \sum_{l \in \text{route}_j(i,d)} \text{delay}(l),$$

where route$_j(i,d)$ is route from $i$ to $d$ through $j$.

The delay($l$) is the delay from end to end from the source $i$ to destination $d$ through the parent $j$ per DIO message, during DODAG structure construction.

5.2.2. Minimum Residual Energy. To approximate the residual energy, we use the model proposed by Rahmeh and Fourthy [16]. This model is based on simple computations on low memory to fit into sensor nodes, while maintaining the original model accuracy. Based on these approximations, we implemented this model. Following RPL metrics recommendations [17], the minimal residual energy between the source and the destination is given by

$$MBR_{ijd} = \min \{\text{Residual\_Energy}(l)\}$$

$$\forall l \in \text{route}_j(i,d),$$

where route$_j(i,d)$ is route from $i$ to $d$ through $j$.

5.2.3. Pheromone. Initially when there is no parent relation between $i$ and $j$, pheromone on link $(i, j)$ is made as $\tau_{ij} = 0.0$. When $j$ is detected as parent of $i$ through DIO message, an initial pheromone is deposited as $\tau_{ij} = 0.1$.

5.3. Route Reinforcement. When the data transmission is going on, the routes are reinforced positively making it more desirable for further selection. The proposed objective function would not be complete without the process of pheromone evaporation trails. Indeed, to avoid getting suboptimal solutions, it is necessary for the system to “forget” bad solutions. There is also an implicit negative reinforcement for the pheromone, end-to-end delay, and residual energy values. We use updated rules proposed by [18] as follows.

5.3.1. The Pheromone Update. The update of the pheromone value is realized as follows:

$$\tau_{ij} = \begin{cases} (1 - \rho) \tau_{ij}, & \text{if } (1 - \rho) \tau_{ij} > 0.1, \\ 0, & \text{if } (1 - \rho) \tau_{ij} \geq 1, \\ 0.1, & \text{otherwise}. \end{cases}$$

When node $i$ loses its connectivity to its parent $j$, the pheromone on the link $i$ to $j$ will be set to 0.

$\rho$ is a parameter which regulates pheromones, delay, and residual energy conservation from cycle to cycle. $(1 - \rho)$ corresponds to the evaporation factor of pheromone, delay, and residual energy on routes.
5.3.2. The Delay Update. The update of the end-to-end delay value is realized as follows:

\[
D_{ijd} = \begin{cases} 
\rho \cdot \text{delay}(l) + (1 - \rho) \cdot D_{ijd}, & \text{if } l \in \text{route}_j(i, d), \\
(1 - \rho) \cdot D_{ijd}, & \text{otherwise}.
\end{cases}
\]

5.3.3. The Residual Energy Update. The update of the residual energy value is realized as follows:

\[
MBR_{ijd} = \begin{cases} 
\rho \cdot \text{Residual Energy}(l) + (1 - \rho) \cdot MBR_{ijd}, & \text{if } l \in \text{route}_j(i, d), \\
(1 - \rho) \cdot MBR_{ijd}, & \text{otherwise}.
\end{cases}
\]

5.4. Calculation of Rank. The rank of a node is a scalar representation that gives its position in a DODAG version and has certain properties. Calculating the rank is considered among features of the objective function. This calculation depends on several factors, for example, all parents, metric links, and nodes configuration. The rank is strictly (monotonically) incremented and can be used to validate a progression from the root. It is also used to prevent and detect routing loops. The calculation of the rank maintains the following properties. The rank value decreases toward the upward direction (destination to the root node) and increases in the opposite direction. Furthermore, the rank provided by a node must necessarily have a greater value than all values of ranks presented by his parents. In this case, there is no risk of creating a loop. A network with nodes with similar rank values can trigger a loop when a node chose a path to a node with the same rank. Once a node \(i\) (nonroot) has chosen its preferred parent \(j\) (the high probability), node computes its own rank from preferred parent's rank as follows:

\[
\text{Rank}(i) = \text{Rank}(j) + \frac{1}{\text{Rank}(j) + P_{ij}},
\]

where \(P_{ij}\) corresponds to the high probability; \(i\) selected \(j\) as the preferred parent.

6. Demonstrating Scenario

First, when the node wakes up it initializes RPL, resets the proper timers, and sends DIS message soliciting a DIO message from the nearest neighbor. Another node will receive this DIS message and replies with a DIO message. The first node in turn will receive the DIO message and processes this message in order to join the DAG of the DIO sender and choose the best parent of the candidate parents. Once the node decides that it will join that DAG it chooses a preferred parent, according to the proposed OF; and knows its default route towards the DAG root. Then, it will send a DAO message upwards to advertise itself in the network and it will be acknowledged by DAO_ACK. Figure 4 shows the sequence diagram of the above scenario [19, 20].

Once the transmission of data is in progress, the paths are reinforced positively making it more desirable for further selection. Also when session is going on, the load on the selected path may increase causing more delay and less available bandwidth. Nodes might fail (exhausted their energy) causing link rupture. In such case, the path preference probability will automatically decrease and hence alternate routes can be used which are found during DAG construction phase. The alternate routes are also periodically checked for their validity even though they are not currently used.

7. Simulation Environment

For the evaluation of our approach we use the simulator Cooja [21]. This simulator is available in Contiki operating system for wireless sensor networks. We use the sky platform which is an emulation of TelosB mote. The energy model used by the Contiki OS takes into account the energy consumption of listening state, transmission, and rest of the radio component.

The following tools have been used during the development of this paper:

(i) Instant Contiki is an Ubuntu image with preinstalled tools to develop a Contiki system.

(ii) VMware Instant Contiki image has been mounted in VMWare Player (v.3.1.3) [22].

(iii) Wireshark is a network protocol analyser. It can capture, display, and analyse packets on selected interfaces (packet sniffer) [23].

(iv) Gnuplot is used to generate plots and statistical graphics [24].

Cooja simulator generates capture files in pcap format when executing. There are several tools for analysing and interpreting the results of these traces. However, the most popular program that can read and display this format is Wireshark. Figure 5 shows a capture treated with the development version of Wireshark, knowing that this version includes the RPL protocol and is properly interpreted. In this figure, contained trace displayed by Wireshark shows a set of information.
Figure 4: Sequence diagram of demonstrating scenario.

Figure 5: Viewing a trace with Wireshark.
packets sent between nodes. These RPL messages sent both sides indicating the type value of these ICMPv6 control, the code messages, the version, the sequence number, the rank value, DODAGID address, and so forth.

### 8. Results and Discussion

We evaluated the performance of RPL with the proposed objective function (QoS\_RPL) in a topology of 20 sensors, and then we compare the results of experiments with RPL based on ETX. Table 1 summarizes Contiki OS and Cooja parameter.

In all scenarios the nodes are static, and they are deployed randomly. The Sink is the node with identifier “1” (Figure 6). Each node periodically sends UDP packets with a success rate Tx/Rx 80\% [25, 26].

In simulations, ACO parameter settings are defined in Table 2, which were experimentally found to be good by [27, 28].

The simulations were performed for fixed period, to show that our proposal increases the lifetime of the network and distributes energy uniformly between nodes without a notable lack of precision in the transmission. The proposed approach was compared with RPL based on ETX. For both cases the Sink collects data generated by nodes. Then we evaluate the energy consumption of nodes, the rate of packet delivery, and the end-to-end delay. Table 3 summarizes protocol parameter.

#### 8.1. Energy Efficiency

We begin with the evaluation of energetic cost for the construction of DODAG structure phase and the data collection phase. We compare the energy consumption of our approach (QoS\_RPL) and RPL based on ETX.

QoS\_RPL routing aims to use nodes with higher remaining power level; thus these nodes drain their battery more quickly and further become less attractive to relay data. The network should be reorganized to find more interesting nodes for routing and so on; thereby a balance on all nodes battery levels should occur. This can be seen in Figure 7 which presents the proportion of nodes in the network with...
the corresponding percentage of remaining energy at the end of the simulation.

Figure 7 shows that QoS_RPL consumes less energy compared with ETX (about 15%), which leads to the fact that nodes will completely exhaust their energy, causing holes in the network. This is an important point, because the network integrity may be affected if some nodes are stopped. The energy consumed in ETX is higher than QoS_RPL; thus increasing the lifetime of the network is approximately 10%. We can conclude that the proposed approach performs better than ETX and increases the lifetime of the network.

8.2. Transmission Performance. We also assessed the accuracy of the routing for data collection. Routing based on routes with ETX promotes a higher rate of packets delivery, while this is not the case for QoS_RPL.

It is therefore not surprising that the number of packets received with ETX is slightly higher compared to our approach, as shown in Figure 8, which shows the total number of packets received from each node to the Sink. The rate of received packets of ETX based routing is better than QoS_RPL, but this difference is only about 5%.

8.3. Improve End-to-End Delay. We define the end-to-end delay as the average time required for an application message from a source node (sensor) to reach the final destination (Sink node).

As can be seen in Figure 9, the proposed QoS_RPL performs better than RPL based on ETX in terms of average end-to-end delay. Even though the routing path may be longer than RPL based on ETX, proposed scheme tries to route packets along a path that satisfies QoS requirements such as low delay paths. In Figure 9, we can observe that, in the case of QoS_RPL, almost all nodes improve their end-to-end delay to the root of the DAG. As expected, the delay has not diminished for nodes in the vicinity of the root (one hop) of the DAG, that is, nodes 2, 3, 4, and 7 (see Figure 6). It can be observed that the delay decreases significantly for the nodes in the periphery, especially those located far from the root of the DAG. For example, the nodes 14 and 19 improve their delay of approximately 40%. These nodes follow different path but it is faster than that in case of RPL based on ETX.

9. Conclusion and Future Work

Routing protocol designated for Low power and Lossy Networks is the subject of several research studies and tests for the evaluation of its performance in multiple platforms because the sensors are integrated increasingly in all areas. These studies are needed to improve the mechanism of RPL within such networks and to develop features adapted to the resource constraint of nodes and provide data transmission in the best conditions.

In this paper, we have focused on the RPL routing protocol, a solution designed to match the requirements of networks characterized by low power supplies and by deployment in lossy environments. Specifically, we have presented a new, energy, and delay efficient RPL routing protocol for WSN based on ACOs, in which energy and delay aware routing metrics and information on resources availability of sensors are used to improve the energy efficiency of RPL. To evaluate the performance of the proposed routing protocol, we have implemented a prototype version of the proposed architecture in Contiki OS. Specifically, the WSNs using the basic RPL protocol and the proposed one have been simulated in Cooja, a flexible Java-based simulator designed for WSNs running the Contiki operating system. Numerical results have shown that the proposed QoS_RPL
has better performance than the basic RPL in terms of energy usage and network lifetime network and distributes the energy uniformly between nodes without a notable lack of precision during the transmission and the end-to-end delay is improved. Despite having discussed the efficiency of the proposed enhanced version of the RPL protocol, this study represents just one first cut analysis and further investigation is required. Firstly, providing additional decision criteria to help routing decisions in WSNs, additional tests can be performed to verify the efficiency of the proposed solution through experimental testbeds deployed in realistic scenarios.

Conflict of Interests

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

References

[1] J. Ko, A. Terzis, S. Dawson-Haggerty, D. Culler, J. Hui, and P. Levis, "Connecting low-power and lossy networks to the internet," IEEE Communications Magazine, vol. 49, no. 4, pp. 96–101, 2011.

[2] C. Bormann, J. P. Vasseur, and Z. Shelby, “The internet of things,” IETF Journal, vol. 6, no. 2, 2010.

[3] P. Kinney and Kinney Consulting, Overview of Mesh Networking over IEEE 802.15.4, IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs), 2013.

[4] T. Winter, P. Thubert, A. Brandt et al., “RPL: IPv6 Routing Protocol for Low power and Lossy Networks,” Draft-ietf-roll-rpl-19, March 2011.

[5] S. Zheng, "Performance study of cooperative routing metric for multi-hop wireless networks," in Proceedings of the 73rd IEEE Vehicular Technology Conference (VTC Spring ’11), Budapest, Hungary, May 2011.

[6] M. Stehlík, Comparison of simulators for wireless sensor networks [M.S. thesis], Masaryk University, Brno, Czech Republic, 2011.

[7] K. Nahrstedt, "Internet of mobile things: challenges and opportunities,” in Proceedings of the 23rd International Conference on Parallel Architectures and Compilation (PACT ’14), Edmonton, Canada, August 2014.

[8] A. Petrescu, N. Benamar, and T. Leinmueller, ”Transmission of ipv6 packets over ieee 802.1p networks,” IETF draft-petrescu-ipv6-over-8021p-02, December 2014.

[9] P. V. P. Reddy, K. M. I. Ali, B. Sandeep, and T. Ravi, "Importance and benefits of IPv6 over IPv4: a study," International Journal of Scientific and Research Publications, vol. 2, no. 12, 2012.

[10] Contiki site, http://www.contiki-os.org/.

[11] http://sourceforge.net/projects/contiki/files/Instant%20Contiki/.

[12] W. Tang, Z. Wei, Z. Zhang, and B. Zhang, "Analysis and optimization strategy of multipath RPL based on the COOJA simulator," International Journal of Computer Science Issues, vol. 11, no. 5, p. 27, 2014.

[13] M. Belghachi and M. Feham, "QoS based on ant colony routing for wireless sensor networks," International Journal of Computer Science and Telecommunications, vol. 3, no. 1, 2012.