Resource Oriented and Energy Efficient Routing Protocol for IPv6 Wireless Sensor Networks

Antimo Barbato
Department of Information Technology and
Mathematical Methods University of Bergamo, Italy
Email: antimo.barbato@unibg.it

Marica Barrano, Antonio Capone and Nicolò Figiani
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Italy
Email: {marica.barrano, antonio.capone, nicolo.figiani}@polimi.it

***Abstract*—In the near future, the IPv6 protocol is expected to provide internet connectivity to any object embedding a communication device, by creating the so-called *Internet of Things* (IoT). In this scenario, IPv6 Wireless Sensor Networks (WSNs) have a key role since they can be used to collect several environment information, hence becoming the eyes, ears and nose of the IoT. Since wireless sensors are limited in power, it is essential to design energy efficient WSNs protocols. To this purpose, in this paper, we propose a Resource Oriented and Energy Efficient (ROEE) routing protocol based on the Routing Protocol for Low power and Lossy networks (RPL). ROEE RPL is intended as the very first building block to achieve the so called IoT. Simulation results show that our protocol has better performance than the basic RPL in terms of energy efficiency without compromising the network throughput.**

***Index Terms*—Sensor Networks, Energy Efficiency, Routing Protocol, RPL.**

I. INTRODUCTION

The growth and evolution of WSNs experienced during the last decade has made possible to develop and deploy inexpensive and self-adaptive monitoring systems composed of multifunctional and distributed wireless sensors [1]. WSNs have significant advantages over traditional communication technologies such as rapid development, low cost, flexibility and aggregate intelligence through parallel processing. The IETF has proposed different standards as methods to interconnect sensor nodes and the Internet by bringing the Internet Protocol version 6 (IPv6) into WSNs [2]. With this solution, in fact, sensors can be natively addressed and connected through the Internet Protocol with several advantages such as plug-and-play installation, simplified development of applications and compatibility with existing architectures. The vision of attaching tiny devices to every single object is known as the *Internet of Things* (IoT) [3]. In the IoT vision, everyday physical objects cab be connected to the Internet and are able to autonomously communicate with other devices. In such a scenario, IPv6 wireless sensors play a key role since they can be used to collect several environment information, hence becoming the eyes, ears and nose of the *Internet of Things*.

The field of IoT, due to the powerful implications it could bring, is a research area of widespread interest. In particular, WSNs have received great attention since they have some limitations that may prevent their diffusion in IoT frameworks. One of the the most crucial constraints in the design and development of sensor networks is represented by their energy consumption: energy is a very critical resource of sensors since WSN nodes are often battery powered and it is usually very difficult to recharge or change batteries. For this reason, because of the limited energy availability, it is of fundamental importance to develop WSNs protocols that optimize sensors energy consumption. Moreover, in the IoT scenario, WSNs can be deployed over large area, therefore also the issue of multi-hop routing must be properly addressed. For these reasons, in this paper, we propose an energy aware version of the the Routing Protocol for Low power and Lossy networks (RPL) [4], one of the most widespread multi-hop routing protocols for WSNs. RPL, defined by the IETF community through the ROLL working group, is a well-known routing protocol for constrained devices, and it has been implemented in the most common operating systems for embedded networked sensors (e.g. Contiki[5] and TinyOS[6]).

The default metric used by the basic implemented version of RPL to build up routing tables is called Expected Transmission count (ETX). It is a link metric that estimates the number of re-transmissions required to successfully send a data packet between two nodes and is a measure of the quality of the channel. In this paper, we extend the RPL protocol by defining a new metric that aims to be resource oriented and more energy efficient with respect to the basic version of RPL. In several IoT scenarios in which WSNs are used, applications (e.g. smart city environment, security, traffic, light monitoring services) can request a certain WSN to monitor a specific resource (e.g. temperature, light and, more in general, environmental and physical parameters). This request is transmitted to sensors by the network sink node that is directly connected to the application data collection center. WSN nodes, in a cooperative way, reply back by providing the requested measurements to the sink node. These data can be monitored both periodically and on-demand, depending on the application and context. Moreover, several resources can be monitored at the same time. As a matter of fact, in heterogeneous WSNs and in reference to each resource request by an IoT application, only a subset of nodes can provide the requested measurements. For this reason, it can be convenient to define multiple routing topologies (as provided in RPL specifications) in the case of multiple resource requests and to assign different roles to nodes in the routes creation phase depending on their monitoring features (i.e. resource availability). In particular, in defining paths towards the root in reference to a specific request, a key routing role must be assigned to nodes that can monitor the requested resource. In our work, we address this aspect, hence defining a resource oriented routing protocol. Moreover, we also focus our attention on defining energy aware routing node-metrics, based on the residual energy and power vulnerability of nodes, so as to reduce the energy consumption of the whole network. By considering these two crucial aspects, we propose a new, energy aware and resource oriented optimized version of RPL, which is called Resource Oriented and Energy Efficient (ROEE) RPL.

The remainder of this paper is organized as follows. In Section II we review the RPL protocol. In Section III we describe the basic characteristics of the routing protocol that we propose to optimize the energy consumption of WSNs in IoT scenarios. Section IV reports some numerical results, obtained through simulations, to evaluate the impact of our optimized routing protocol in terms of network performances with respect to the basic RPL protocol. Finally, in Section V, the paper is concluded and further developments are discussed.

II. ROUTING PROTOCOL FOR LOW POWER AND LOSSY NETWORKS

A Low power and Lossy network (LLN) consists of a multitude of constrained nodes, with limited processing power and memory, short range wireless communication channels and low data rate, interconnected by lossy links that are usually unstable. The traffic patterns of such networks are mainly Point-to-Multipoint (P2M) or Multipoint-to-Point (M2P). Based on these features, the scientific community has produced big efforts to offer a routing solution suitable for this type of networks. In particular, the IETF ROLL working group has defined, over the years, specific requirements for a LLN routing protocol, giving rise to RPL. RPL routes are optimized for traffic to, or from, one or more roots that act as the sink of the routing topology. Specifically, in order to define routing paths, this protocol creates Destination Oriented Directed Acyclic Graphs (DODAGs), each one associated with a specific sink node and characterized by a proper routing Objective Function (OF). The OF identifies a routing performance objectives (e.g. low delay, high throughput, energy efficiency) as well as the specific routing metrics to use to determine link costs.

RPL uses several identifiers to build and maintain the routing topologies:

- *RPLInstanceID*: it is a unique identifier within a network, that identifies a set of one or more DODAGs, and is characterized by a specific objective function. The set of DODAGs, identified by the same RPLInstanceID, is called RPL instance.
- *DODAGID*: it is the identifier of a DODAG root and it's unique within a RPL instance.
- *Rank*: it is a scalar number which represents the link (or node) cost. The exact way how the rank is computed depends on the OF and it is often application-dependent.

Upward routes

RPL provisions routes towards DODAGs roots, by defining acyclic graphs that are optimized according to a given objective function. Each OF defines how RPL nodes have to choose their parents and how they select and optimize routes within a RPL instance. As a result, the main role of an OF is to define the mechanism used to translate metrics and constraints into rank values. Nodes construct and maintain the topologies through particular signalling packets named DODAG Information Object (DIO) messages. These packets carry information that allows a node to discover a RPL instance and its parameters, join a DODAG and maintain the network topology.

Upward routes, supporting multipoint-to-point traffic, are identified by means of defining DODAGs. The macro-steps involved in the definition of a DODAG by a root node are the following:

1) DODAG formation is started by the root node by sending DIOs messages. These messages contain the rank of the sender of the DIO;
2) When a node receives a DIO, it computes its own rank based on the OF and on the rank of the DIO sender. Moreover, it updates the rank field of the message and forwards the DIO to other sensors. In this way, DIOs are propagated through the network;
3) Each node identifies the parent set by selecting neighbours from which a DIO has been received and characterised by lower ranks. Within the parent set, a preferred parent is identified (based on routing metrics) to be the preferred next hop in upstream routes (i.e. from the nodes towards the root), hence identifying the DODAG.

After a DODAG has been defined, it is updated and maintained mainly by means of DIO massages. In particular, RPL supports mechanisms which can be used for local repair and loop detection within DODAGs.

Downward routes

The RPL routing algorithm uses another type of signalling packets, called Destination Advertisement Object (DAO) messages, to propagate destination information upwards along the graph in order to establish downward routes (i.e. from the root towards the other nodes). Downward routes support point-to-multipoint data flows, from the DODAG root to the lowest-ranked nodes and also Point-to-Point (P2P) flows where messages firstly reach the sink through an upward route and then the proper destination through a downward route.

In the downstream routes definition, each node sends a DAO to the preferred parent(s). In turn, the parent(s) forwards the DAO to its preferred parent(s), and so on until it eventually reaches the root. In this way, the downward routes are created. In RPL, two modes are available to store and manage downward routes. In the first mode, called *storing*, all nodes must store downward routing information for their DODAG in a local table. On the other hand, in the second mode, called *non-storing*, nodes do not store downward routing tables and the DODAG root is the only one in charge of storing downstream routing table entries.

Routing Metrics

A routing metric is a quantitative value that is used to evaluate the path cost, therefore the best path is the one that satisfies all the supplied constraints (if any) and that has the lowest cost with respect to some specified metrics. Metrics and constraints are advertised in the DIO packets and the set of routing parameters, used by the RPL instance, is signalled along the DAG, which is built according to the particular OF. In the RPL implementations available in the literature (TinyRPL for the operating system TinyOS and ContikiRPL for Contiki), two OFs [7], [8] are defined and implemented based on two routing metrics:

- Expected transmission count (ETX): it represents the number of transmissions (eventually including retransmissions) a node expects to make in order to successfully deliver a packet to its destination. This metric is used as a cumulative link metric, therefore its value is the sum of the same metric computed on all the link previously used. The expected transmission count takes into account two parameters:
  - P_f: represents the probability that a data packet is successfully delivered to the receiver;
  - P_r: is the measured probability that the acknowledgement packet is successfully received;

  and, since each transmission attempt can be considered as a Bernoulli trial, it is generally computed according to the specific formula:

  $$ETX = \frac{1}{P_f \cdot P_r} \tag{1}$$

  In this case, in defining DODAG graphs, each node selects as parent the sensor with the minimum value of the expected transmission count.
- Hop Count (HP): it represents the number of traversed nodes along the path. In this case, the definition of the optimal path is made by taking into account the number of hops required to reach the destination. Specifically, in constructing DODAG graphs, each sensor selects as its parent the node with the minimum hop count and no energy optimization mechanism is considered to route packets.

These metrics, used by the default RPL version, are not able to save energy of nodes, although in conditions of an ideal transmission channel, they allow identifying optimal paths.

III. ROEE RPL PROTOCOL

In the current implementation of RPL, the most relevant objective function is based on the ETX metric as link cost, whereas the other proposed OF uses the hop-count and is not very useful to deploy efficient WSNs. Since an energy-aware RPL implementation is missing, in this paper we propose and implement it to extend the available features of this routing protocol for low power and lossy networks.

In order to define this energy-aware OF, it is required to use a node metric that takes into account nodes energy consumption and usage. To this end, we have combined two metrics discussed in the literature: Energy consumption (E_c) and Battery Index (BI).
The energy consumption [9] represents the amount of energy used by a certain node and can be expressed as:

$$E_c = \frac{B_{cap} - B_l}{B_{cap}} \tag{2}$$

where B_{cap} represents the battery capacity of the sensor and B_l is the current battery charge level. This metric can be used to try and set up routes involving nodes with the highest residual energy, so as to increase the life of the whole network. The advantage of this metric is that it is easy to compute and low CPU resources are required. However, the values of this parameter have little meanings unless they are compared with each other. For this reason, in optimizing routing graphs, the global knowledge of all node information is required. Moreover, nodes have to periodically monitor and report to the sink their energy consumption, therefore the signalling frequency can be quite high.

The other metric that we have used in defining our protocol is the battery index [10] [11] that represents how much prone a node is to consume energy, depending on its position within the network and other factors. It can be computed based on the power used by the node in each of the following four states in which a node can operate: Transmission (TX), Reception (RX), idle and sleep (actually other states are involved in the life-cycle of nodes, but almost all the energy consumed by nodes is associated with those four states). Let T_{state} and ω_{state} be, respectively, the amount of time a node has spent in that particular state and the corresponding energy consumed in that state per unit of time. The battery index can be computed, for each node, with the following formula:

$$BI = K' \cdot \frac{\omega_{TX} \cdot T_{TX} + \omega_{RX} \cdot T_{RX} + \omega_{idle} \cdot T_{idle}}{T_{sleep}(t) \cdot D_c} \tag{3}$$

where K' is a normalization factor and D_c is the duty cycle, that is the ratio between the Active Time (ACT) and the sum of the active and the inactive (i.e. Low Power Mode (LPM)) time of a node:

$$D_c = \frac{T_{ACT}}{T_{ACT} + T_{LPM}} \tag{4}$$

In using this metric, we have simplified the equation (3) so as to reduce the sensors CPU usage. In particular, since radio communication is the main energy consumer of a sensor battery, $w_{idle} \cdot T_{idle}$ can be ignored. Moreover, in this paper we have used sensors Tmote Sky using a CC2420 chip whose transmission and reception consumptions are almost the same [12]. For this reason, w_{TX} and w_{RX} can be considered equal and equation (3) can be modified as follows:

$$BI = K \cdot \frac{T_{TX} + T_{RX}}{T_{sleep} \cdot D_c} \tag{5}$$

where K is a new normalization factor that incorporates ω_{TX} and ω_{RX}. The battery index can be used to detect the nodes that are prone to run out of energy. In particular, vulnerable nodes are those having significantly high BI values, since the larger the BI, the more vulnerable to energy consumption the node tends to be. The main disadvantage of this metric is that it requires more CPU resources than the energy consumption case. However, the BI metric is more stable in time so that a lower signalling frequency is required.

In addition to the energy consumption and battery index metrics, we have also used the resource availability information to define the rank of each node. In fact, our routing protocol intends to be resource oriented which means that in defining the DODAG paths associated with a certain data request by an IoT application, network nodes must play different roles in the routing topology depending on their features. In particular, a key routing role has to be assigned to nodes that can monitor the requested resource by means of decreasing their ranks so to have more chances to become parents of other nodes. To this end, for each node, we define the binary Resource Availability (RA) parameter as follows:

$$RA = \begin{cases} 1 & \text{if the node can retrieve the requested resource} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

By combining the energy consumption and battery index metrics with the resource availability information, we have defined a new version of the RPL protocol, called Resource Oriented and Energy Efficient (ROEE) RPL, whose goal is to define the routing RPL topology by assigning a more important role to motes that can reply positively to requests made by the application and by improving the energy efficiency of the network by means of using energy-aware metrics. In this new version of the RPL protocol, the rank of each node, R, is computed as follows:

$$R = \frac{E_c \cdot BI}{N} - RA \tag{7}$$

where N is a normalization factor defined based on experimental tests. The battery index is computed according to the equation (5), using shift operators whenever possible to reduce the sensors CPU usage. This value is then multiplied by the estimated energy consumption, so that each node uses as energy-aware metric the consumed energy weighted for the battery index. Finally, the RA binary parameter is subtracted from the result of the previous computation. The normalization parameter, N, is introduced to make parameters $E_c \cdot BI$ and RA comparable to each other. At the end of the DIOs propagation described in Section II, each node selects as parent the sensor with the lowest rank among neighbour nodes. As a consequence, sensors with low energy consumption, low BI value and that are able to measure the requested resource are the most preferable nodes in the selection of RPL DODAG parents.

IV. NUMERICAL RESULTS

To evaluate the performance of the ROEE RPL protocol, a simulation campaign has been carried out. In our tests, in fact, we have relied on simulations mainly to test the proposed solution in several testing scenarios and evaluate several parameters. The purpose of simulations was to evaluate the performance of the proposed routing protocol with respect to the basic version of RPL both in terms of energy consumption and network performance.

In our tests, we have used the Contiki operating system, a widely used open source, portable and multi-tasking operating system designed for memory-efficient networked-embedded systems and wireless sensor networks [13]. Contiki, in particular, contains μIPv6, a full IPv6 stack including a comprehensive Application Programming Interface (API) for programming protocols using User Datagram Protocol (UDP), Transmission Control Protocol (TCP) or Internet Control Message Protocol version 6 (ICMPv6). In our tests, a specific version of Contiki OS has been used, called ContikiRPL, containing a RPL implementation. The main goal of ContikiRPL is to provide a versatile and simple programming interface that can be used to study objective functions. In order to test the above stated ROEE protocol, we have implemented a new version of ContikiRPL, called ContikiROEE-RPL, implementing the proposed routing protocol.

In order to simulate the sensors network, we have used Cooja, a flexible Java-based simulator designed for WSNs running the Contiki operating system [14]. Cooja, in particular, simulates networks of sensor nodes where each node can be of a different type in terms of software and hardware.

In our tests, we have simulated a 31 node WSN (node 0 is the sink sensor), randomly placed in an area of $200m^2$. Even if Cooja can be used to simulate networks composed of different types of sensors, we have decided to use the same simulated platform for all sensors, the Tmote Sky, also named Telos B, a ultra low power IEEE 802.15.4 compliant wireless sensor module based on a TI MSP430 and Chipcon CC2420 radio. For this reason, in the simulated network, the MAC and PHY layers are compliant with the IEEE 802.15.4 specifications, while the IPv6 RPL/ROEE RPL is used to implement the Network layer. Moreover, every sensor but the sink, has a UDP application agent used to periodically send (i.e. every 10 seconds) a UDP data packet to the sink node. For this reason, the sink sensor, is both a RPL router and a UDP server. The channel model used in our simulation is the Unit Disk Graph Medium (UDGM) distance loss model in which the transmission range is modelled as a disk. All nodes behind that disk do not receive packets while the nodes within the transmission distance receive all the packets. The disk radius depends on the transmission power and, in our tests, is the same for all sensors (i.e. 30 meters). The UDGM-distance loss model also considers the interferences even if in a very simple manner: if packets interfere they are lost. As a consequence, all communications running at the same time are unsuccessful. Simulation time for each test case was set to 72 minutes and repetitive simulations for each scenario were performed to verify the reliability of our results. Specifically, two different scenarios have been simulated:

- *100% Resource availability*: all 30 sensors periodically transmit best effort UDP packets, containing the requested information, to the root node;

- *50% Resource availability*: only 15 sensors are able to monitor the requested resource, hence periodically transmitting best effort UDP packets, containing the measures, to the root node.

Test results in terms of network lifetime are represented in Figure 1 and Figure 2. In Figure 1, in particular, we represent the results obtained with RPL and ROEE RPL in terms of number of nodes alive as a function of the simulation time in the *100% Resource availability* scenario. Figure 1 shows that after about 2800 seconds all the sensor nodes of the basic ContikiRPL network have already run out of energy and only the sink mote, which is not battery powered, is still alive. On the other hand, in the ROEE RPL implementation proposed in this paper, 14 motes (including the root node) are still alive at the end of the simulation which lasted nearly 4300 seconds (about 72 minutes). Therefore, by employing our version of the protocol, the life of the network can be substantially increased. The same consideration can be done for the *50% Resource availability* scenario, whose results are represented in Figure 2. Specifically, in the RPL case, there is almost no difference between the average lifetime of resource (i.e. nodes with $RA = 1$) and non-resource nodes (i.e. nodes with $RA = 0$). On the other hand, in the ROEE RPL implementation, at the end of the simulation there are 4 alive resource nodes and 8 non-resource motes. This different behaviour is due to the fact nodes with $RA = 1$, have lower ranks, so they are more involved in the network routing operations.

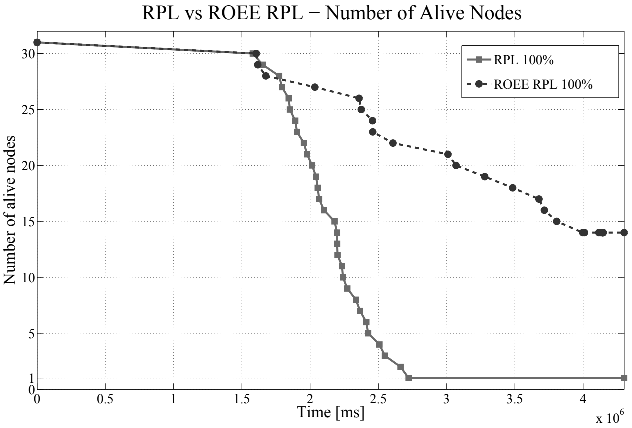

Figure 1: Number of alive nodes with RPL and ROEE RPL in the *100% Resource availability* scenario.

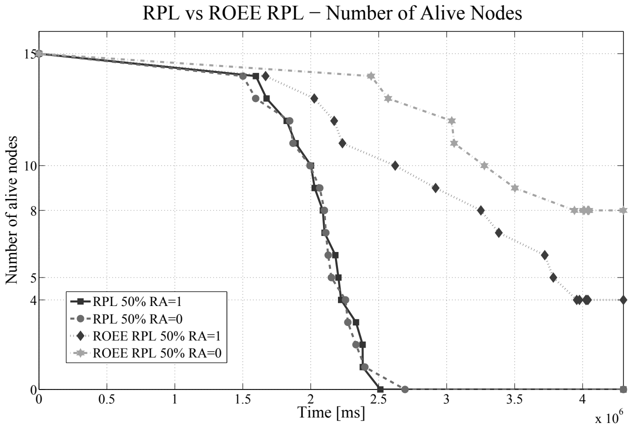

Figure 2: Number of alive nodes with RPL and ROEE RPL in the *50% Resource availability* scenario for resource and non-resource nodes.

In Figure 3 we show the numerical results obtained with RPL and ROEE RPL in terms of cumulative energy consumption of the network as a function of the simulation time, both in the *100% Resource availability* an *50% Resource availability* scenario. Results show that the ETX based RPL is the most energy intensive RPL routing protocol. Specifically, for the two standard versions of ContikiRPL, the energy consumption rapidly increases, reaches its maximum value and then it gets stabilized since all motes but the sink run out of energy. On the other hand, the energy consumption of the two ROEE RPL versions grows very smoothly, almost linearly and, at the end of the simulation, it reaches a smaller value of total power consumption than in the corresponding RPL case so that the entire network can last longer.

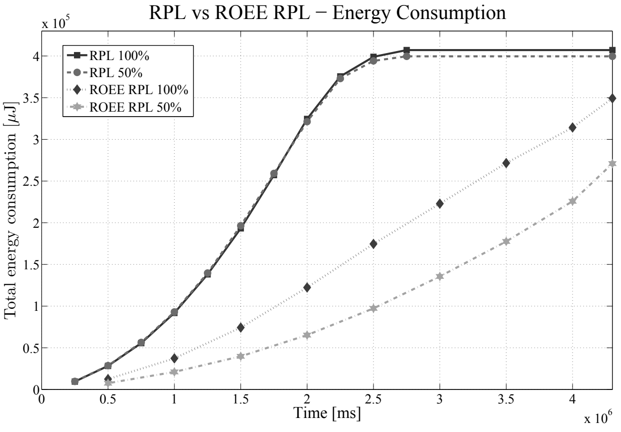

Figure 3: Cumulative energy consumption with RPL and ROEE RPL in the *100% Resource availability* and *50% Resource availability* scenarios.

Results in terms of sensors energy consumption are also represented in Table I in which the mean values and the standard deviation of the sensors energy consumption are computed for each considered test scenario. Numerical data show that in the *100% Resource availability* scenario, the proposed protocol allows saving around 31% of the WSN consumption with respect to the corresponding RPL basic network. This saving is even higher (i.e. 48%) in the *50% Resource availability* case because of the resource-oriented feature of the ROEE RPL routing metric.

Finally, in Figure 4, we show the results obtained in terms of throughput as a function of the simulation time. Results

Table I: Sensors energy consumption mean value and standard deviation with RPL and ROEE RPL in the 100% Resource availability and 50% Resource availability scenarios.

Scenario	**Mean value [μJ]**	**Standard deviation**
RPL 100%	1167	283.5
ROEE RPL 100%	802.7 (-31%)	310.8
RPL 50%	1180	310.9
ROEE RPL 50%	604 (-48%)	226.3

show that there's no significant variation of this parameter between the basic RPL version and the one proposed in this paper. Although the new protocol requires more processing at the application layer, the network throughput is just slightly affected. Consequently, in the ROEE RPL cases, very few packets must be retransmitted and a very small amount of energy consumption has to be used to this end.

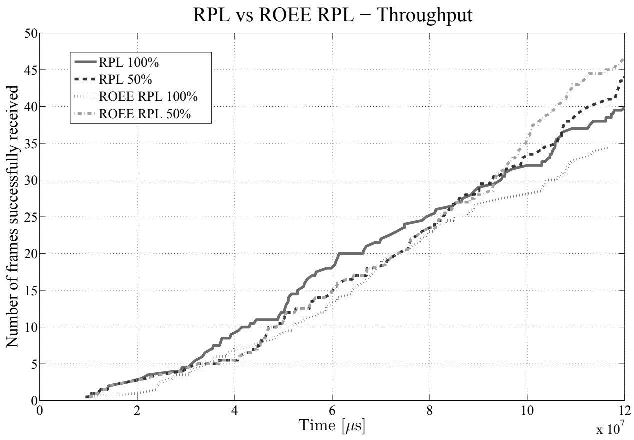

Figure 4: WSN throughput with RPL and ROEE RPL in the *100% Resource availability* and *50% Resource availability* scenarios.

V. CONCLUSIONS

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 aware and resource oriented optimized version of RPL, called Resource Oriented and Energy Efficient (ROEE) RPL, in which energy aware routing metrics and information on resources availability of sensors are used to improve the energy efficiency of RPL. In our protocol, we have combined two energy-aware routing metrics, Energy Consumption and Battery Index, which represent, respectively, the energy consumed by each sensor and the vulnerability of each node in consuming power. The resulting new metric has been further extended to take into account the resource availability of each node with respect to data requests of IoT applications. As a result, ROEE RPL uses an energy aware routing metric which takes advantage of the capability of motes to retrieve a given resource requested by the application. 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 ROEE RPL has better performance than the basic RPL in terms of energy usage and network lifetime without compromising the network throughput.

Although 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, additional tests can be performed to verify the efficiency of the proposed solution through experimental testbeds deployed in realistic IoT use-case scenarios. Moreover, ROEE RPL can be further improved by designing a mechanism in which the algorithm itself can weight the resource availability metric and the energy aware metrics depending on the context and application, in a self-adaptive way.

ACKNOWLEDGMENT

This work was partially supported by Italian MIUR in the framework of the PRIN Gatecom project.

REFERENCES

[1] Waltenegus Dargie and Christian Poellabauer. *Fundamentals of wireless sensor networks: theory and practice*. Wiley, 2010.
[2] Jonathan W Hui and David E Culler. Ip is dead, long live ip for wireless sensor networks. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 15–28. ACM, 2008.
[3] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.
[4] Tim Winter. Rpl: Ipv6 routing protocol for low-power and lossy networks. 2012.
[5] Contiki OS web site. http://www.contiki-os.org, 2013.
[6] TinyOS web site. http://www.tinyos.net, 2013.
[7] Pascal Thubert. Objective function zero for the routing protocol for low-power and lossy networks (rpl). 2012.
[8] Omprakash Gnawali. The minimum rank with hysteresis objective function. 2012.
[9] V. Shnayder, M. Hempstead, B. Chen, G.W. Allen, and M. Welsh. Simulating the power consumption of large-scale sensor network applications. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 188–200. ACM, 2004.
[10] F. Kerasiotis, A. Prayati, C. Antonopoulos, C. Koulamas, and G. Papadopoulos. Battery lifetime prediction model for a wsn platform. In *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, pages 525–530. IEEE, 2010.
[11] Joan Cortés, Qi Wang, and John Dunlop. Novel metric for identifying energy-vulnerable nodes and corresponding proactive schemes in wireless sensor network. In *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, pages 1–6. IEEE, 2009.
[12] L. Paradis and Q. Han. A survey of fault management in wireless sensor networks. *Journal of Network and Systems Management*, 15(2):171–190, 2007.
[13] M. Dohler, D. Barthel, T. Watteyne, and T. Winter. Routing requirements for urban low-power and lossy networks. 2009.
[14] A. Dunkels, B. Gronvall, and T. Voigt. Contiki-a lightweight and flexible operating system for tiny networked sensors. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 455–462. IEEE, 2004.