ALABAMO: A LoAd BAlancing MOdel for RPL

Tarcisio Bruno Oliveira[1], Pedro Henrique Gomes[2]
Danielo G. Gomes[1], Bhaskar Krishnamachari[2]

[1]Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat)
Departamento de Engenharia de Teleinformática – Universidade Federal do Ceará (UFC)
Fortaleza – CE – Brazil

[2]Autonomous Networks Research Group (ANRG)
Ming Hsieh Department of Electrical Engineering – University of Southern California (USC)
Los Angeles – CA – USA

{tarcisiocarneiro,dgomes}@great.ufc.br

{pdasilva,bkrishna}@usc.edu

Abstract. *Most of the nodes that compose the Internet of Things (IoT) are battery-operated, which makes energy efficiency a critical goal. However, more important than simply saving energy is consuming energy uniformly among the nodes. Imbalanced energy consumption may disrupt the network (energy hole problem). We present a solution that solves the network load balance problem; it is based on the RPL protocol, making it suitable for dynamic environments and compliant with standard network stack for IoT devices. The proposal was implemented in ContikiOS and tested in a real 41-node testbed using TelosB motes. Network lifetime had a 2-fold increase compared to the default RPL implementation. We also reduced the standard deviation of consumption by 50.64%, which indicates that nodes spend energy homogeneously, thus extending the lifetime of most of the nodes.*

1. Introduction

The Low-Power and Lossy Networks (LLN) are comprised of several sensing embedded systems that communicate and enable different applications, such as smart building, habitat monitoring [Mainwaring et al. 2002] and industrial automation. These networks are part of the Internet of Things (IoT), which will be the biggest driver for productivity growth [Accenture 2014]. The Internet of Things (IoT) is expected to have tens of billions of devices by the end of the decade.

In most of the LLN-based applications, the sensor nodes have constrained resources: memory, processing power and energy. Considering these limited capabilities, the routing protocols need to be well designed, since protocols impact the use of node's resources and energy consumption. In addition to providing reliable and efficient delivery, the routing algorithms need to take care of energy efficiency in order to extend the network lifetime.

In 2012, the IETF ROLL working group published the RFC 6550, which specifies a Routing Protocol for Low-Power and Lossy Networks (RPL). The RPL became the standardized IPv6-compliant routing protocol for LLNs. It is a distance vector protocol that establishes Destination Oriented Directed Acyclic Graphs (DODAGs) based on links and/or nodes metrics [Winter et al. 2012]. A DODAG is a tree-like topology that supports both downstream and upstream traffic and is built according to an Objective Function (OF).

Different Objective Functions can be designed in order to achieve specific optimization criteria and satisfy the requirements of a particular application. Supplementing RPL specification, the ROLL working group also standardized the Minimum Rank with Hysteresis Objective Function (MHROF). MHROF is designed to be extended with different metrics to calculate the cost of a path. A default metric defined by RFC 6719, which is used in most of RPL implementations, uses the Expected Transmission Count (ETX) of links to calculate the path with minimum cost. With DODAGs created using MHROF with the ETX metric, nodes tend to select parents with best link quality and smallest hop-count to the root node.

Even though selecting the parent with best link quality is a local optimum choice, it may cause a load-balancing problem. In scenarios where nodes distribution and/or traffic pattern are heterogeneous, the workload imbalance becomes critical. Nodes that are closer to the root and/or that have many children with good-quality links will receive and forward a large number of packets which may quickly deplete their battery life.

To tackle the above issues, we propose a new Objective Function based on MHROF. The new OF uses both the traffic profile of the nodes and the ETX of the links in order to solve the imbalance problem. The proposal is called **ALABAMO** (**A** **L**o**A**d **BA**lancing **MO**del for RPL) and it is a flexible model that uses two input parameters (workload characterization) and improves the task of choosing the preferred parent in RPL aiming at the extension of the network lifetime.

In this context, the main contribution of this paper is a new Objective Function (implemented in ALABAMO) that is compatible with RPL and provides traffic-aware balanced routing. The main target application of ALABAMO is multi-hop data collection applications, such as envrionmental monitoring. We implemented ALABAMO in a real operating system and evaluated it in a testbed with 41 nodes. Afterwards, we compared our solution with the default MHROF.

The paper is organized as follows. Section 2 provides the background of RPL and its issues. The ALABAMO model and its algorithm is described in Section 3. Following, Section 4 evaluates the performance of our proposal. We present prior work in Section 5. Finally, Section 6 concludes this paper.

2. Background and Problem Statement

Routing protocols for LLNs have to face challenging problems such as unreliable wireless links, lack of infrastructure and constrained resources. RPL was developed to work efficiently in networks comprised of thousand of nodes with limited resources and high packet loss. It is an adaptable routing protocol that dissociates packet processing and forwarding from the routing optimization. Decoupled from its routing core, the Objective Function (OF) is the way of achieving the optimization criteria required by the applications, such as minimum energy consumption or minimum end-to-end latency. An OF describes how nodes should convert one or more metrics and/or constraints into a rank. A rank represents the node's distance to the Directed Acyclic Graph (DAG) root. In addition, an OF also is used to assist nodes in choosing the best parent to be used in the upwards direction to reach the root.

2.1. RPL Overview

RPL builds a DAG that is composed of one or more Destination Oriented DAGs (DODAGs), each one rooted at a border router or sink node. RPL supports all three types of traffic: multipoint-to-point, point-to-multipoint and point-to-point traffic [Clausen et al. 2014]. Four types of custom ICMPv6 control messages are used to build and maintain the routing topology [Conta et al. 2006]:

- DIO (DODAG Information Object) : Message that carry information about the DODAG, such as RPLInstanceID, Version Number, Rank and the routing metrics used to compute the routes. The messages are transmitted through the network to construct and maintain the DODAG. Only the root node can start their dissemination. Every node that joined a DODAG also broadcasts DIO periodically;
- DIS (DODAG Information Solicitation): A node can send a DIS message to one (unicast) or multiple (multicast) neighbors, to proactively solicit configuration information;
- DAO (Destination Advertisement Object): These messages are used when downward traffic is needed;
- DAO-ACK (Destination Advertisement Object Acknowledgement): This control message reveals if the neighbor that received the DAO intends to be a previous hop for the sender in the downward route.

2.1.1. Topology Formation

The DODAG construction starts when the root node broadcasts a DIO message that indicates the initial configuration parameters. The root's neighbors take a rank based on the objective function and choose their preferred parent. The node's ranks are at least *MinHopRankIncrease* larger than the root's rank. Following, the root's neighbors start to broadcast their own DIO messages. As the process converges, each node in the network receives DIO messages and has a preferred parent set.

RPL implements the Trickle Algorithm [Levis et al. 2011] to reduce the rate at which the DIO messages are sent after the topology has reached an stable point. This mechanism prevents the nodes from wasting energy, since it continually doubles the transmission period until it reaches a maximum value. However, when some inconsistency is detected, e.g. a loop, link disruption or parent change, Trickle resets the timeout to a minimum value in order to promptly propagate the updated DODAG.

2.1.2. Objective Function (OF)

RPL's DODAG construction process is mainly determined by the Objective Function (OF) and the path metrics. An OF defines how a node computes its rank, which is based on one or more metrics. A rank represents the relative distance to the DODAG root. Moreover, it also describes how a node must choose its preferred parent. Thus far, the IETF ROLL working group detailed two OFs, which are implemented in most of the operating systems developed for LLNs, such as TinyOS and ContikiOS:

- Objective Function Zero (OF0): This OF uses hop-count as a routing metric. A node calculates its rank by adding a positive and indirectly normalized scalar value to its preferred parent rank;
- Minimum Rank With Hysteresis Objective Function (MHROF): This OF chooses routes that minimize additive routing metrics such as latency, hop-count and ETX [Vasseur et al. 2011]. In addition, MHROF uses hysteresis to reduce instability due to small metric changes. ContikiOS uses MHROF with Expected Transmission Count (ETX) routing metric as default objective function.

2.2. RPL Problem

Both objective functions proposed by the IETF ROLL group aim to reduce the number of packet retransmissions. While OF0 tries to minimize the number of hop-count, MHROF focus on the pairwise communication quality, based on the ETX metric.

In a network using RPL with MHROF based on ETX metric, nodes will choose the parents with best link quality. Considering a network with non-uniform distribution and uneven data traffic, it may result in significant load imbalance. The sensor nodes that have links with better quality will forward more packet than the others. Thereby, their energy depletion will be notably faster than the nodes with light traffic, and thus, gaps and holes in the network will appear, reducing the network lifetime.

3. ALABAMO: A Load Balancing Model for RPL

To solve the imbalance problem of RPL, we propose a load-balancing model called ALABAMO. Our model takes into account the traffic profile to avoid overloading the nodes with high-quality links. The implementation of ALABAMO is an objective function based on the Minimum Rank with Hysteresis Objective Function (MHROF). In our solution, every node broadcast the number of transmitted packets during the last measurement interval, which includes the generated and forwarded packets; such information is embedded into DIO messages. To choose the preferred parent, the nodes consider standard path quality metrics, such as ETX, and the packets number that each of its candidate parents has already sent.

ALABAMO uses a hysteresis mechanism similar to the one employed in MHROF to prevent unstable changes during fast fluctuations. We define two auxiliary constants to provide flexibility when weighting the parameters in our objective function.

The first constant is $MaxETX_{Ratio}$, which adjusts how parameter ETX will be used in the path metric. The ETX metric is still an important metric in our optimization, but we impose a range of ETX values in which all parents can are equally eligible to become the preferred one. During the optimization process, if the ratio of the ETX path metric of two parents is greater than $MaxETX_{Ratio}$, a node must select the parent with fewer sent packets. Considering node n and p_1 and p_2 as candidates parents of n, if ETX_{p_1} is the ETX metric of p_1 and ETX_{p_2} is the ETX metric of p_2, then the ETX metric ratio is calculated as follows:

$$ETX_{Ratio} = \begin{cases} \frac{ETX_{p_2}}{ETX_{p_1}} * 100 & \text{if } ETX_{p_1} \geq ETX_{p_2} \\ \\ \frac{ETX_{p_1}}{ETX_{p_2}} * 100 & otherwise \end{cases} \quad (1)$$

The second constant is $MaxWorkload_{Ratio}$, and it adjusts how the difference between the number of transmitted packets by different parents will affect the optimization function. We calculate the $Workload_{ratio}$ similarly to ETX_{ratio}, except that an initial offset is inserted to prevent instability at the beginning of the network execution. A common $offset$ value used in our experiments is 100. Considering $SentPkt_{p_n}$ as the number of packets transmitted by parent p_n, we have the following:

$$Workload_{Ratio} = \begin{cases} (\frac{SentPkt_{p_2}+offset}{SentPkt_{p_1}+offset}) * 100, & \text{if } SentPkt_{p_1} \geq SentPkt_{p_2} \\ \\ (\frac{SentPkt_{p_1}+offset}{SentPkt_{p_2}+offset}) * 100, & otherwise \end{cases} \quad (2)$$

The main piece of code implemented in ALABAMO is the function responsible for choosing the preferred parent, whose pseudocode is shown in **Algorithm 1**. **Algorithm 1** is executed for each pair of neighbors, in order to rank all possible parents.

Algorithm 1 is executed to analyze the parents pairwise. It receives as argument both parents and the pre-calculated workload ratio and the ETX ratio. Normally, the RPL would choose the parent with small path metric (only accounting for ETX); However, ALABAMO's mechanism checks the proportion of transmitted packets and the ETX metric. If the ETX metric of some parents are within an acceptable range and the difference of sent packets is substantial (larger than a certain threshold), the node chooses the parent with smallest number of transmitted packets.

As a result, the nodes in the network select the parent with a tolerable link quality and with less packets transmitted, increasing the load balancing. In our proposal, there is a trade-off between load balancing and packet loss. By tuning the constants $MaxETX_{Ratio}$ and $MaxWorkload_{Ratio}$ we can prioritize each of these two factors. If we reduce $MaxETX_{Ratio}$ the network will be more balanced, but packet loss will increase because more nodes with low-quality links will be selected as preferred parents. Besides, reducing the $MaxETX_{Ratio}$ increases the number of parent switches, which results in a larger number of packets transmitted to update the DODAG, consuming more energy.

$MaxWorkload_{Ratio}$ plays a secondary role and is important to prevent churning and unnecessary network overloading. If there is more than one candidate parent that could be equally selected according to the ETX metric, we select the one with lowest workload, as long as the cost of switching is not high. If we increase $MaxWorkload_{Ratio}$ we have more freedom to switch. In other words, we will select a better parent even if it is not so much better than the current one. If $MaxWorkload_{Ratio}$ is reduced it becomes harder to switch between parents, but it may be beneficial if the network is very dynamic and thus making too many switches, incurring a large cost.

Algorithm 1 Best parent selection algorithm

Require: $p_1, p_2, Workload_{ratio}, ETX_{ratio}$
Ensure: $best\_parent$

$min_{diff} \leftarrow rpl\_dag\_mc\_etx\_divisor \div parent\_switch\_threshold\_div$
$m_{p_1} \leftarrow calculate\_path\_metric(p_1)$
$m_{p_2} \leftarrow calculate\_path\_metric(p_2)$
if $(p_1\ OR\ p_2\ are\ current\_preferred\_parent)$ **then**
    if $(m_{p_1} < m_{p_2} + min\_diff)\ AND\ (m_{p_1} > m_{p_2} - min\_diff)$ **then**
        if $(SentPkt_{p_1} > SentPkt_{p_2})\ AND\ (Workload_{ratio} < MaxWorkload_{Ratio})$ **then**
            $return\ p_2$
        end if
    else
        if $(SentPkt_{p_2} > SentPkt_{p_1})\ AND\ (Workload_{ratio} < MaxWorkload_{Ratio})$ **then**
            $return\ p_1$
        end if
        $return\ current\_preferred\_parent$
    end if
end if
if $m_{p_1} < m_{p_2}$ **then**
    if $(Workload_{ratio} < MaxWorkload_{Ratio})\ AND\ (ETX_{ratio} > MaxETX_{Ratio})$ **then**
        $return\ p_2$
    end if
    $return\ p_1$
else
    if $(Workload_{ratio} < MaxWorkload_{Ratio})\ AND\ (ETX_{ratio} > MaxETX_{Ratio})$ **then**
        $return\ p_1$
    end if
    $return\ p_2$
end if

Therefore, the optimal value of the ETX_{ratio} and $MaxWorkload_{Ratio}$ should be set according to the network scenario. Some factors that have influence in these parameters choice are: link quality variance, packet transmission rate and network density. In our experiments we fixed $MaxWorkload_{Ratio}$ to 70 and varied ETX_{ratio} between 50 and 90 to evaluate its impact.

At line 1 of **Algorithm 1**, a hysteresis constant is set based on two other parameters. These parameters are related to how the ETX metric is calculated and its range. In our experiments the value was 128. The ETX metric of p_1 and p_2 are assigned at lines 2 and 3. Lines 4 through 15 are responsible for the hysteresis mechanism. If the difference of the two calculated metric is less than the min_{diff}, the function returns the current preferred parent and the network does not change. However, if the current preferred parent has sent more packets than the other one, the function returns the parent with less transmitted packets. It represents the load balancing mechanism. This can be seen at lines 6, 7 and 9-11.

At line 16, the algorithm checks if the metric of p_1 is smaller than p_2. Line 17 implements the balanced mechanism. Even the metric of p_1 is smaller than metric of p_2, the algorithm returns p_2 if the $Workload_{Ratio}$ is smaller than $MaxWorkload_{Ratio}$ and ETX_{ratio} is greater than $MaxETX_{Ratio}$. That means that the ETX metric of p_1 and p_2 are close and the number of transmitted packets by each is significantly different. Lines 21 through 26 work similarly to lines 17 through 20 but in accordance with the other case; when the metric of p_2 is smaller than the metric of p_1.

4. Performance Evaluation

We evaluated the ALABAMO performance and compare it to the default MHROF [Gnawali and Levis 2012], which is implemented in ContikiRPL [Tsiftes et al. 2010]. Both ALABAMO and MHROF were used in a real network with nodes running ContikiOS.

In a wireless sensor network, the nodes that are first to deplete their battery are the ones closer to the root, assuming that all nodes have the same transmitting power and battery size, and that the network traffic is uniform. Hence, it is important that all sub-trees have an evenly distributed number of nodes.

4.1. Evaluation scenario and performance metrics

The scenario used for the real experiments consisted of 41 nodes in the Tutornet testbed; one sink and 40 sensor nodes. The motes are deployed in an area of 55 x 30 meters, and placed above the ceiling. The testbed is located in a working-environment with 7 laboratories and more than 10 working offices, all of them separated by dry-walls. There are 8 Wi-Fi access points spread on the floor, using the whole 2.4 GHz spectrum, and various other sources of interference, such as Bluetooth devices and micro-waves. Moreover, it can be estimated that more than 60 people work or pass by the considered floor daily, which creates a real dynamic and challenging environment for wireless networking. The location of the nodes can be found at the Testbed's website at *http://anrg.usc.edu/www/tutornet/*. In the experiments the sink node was *Node 1* and the 40 sensor nodes were *Nodes 2 to 41*, in the topology shown on the mentioned website. All nodes are TelosB motes running ContikiOS 3.0. ALABAMO was implemented using the interfaces provided by ContikiOS [1].

Each sensor node transmits one UDP packet towards the sink every 30 seconds. Each experiment lasts for 2 hours, summing up to $9,600$ data packets per experiment. In addition to data packets there are also regular ICMPv6 packets used by RPL protocol. In all experiments DIO and DAO maximum timeout was set to $1,048$ seconds. The utilized Medium Access Control (MAC) layer has a maximum of 3 link-layer retrials.

In the experiments, we disregarded issues with nodes synchronization in the MAC layer and assumed that nodes are capable of sleeping in order to save energy. Hence, energy consumption is restricted to packets sent, received and retransmitted. We found out throughout our experiments that X-MAC implementation in ContikiOS does not guarantee highly precise synchronization using TelosB motes, so we decided to employ NullMAC, with no sleep functionality, and calculate the total energy consumption using the packets statistics. It is important to emphasize that this synchronization problem has already been solved by standard IEEE 802.15.4e and its protocols DSME and TSCH. Operating systems such as OpenWSN [2] that are focused on critical applications provide MAC layers that guarantee a minimum waste of energy due to desynchronization and overhearing.

The experiments considered the default **MHROF** as found in ContikiOS 3.0 implementation. ALABAMO proposal with $MaxWorkload_{Ratio}$ equals to 70 when the constant $MaxETX_{Ratio}$ is set to either 80 or 90. Hence, we used names **ALABAMO-80** and **ALABAMO-90** for differentiating the proposal with two different constants.

Each experiment was repeated 12 times (each lasting 2 hours) in order to calculate standard deviation and confidence intervals.

4.2. Routing tree

We first analyzed the impact of ALABAMO on the routing tree. Since it tries to balance out the number of packets forwarded by intermediate nodes, it tends to result in sub-trees with approximately the same number of nodes and a shallow routing tree (with smaller maximum hop count), since nodes have more flexibility to use links with lower ETX to reach the sink.

In the experiments, a snapshot of the routing tree was taken every 30 minutes. Table 1 shows the average size of all sub-trees and the average size of the heaviest sub-tree. The heaviest sub-tree is the one with the most nodes connected to it. It is of special interest because it is usually the first one to disrupt due to battery depletion of the head node.

Model	Sub-trees size	Std deviation	Heaviest sub-tree size	Std deviation
MHROF	4.00	4.93	18.00	0.12
ALABAMO-80	3.63	3.79	11.93	1.67
ALABAMO-90	3.64	3.34	10.33	1.23

Table 1. Average size of sub-trees and average size of heaviest sub-tree.

[1]Code can be found at http://anrg.usc.edu/downloads

[2]http://www.openwsn.org

The experiments showed that MHROF has the most loaded sub-trees, with 4 nodes on average. Both of ALABAMO solutions have a similar average of sub-trees with 3.6 nodes, which is less than the MHROF. An important aspect to be considered is the standard deviation; both ALABAMO solutions decreased the standard deviation of sub-tree sizes (when compared to MHROF), which means that the distribution of nodes in the sub-trees is more uniform, favoring a better network load balancing.

Concerning the heaviest sub-trees, we clearly notice the critical imbalance problem in MHROF, which has sub-trees with 18 nodes on average, corresponding to almost half of the nodes in the network. ALABAMO reduced the heaviest sub-tree size by at least 6 nodes in comparison to MHROF.

We also calculated the maximum hop-count of the network, which is the number of hops from the furthermost node from the sink. MHROF and ALABAMO-90 had both the same maximum hop count of 4 in all experiments. In ALABAMO-80, the maximum hop count fluctuated between 3 and 4, with an average of 3.5 and standard deviation of 0.52. Lower maximum hop count results in less packets being forwarded, which may increase the lifetime of the network and the overall delivery ratio. The solution ALABAMO-80 showed results better than MHROF's, which highlights the benefits of the proposed scheme.

4.3. Network Delivery Ratio

The delivery ratio of a network is the percentage of packets sent by the sensor nodes that are actually received by the sink. ALABAMO presents a trade-off between ETX and the parent's workload; it picks links with lower quality for better balancing, but compromises the network delivery ratio.

All packets transmitted were UDP datagrams, hence no transport-layer retransmission was employed. This way, we intend to verify the impact of using links with lower quality in the overall performance; transport-layer retransmissions would influence the delivery ratio and make it more difficult to extract such statistics. The link layer used acknowledgement packets and a maximum of 3 retransmissions. It is important to notice that, because of the light traffic, there is no loss due to queue dropping. In fact, during the experiments, we monitored the number of packets dropped and confirmed that only in a few round there was at most 1 packet dropped by some of the nodes.

Figure 1 shows the average network delivery ratio. As expected, the ALABAMO's delivery ratio is lower than MHROF's since the nodes can select parent with lower quality links. The difference between MHROF and ALABAMO-80 is approximately 22%, which is acceptable in certain applications and does not incur very large overhead if transport-layer retransmissions are employed. Since ALABAMO-90 is expected to choose links with higher ETX than ALABAMO-80, the delivery ratio of the former was supposed to be higher. However, ALABAMO-90 resulted in trees with more hops, which increases the number of packets being forwarded, and consequently, the chances of transmission failures is also increased. Indeed, results show that ALABAMO-90's delivery ratio is lower than ALABAMO-80's. This fact stresses that using links with lower ETX may result in improvements in various aspects of the network, with regard to reducing energy consumption and even creating a more efficient routing tree. The obtained delivery ratio on all three scenarios is sufficient for most stateless Restful applications based on protocols

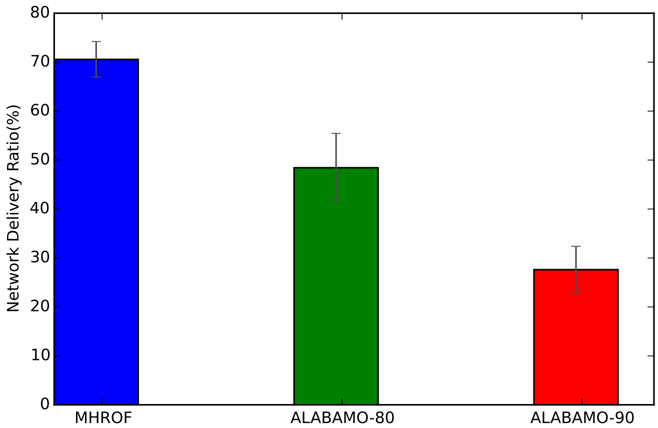

Figure 1. Network delivery ratio (percentage of data packets received by the sink).

such as CoAP, where retransmissions of missed requests are expected to happen.

4.4. Energy consumption and network lifetime

The energy consumption of a node can be split into processing (microcontroller and sensor) and communication. In general the communication consumption is the highest burden for achieving a longer lifetime. Energy is consumed by the radio chip when it is transmitting, receiving and hearing the medium. We disregard the waste due to overhearing. Hence, our calculations consider energy consumed due to reception of messages to be forwarded (since all messages are destined towards the sink) and transmission of generated and forwarded messages.

We measured the power consumed during two hours of experiment. The lower the standard deviation, the more balanced is the network. In addition, the more balanced is the network, the longer is its lifetime. Table 2 shows the average power consumption of all nodes and the average power consumption of the most loaded node (the head of the heaviest sub-tree).

Figure 2 shows that expected lifetime of the network. For this estimation, we assume that the most loaded node is the first node to deplete its battery, causing the network disruption. The bar graphs shows the lifetime normalized by the shortest one (MHROF).

Model	Average consumption (per node in mW)	Std. deviation	Average consumption (most loaded node in mW)	Std. deviation
MHROF	4.65	6.24	34.72	9.04
ALABAMO-80	4.07	3.08	16.69	3.70
ALABAMO-90	4.04	4.79	21.21	15.12

Table 2. Average power consumption of all nodes and of the most loaded node.

We noticed that all models have similar average consumption, which is expected, since the same number of packets are generated in the experiments. However, we highlight the difference of the standard deviation values. ALABAMO-80 has the lowest value. MHROF builds an unbalanced network and its standard deviation value is the highest. In

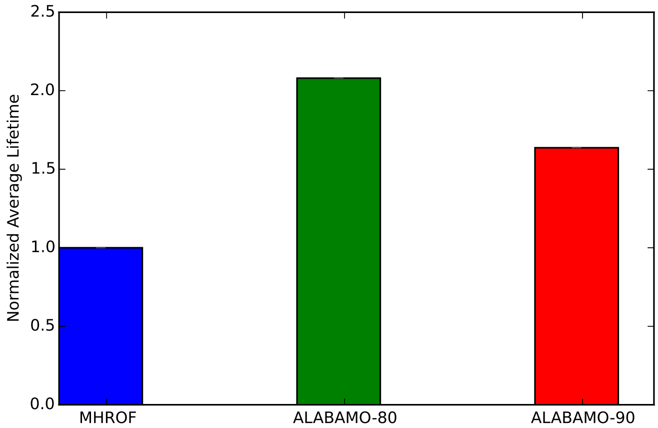

Figure 2. Normalized average network lifetime.

regards to the network lifetime, we can verify that ALABAM0-80 has a two-fold increase related to MHROF, while ALABAMO-90 has a network lifetime slightly lower than the ALABAMO-80.

4.5. Parent switching

The flexibility of dynamically choosing parents with lower workload comes with a cost. Every time the preferred parent of a node is changed, the routing tree is disturbed and nodes need to learn the new preferred paths.

MHROF has the smallest average number of parent switches per node, with an average of 0.30 and small standard deviation of 0.69. On the other hand, ALABAMO-90 almost doubled the number of parent switches (average of 0.59), and ALABAMO-80 increased in more than 4 times this number (average of 1.24). The standard deviation of the statistics of ALABAMO solutions was also much higher, more than 1.2. It is clear that even though MHROF cannot form balanced trees, it generates trees that do not change over time, which can be beneficial for critical applications that cannot afford network churn.

5. Related Work

Load-balancing in LLNs has been widely investigated over the last few years. Some hierarchical solutions achieve a balanced network, which organize the nodes into clusters and selecting the best arrangement and cluster heads [Kumar et al. 2011]. On the other hand, flat-based routing solutions are more suitable for non-uniform deployment and heterogeneous traffic pattern. Workload balancing with flat routing algorithms is achieved through the construction of balanced routing trees. Such trees may be constructed either centrally [Durmaz Incel et al. 2012] or in a distributed fashion [Bhatti et al. 2013, Kim 2015]. The former type requires global knowledge of the network and is less flexible for mobile or heterogeneous networks, but usually obtains best results; the latter is based on local information and is more adaptable to network changes but, in general, presents non-optimal results.

Bhatti et al. (2013) proposed a solution where nodes with heavy workload signal their status by delaying the transmission of DIO messages. Based on the past DIOs time-stamps, each node can dynamically decide how to spread out data among the candidate parents. Although the implicit signaling does not require any information to be transmitted, it might be easily affected by fading links that can increase packet loss and then be interpreted as an overloading signal. The proposal was only evaluated on the NS2 simulator.

Gaddour et al. (2014) proposed a holistic OF that combines four metrics: hop count, link quality, node energy and end-to-end delay, using fuzzy parameters. Although this type of composite solution is flexible for different scenarios, it is not clear how applications would signal to the underlying routing layer which metric should be optimized. Beside this issue, RPL protocol defines that there may exist multiple DODAGs in a network, each one with a different OF that targets a specific application. Hence, composite metrics that consider too many metrics may incur greater complexity than the use of multiple simpler OFs. The solution in [Gaddour et al. 2014] was only evaluated with the Cooja simulator [Dunkels et al. 2004].

Some recent works [Capone et al. 2014, Iova et al. 2014, Kim 2015] have designed OFs that optimize the network lifetime considering two different metrics, one aimed at choosing links with good quality and the other for saving energy. [Capone et al. 2014] presented two different ways of estimating the residual energy of nodes and use this model to calculate a lifetime cost metric. The residual energy concept was also used by Iova et al. (2014), but rather than equally saving energy among all nodes, their proposed metric focused on energy consumption of the bottleneck nodes. It also considered parameters such as link data rate, different power transmission, and link qualities. Both solutions [Capone et al. 2014, Iova et al. 2014] present good results but were exclusively evaluated using simulation models. Even these are distributed solutions, they rely on models and equations that require parameters that may be different for each node, such as energy source (battery size) and transmission power.

ALABAMO works primarily based on hardware-independent information, such as the number of transmitted and forwarded packets. It is important to highlight that, although many proposals use residual energy as metric, it is not well stated how they can be implemented in a distributed scenario using RPL as routing protocol. Plus, our evaluation process is based on a real network in a dynamic environment.

A proposal that is closest to ALABAMO is that of Kim (2015). His approach, named QU-RPL, shows that under heavy traffic LLNs suffer from a serious load balancing problem that cause network lifetime to be drastically reduced. QU-RPL, like ALABAMO, takes into account the number of packets being forwarded by the candidate parents. However, while QU-RPL optimizes the packet delivery ratio and is best employed in cases where queue utilization is high, ALABAMO aims at extending the network lifetime and may be used in both heavy and light traffic conditions.

6. Conclusion

Here we present a model, called ALABAMO, that implements a new Objective Function for RPL. The proposal aims at balancing the traffic among forwarder nodes and, hence, extending the network lifetime. Results show that ALABAMO can reduce in half the energy consumption of the most loaded node, which represents a 2-fold increase in network lifetime (considering that the network is disrupted when the first node dies). Besides, the standard deviation of energy consumption of all nodes was also reduced in half, which means that all nodes have their battery lasting a similar amount of time, making the network last longer as a whole.

Therefore, we conclude that ALABAMO builds a better-balanced tree than the MHROF and improves the network lifetime, even though it reduces the network delivery ratio a little. Lastly, ALABAMO is a practical solution to imbalance problem and easy to implement.

Future work will compare ALABAMO with other load balanced algorithms that uses centralized and distributed techniques. Moreover, we will analyze the impact of ALABAMO in other metrics such as delay.

References

Accenture (2014). Driving Unconventional Growth through the Industrial Internet of Things.

Bhatti, G., Orlik, P., and Parsons, K. (2013). Load balanced routing for low power and lossy networks. *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2238–2243.

Capone, S., Brama, R., Accettura, N., Striccoli, D., and Boggia, G. (2014). An Energy Efficient and Reliable Composite Metric for RPL Organized Networks. *2014 12th IEEE International Conference on Embedded and Ubiquitous Computing*, pages 178–184.

Clausen, T., Verdiere, A., Yi, J., Herberg, U., and Igarashi, Y. (2014). Observations of rpl: Ipv6 routing protocol for low power and lossy networks. Internet-Draft draft-clausen-lln-rpl-experiences-09, IETF Secretariat. `http://www.ietf.org/internet-drafts/draft-clausen-lln-rpl-experiences-09.txt`.

Conta, A., Deering, S., and Gupta, M. (2006). Internet control message protocol (icmpv6) for the internet protocol version 6 (ipv6) specification. RFC 4443, RFC Editor. `http://www.rfc-editor.org/rfc/rfc4443.txt`.

Dunkels, A., Grönvall, B., and Voigt, T. (2004). 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.

Durmaz Incel, O., Ghosh, A., Krishnamachari, B., and Chintalapudi, K. (2012). Fast Data Collection in Tree-Based Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, 11(1):86–99.

Gaddour, O., Koubaa, A., Baccour, N., and Abid, M. (2014). OF-FL: QoS-aware fuzzy logic objective function for the RPL routing protocol. *2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 365–372.

Gnawali, O. and Levis, P. (2012). The minimum rank with hysteresis objective function. RFC 6719, RFC Editor. `http://www.rfc-editor.org/rfc/rfc6719.txt`.

Iova, O., Theoleyre, F., and Noel, T. (2014). Improving the network lifetime with energy-balancing routing: Application to RPL. *2014 7th IFIP Wireless and Mobile Networking Conference (WMNC)*, pages 1–8.

Kim, H.-S. (2015). QU-RPL : Queue Utilization based RPL for Load Balancing in Large Scale Industrial Applications. In *12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON 2015)*, pages 265–273, Seattle.

Kumar, V., Jain, S., Tiwari, S., et al. (2011). Energy efficient clustering algorithms in wireless sensor networks: A survey. *IJCSI International Journal of Computer Science Issues*, 8(5).

Levis, P., Clausen, T., Hui, J., Gnawali, O., and Ko, J. (2011). The trickle algorithm. RFC 6206, RFC Editor. `http://www.rfc-editor.org/rfc/rfc6206.txt`.

Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J. (2002). Wireless Sensor Networks for Habitat Monitoring. *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 88–97.

Tsiftes, N., Eriksson, J., and Dunkels, A. (2010). Low-power wireless IPv6 routing with ContikiRPL. *Proc. 9th ACM/IEEE International Conference on Information Processing in Sensor Networks - IPSN '10*, pages 406–407.

Vasseur, J., Kim, M., Pister, K., Dejean, N., and Barthel, D. (2011). Routing metrics used for path calculation in low power and lossy networks. Internet-Draft draft-ietf-roll-routing-metrics-19, IETF Secretariat. `http://www.ietf.org/internet-drafts/draft-ietf-roll-routing-metrics-19.txt`.

Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, J., and Alexander, R. (2012). Rpl: Ipv6 routing protocol for low-power and lossy networks. RFC 6550, RFC Editor. `http://www.rfc-editor.org/rfc/rfc6550.txt`.