Exploiting Multiple Parents in RPL to Improve both the Network Lifetime and its Stability

Oana Iova, Fabrice Theoleyre, Thomas Noel

Exploiting Multiple Parents in RPL to Improve both the Network Lifetime and its Stability

Oana Iova, Fabrice Theoleyre, Thomas Noel
CNRS, ICube, University of Strasbourg, France
Email: {otiova, theoleyre, noel}@unistra.fr

***Abstract*—The devices composing Wireless Sensor Networks (WSN) are very limited in terms of memory, processing power and battery. RPL has emerged as the *de facto* routing standard in low-power and lossy networks. While most of the proposals focus on minimizing the global energy consumption, we aim here at designing an energy-balancing routing protocol: each node should efficiently consume the same quantity of energy to improve the network lifetime. To this end, we exploit an *Expected Lifetime* metric, denoting the *residual time* of the nodes (time until the node will run out of energy). We propose mechanisms to detect the energy-bottleneck nodes and to spread the traffic load uniformly among them. While RPL constructs a Destination-Oriented Directed Acyclic Graph (DODAG) structure, it only implements single path. We propose here to exploit its natural multipath structure. This multipath approach helps reducing the number of DODAG reconstructions that leads to instabilities and convergence problems. Simulations highlight we improve both the routing reliability and the network lifetime.**

***Index Terms*—RPL; multipath; energy-balancing; network lifetime; WSN; stability**

I. Introduction

Routing in Wireless Sensor Networks (WSNs) has been extensively studied in the last decade. In this type of environment, a *good* routing protocol should:

a) save energy, since most of the nodes are battery powered;
b) deal with lossy links for both control and data packets;
c) avoid routing loops and enable fast convergence.

RPL has emerged as the *de facto* routing standard for WSNs [1]. It aims at optimizing the routing scheme for the convergecast traffic pattern, i.e., all the packets are sent to a collection of *border routers*, connected to the Internet. RPL is based on a Destination-Oriented Directed Acyclic Graph (DODAG) rooted at the border routers, which reflects the current evolution of this research area: introducing redundancy in the routing structure to be fault-tolerant.

However, in our opinion, the current version of RPL presents two ways of improvement. First, the IETF Roll working group has focused on standardizing an efficient routing protocol. We have now to provide metrics and mechanisms to make RPL energy-efficient: the topology (i.e., the DODAG) should be constructed based on energy criteria. Second, a node selects one preferred parent to construct the DODAG without loops. However, only this preferred parent is used for routing: the other ones have just a *backup* purpose. Consequently, the data plane only uses a **tree** structure. We are convinced we should exploit this built-in diversity to distribute the traffic load in the network and to create energy-balanced paths.

This work was partially supported by the French National Research Agency (ANR) project IRIS under contract ANR-11-INFR-016.

Two main approaches exist in the literature to save energy. The first one minimizes the global energy consumption. The ETX routing metric for instance, can be used to select energy-efficient links [2]. However, the nodes offering the best links will be chosen uppermost to route the packets, and this will deplete their energy faster. The second one selects in priority nodes with a large residual energy [3]. Still, these nodes, with possibly bad links, will receive most of the traffic and will run out of energy faster. We propose here a third approach: balancing efficiently the energy among all the nodes.

In this paper, we use existing concepts like multipath communication [4], load balancing, and energy-aware path selection, to present a complete and pragmatic solution. We map our proposal to RPL and we present a methodology for its implementation. The contribution of this paper is fourfold:

1) we identify the energy bottleneck nodes and construct accordingly the DODAG based on an *Expected Lifetime* metric, which denotes the time until the node will run out of energy;
2) using one single parent would require to estimate actively the link quality for all other possible parents (no unicast traffic is transmitted to them). We propose rather to exploit the multipath feature of our approach to estimate passively the link quality toward **all the parents**.
3) we address the stability problem [5]: multipath helps avoiding sudden routing reconfigurations: a node changes its preferred parent only when it becomes useless (i.e., it does not forward traffic anymore), and sends less control traffic (i.e. energy);
4) we propose an algorithm to split the traffic among several paths, proportionally with the lifetime of their corresponding bottlenecks.

II. Related Work

A. RPL: Routing Protocol for Low-power and Lossy Networks

RPL is a distance vector protocol for low-power and lossy networks [1]. Starting from a border router, RPL constructs a Destination-Oriented Directed Acyclic Graph (DODAG).

The DODAG construction is based on the *Rank* of a node, which depicts its relative distance to the DODAG root. An *Objective Function* defines how the routing metrics have to be combined to compute the *Rank*. Information about the *Rank*, the *Objective Function*, and other configuration parameters are included in the DODAG Information Object (`DIO`) messages. These messages are periodically broadcasted by all the nodes, in order to construct and maintain the DODAG.

When a node receives a `DIO`, it inserts the emitter in the list of possible successors (next hops to the border router), from which it chooses one preferred parent. It then computes its own *Rank* with the *Objective Function* and starts broadcasting `DIO` messages.

The DODAG structure creates and maintains multiple routes towards the sink. However, RPL only uses a single path to route the packets (through the preferred parent). Pavkovic *et al.* extended RPL to be used opportunistically with IEEE 802.15.4-2006: a node sends a data packet to the first available parent, instead of waiting for the preferred parent to be available [6]. However, their focus is on offering QoS for delay-sensitive packets, and not on improving the network lifetime.

Hong *et al.* proposed to choose the preferred parent using the hop count and then to select as forwarding node the parent offering the best link quality [7]. However, this may be equivalent with choosing the best parent among the worst available ones. Besides, a collection of nodes may still forward most of the traffic, and will run faster out of energy.

B. Multipath Routing

Multipath routing has been widely used in the literature to improve the fault-tolerance (reliability), to balance the load (congestion avoidance), or for QoS improvement [4]. *Braided* paths (i.e., partially overlapping) have been proven to reduce the maintenance cost while still balancing the load efficiently [8]. Chen *et al.* used multipath in WSNs, but in a centralized manner [9]. Ming-hao *et al.* proposed a reactive multipath approach, but do not consider link reliability [10].

C. Energy Aware Routing

The most attention for energy efficient routing has been given to the design of protocols that reduce the overhead [11], and not to the *balancing* of the energy consumption. Pantazis *et al.* argued that *Energy Balanced Networks* represent a key future direction.

For an efficient energy balancing, a routing metric should take into consideration the remaining energy of the nodes, the current traffic, and radio link conditions. We will present next the related papers that addressed one subpart of this problem.

1) Routing Metrics in the Literature: ETX is widely used to estimate the energy budget for using a link [2]. It computes the average number of transmissions before a packet is correctly acknowledged. However, ETX does not balance the load in the network, it only concentrates traffic on energy efficient routes. Similarly, PWave minimizes the cumulative path cost [12], but it does not avoid the most loaded nodes.

On the contrary, the Residual Energy Depletion Rate (REDR) [13] uses directly the depletion ratio of the residual energy. Kamgueu *et al.* [3] proposed similarly to use the residual energy to construct the RPL DODAG. However, both approaches neither consider the radio link quality, nor the load of each node.

TABLE I: Notation used in the article

Notation	Meaning
ELT(X)	Expected lifetime of X
$\boldsymbol{E_{res}(X)}$	Residual energy of X (in Joule)
$\boldsymbol{P_{TX}(X)}$	Radio power in transmission mode (in Watt or Joule/s)
ETX(A,B)	ETX of the link $A \rightarrow B$
$\boldsymbol{\alpha_P}$	Ratio of traffic sent to parent P
$\boldsymbol{r_{X,B}}$	Ratio of traffic forwarded by X to bottleneck B
$\boldsymbol{T_{tot}(X)}$	Throughput (bits/s) of X
$\boldsymbol{T_{gen}(X)}$	Traffic generated by X
Children(X)	Children set of node X
Parents(X)	Parents set of node X
Bottlenecks(X)	Bottlenecks set of node X
DATA_RATE	The rate at which the data is sent (bits/s); All nodes transmit at the same rate

In [14] the authors formulated the routing problem as a linear programming problem where the objective is to maximize the network lifetime. However, they only presented a centralized algorithm to route the packets, where the radio topology has to be known perfectly.

2) The Expected Lifetime (ELT): We will detail here more clearly the *Expected Lifetime metric* (ELT) used to detect the energy-bottleneck nodes [15]. Instead of minimizing the sum of energy, or considering only the residual energy, ELT aims at maximizing directly the lifetime of the most constrained node, denoted *bottleneck*.

ELT estimates the expected lifetime, i.e., the time before a node dies if it keeps on forwarding the same quantity of traffic. ELT helps quantifying the impact of a routing decision on the bottlenecks.

In this paper, we consider the network lifetime as the time before the first node runs out of energy, since this is the most frequent definition [3], [13]. Moreover, with the routing metric that we propose, a node chooses to send its traffic on the least energy-constraint path. Hence, if a node runs out of energy it will most surely disconnect the network, otherwise, its neighbors would not have chosen it as parent.

To compute its ELT, a node needs to know (using the notation from Table I):

- the traffic it has to forward (generated by itself, plus the traffic received from its children): $T_{tot}(N)$;
- the average number of retransmissions of the link from the node to its parent (to count also the number of retransmissions): ETX(N, P);
- the energy drained per transmitted bit: $\frac{P_{TX}}{\text{DATA\_RATE}}$;
- its residual energy: $E_{res}(N)$.

Finally, a node N estimates its ELT as following:

$$ELT(N) = \frac{E_{res}(N)}{\frac{T_{tot}(N) \times \text{ETX}(N,P)}{\text{DATA\_RATE}} \times P_{TX}(N)} \qquad (1)$$

III. Multipath for Network Lifetime Maximization

We propose here to take advantage of the DODAG structure and to balance the traffic of a node to all its parents, proportionally with the expected lifetime of the corresponding bottlenecks. Consequently, we manage to further maximize the network lifetime, and also:

1) the traffic is energy balanced and the bottlenecks are equally charged;
2) the quality of the links can be continuously estimated, since the traffic is sent to all the parents;
3) there is no need to switch the preferred parent that often and hence, the trickle timer will be less frequently reset, which will save energy.

First, we propose to generalize the ELT metric to the multipath routing. In particular, we have to take into account the fact that a node will send its data packets to several parents with different link qualities. Hence, the energy consumed by the node will depend on the amount of traffic sent to each parent. If α_P is the ratio of traffic sent to parent P, then the computation of ELT from Equation 1 becomes in the multipath scenario:

$$ELT(N) = \frac{E_{res}(N)}{\sum\limits_{\mathbf{P \in Parent(N)}} \frac{\alpha_\mathbf{P} \times T_{tot}(N) \times ETX(N,P)}{DATA\_RATE} \times P_{TX}(N)} \tag{2}$$

where $\sum\limits_{P \in Parent(N)} \alpha_P = 1$.

We will now explore how to implement this metric in a gradient routing scheme. To use ELT with RPL in a multipath scenario we have to address the following challenges:

1) **Metric computation and advertisement**: how a node computes the ELT of a bottleneck and how it piggybacks the required information in its `DIO`;
2) **Multipath construction:** how to create an energy balanced topology, while avoiding loops;
3) **Energy balancing:** how a node computes the traffic load to each parent such that the paths are energy balanced.

Each of these challenges are addressed into more detail in the following sections.

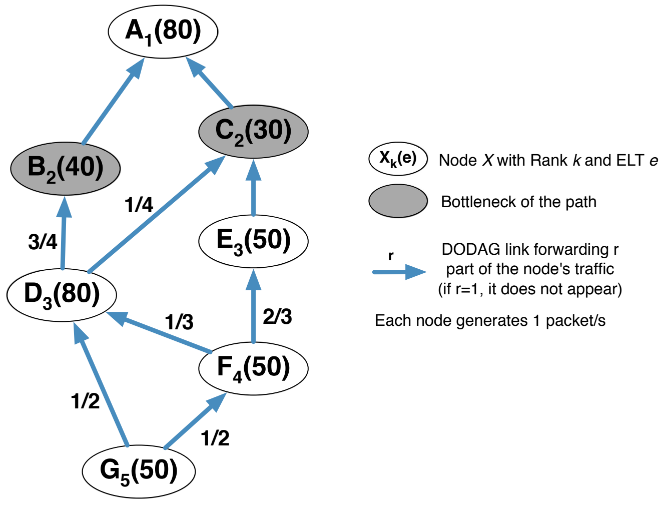
IV. Metric Computation and Advertisement

In the multipath scenario, a node will send its traffic through several paths, i.e., it will have to maintain information about several bottlenecks. Moreover, only a part of its traffic will arrive at a specific bottleneck. We present here how to compute the ELT of a bottleneck and how to send the information about the bottlenecks along the path in a compact manner.

A. ELT Estimation with Multiple Bottlenecks

Let us consider a node N, which has to associate with the DODAG. Since the bottleneck is most likely to be the first node to die, the new node has to estimate the impact of its own traffic on the lifetime of the bottleneck. But how does a node compute the ELT of a bottleneck?

First, in a multipath scenario, we have to take into account that a node sends its traffic to several parents. Hence, only a part of its total traffic will arrive at a specific bottleneck. We need to determine the proportion of traffic that a node forwards to a bottleneck.



A₁(80)
B₂(40)
C₂(30)
D₃(80)
E₃(50)
F₄(50)
G₅(50)
3/4
1/4
1/3
2/3
1/2
1/2
Xk(e) Node X with Rank k and ELT e
Bottleneck of the path
r DODAG link forwarding r part of the node's traffic (if r=1, it does not appear)
Each node generates 1 packet/s

Fig. 1: Path selection with ELT (`MinHopRankIncrease`=1)

Let us take an example. Consider node G and its parents D and F in Figure 1. $3/4$ of the traffic of D is forwarded through the bottleneck B. This is also the case of $1/3 \times 3/4$ of the packets of F. Finally, the actual quantity of traffic of G that reaches B is: $1/2 \times 3/4 + 1/2 \times 1/3 \times 3/4$. We can note that this ratio may be computed recursively.

Let $r_{N,B}$ be the ratio of traffic that N forwards to a bottleneck B. Given a parent P, N computes the ratio of traffic that will reach B through P as the product of the proportion of traffic that it sends to P (α_P) and the ratio of traffic that P forwards to the bottleneck B ($r_{P,B}$). Then, it will sum over all its parents these values. More formally:

$$r_{N,B} = \sum_{P \in Parents(N)} (\alpha_P \times r_{P,B}) \tag{3}$$

Thus, every node computes recursively the ratio of traffic forwarded to a bottleneck by summing over all the parents the ratio of traffic forwarded to it. In the case when a node is itself the bottleneck, the ratio of the traffic that it forwards to the bottleneck is equal to 1 (i.e., $r_{B,B} = 1$).

Next, to estimate the ELT of a bottleneck B, a node needs to know the following information about B:

- $T_{tot}(B)$ - the tot traffic currently handled by the bottleneck;
- $Energy\_spent(B)$ - a constant independent of the traffic of N, that represents the energy spent by the bottleneck B to transmit one bit of information, considering the retransmissions (with ETX) and the rate at which it transmits:
$$\frac{\sum\limits_{P \in Parents(B)} (\alpha_P \times ETX(B,P)) \times P_{TX}(B)}{DATA\_RATE}$$
- $E_{res}(B)$ - a constant representing the residual energy of the bottleneck B.

Knowing this information, a node N can estimate now its impact on the lifetime of the bottleneck B by simply adding the ratio of its own traffic to the traffic of B:

$$ELT(B) = \frac{1}{T_{tot}(B) + \mathbf{r_{N,B}} \times \mathbf{T_{tot}(N)}} \cdot \frac{E_{res}(B)}{Energy\_spent(B)} \tag{4}$$

B. Compact `DIO` Advertisement

A node maintains a list of all its bottlenecks. This information has to be updated and included in each of its `DIO` so that all the nodes in the network have the last updates. Moreover, in this way, we are able to determine partially overlapping paths: several parents may lead to the same bottleneck.

In consequence, a `DIO` will contain a list of bottlenecks with the following information for each of them:

id:	the bottleneck id;
ratio:	the ratio of traffic forwarded by the node to this bottleneck ($r_{N,B}$);
exist_traffic:	existing traffic forwarded by the bottleneck to the sink ($T_{tot}(B)$);
avg_energy:	in order to save memory and since both $E_{res}(B)$ and $Eng\_spent(B)$ represent energies, we can compress this information into a single variable equal to: $\frac{E_{res}(B)}{Eng\_spent(B)}$.

However, a tradeoff exists between the accuracy of the information and the overhead induced. If a node advertises too few bottlenecks, the lack of information could lead to less energy balanced paths. However, if the number of bottlenecks advertised is too large, it can induce more overhead in the network, and make the nodes consume more energy.

We will study this tradeoff in the performance evaluation section and verify that advertising a limited number of bottlenecks is sufficient to balance efficiently the energy consumption in the network.

V. Multipath Construction

ELT is a minimum metric along a path and it cannot be used to compute the *Rank* of a node. We need to define how to choose the preferred parent (i.e., the next hop) and how to compute the *Rank*, to create an energy balanced topology, while avoiding the formation of loops.

A. Preferred Parent Selection

When choosing its preferred parent, a node must consider both its own lifetime and the lifetime of the bottlenecks, in order to estimate which of them becomes the new bottleneck. However, it is not possible to know the ratio of traffic that will be sent to each of the parents before actually choosing the set of parents. Hence, such approach cannot be applied directly.

We propose that during the preferred parent selection we assume that a node will send all its traffic to one single parent. Even if we underestimate the lifetime of the bottlenecks, we are sure to choose as the preferred parent the node maximizing the lifetime of the bottlenecks. In other words, we prefer considering the worst case, to balance more efficiently the energy consumption.

We consequently propose the Algorithm 1 to select the preferred parent (using the notation of Table I). For each possible parent (i.e., a neighbor advertising a *Rank* smaller than itself) a node N will:

1) compute the ELT of all the bottlenecks advertised by a parent P, as if it will send all its traffic to that parent and save the minimum value among all (line 4);
2) compute its own lifetime when choosing this parent and verify if the node did not become the new bottleneck (line 5);
3) remove the traffic to this parent to test the other ones: we have to test all the parents before taking a decision (line 10);
4) choose as preferred parent the node that maximizes the lifetime of the bottleneck with the minimum ELT, itself included (lines 6, 7, 8).

After choosing its preferred parent, the node then removes from its parents set all the nodes having a *Rank* larger than its, computes the new bottlenecks and updates the corresponding information in its `DIO`s.

Algorithm 1: Preferred parent selection

```
   Data: $N$
   Result: preferred_parent of $N$
1  $max\_elt \leftarrow 0$;
2  for $P \in Parents(N)$ do
       // all the traffic is sent to P
3      $\alpha_P \leftarrow 1$;

       // track the minimum ELT (all bottlenecks &
          myself)
4      $min\_elt \leftarrow \min_{B \in Bottlenecks(P)} \{ELT(B)\}$;
5      $min\_elt \leftarrow \min\{min\_elt, ELT(N)\}$;

       // is this parent the best one?
6      if $max\_elt < min\_elt$ then
7          $max\_elt \leftarrow min\_elt$;
8          preferred_parent $\leftarrow P$;
9      end

       // test now the other parents
10     $\alpha_P \leftarrow 0$;
11 end
12 return preferred_parent;
```

B. Rank Computation

The *Rank* of the nodes in the DODAG must strictly monotonically increase from the border router (sink) towards the leaves, in order to avoid the formation of loops. Since the *Expected Lifetime* represents a minimum metric along a path, its value cannot be used to compute the *Rank*: all the nodes in the sub-DODAG will have the same *Rank*.

Similarly to Iova *et al.* [15], we propose that a node computes its *Rank* by adding a constant step value to the *Rank* of its preferred parent:

$$\begin{aligned} Rank(N) &= Rank(P_N) + Rank\_increase \\ Rank\_increase &= Step \times \texttt{MinHopRankIncrease} \end{aligned} \tag{5}$$

where *Step* is a scalar value and `MinHopRankIncrease` the RPL parameter [1].

C. Maintaining the Stability

When a node changes its preferred parent, it triggers the reset of the trickle timer. This means that control packets are sent more frequently and hence, a node consumes more energy. Moreover, parent changes imply also traffic redirection. Since the metric depends on the traffic forwarded by the bottlenecks, all the nodes must in this case update their path metric.

In order to minimize the number of preferred parent changes, while keeping an up-to-date list of parents, we propose the following procedure to update the parent list:

1) a node removes a parent P only if this parent is not useful anymore, i.e., no traffic is actually forwarded to P ($\alpha_P = 0$). If P was the preferred parent, the node re-executes the Algorithm 1 to select the new preferred parent and updates accordingly its *Rank*;
2) any neighbor with a lower *Rank* is inserted in the parent list. The node N updates dynamically the weight of this parent P (α_P).

By only changing the preferred parent when its traffic load reaches zero, we reduce the number of preferred parent changes while keeping up-to-date information about all the parents.

VI. Energy Balancing By Exploiting Multiple Paths

After having constructed several paths with RPL, we have now to address the problem of the forwarding plan. A node must split its traffic among all the available paths, taking into account the lifetime of each bottleneck. We have to propose a heuristic to determine the weights associated to each parent, so that all the paths (and bottlenecks) have finally the same lifetime.

We may use a linear solver to assign a weight for each parent while maximizing the network lifetime (i.e. death of the first node). However, we consider the extra memory and CPU for such resolution are unrealistic for small nodes.

Hence, we present here a greedy algorithm. A node N has to distribute the load to each parent so that it balances the expected lifetime of the corresponding bottlenecks. A node divides its traffic into $\frac{1}{load\_step}$ equal fractions, and assigns sequentially each fraction to the parent which maximizes the minimum lifetime among all its bottlenecks.

Algorithm 2 defines more formally the heuristic. For each parent, N tries to find the best one to send $load\_step$ of traffic, by iteratively testing all of them (line 3):

1) N computes the minimum ELT that would be obtained by increasing the weight of this parent by $load\_step$ (line 4). It considers the lifetime of each bottleneck (line 5) and of itself (line 6);
2) If this minimum value maximizes the network lifetime, it saves the current parent as the best one (line 7-10);
3) N re-initializes α_P to its previous value to test the other parents (line 11);

Finally, N assigns $load\_step$ to the best parent (line 13) and re-iterates with the next $load\_step$ (line 1).

Algorithm 2: Load balancing

```
Data: $N$, $load\_step$
Result: compute $\{\alpha_P\}_{P \in Parents(N)}$ — the ratio of traffic to send to
        each parent;
1  for $i = 1$ to $load\_step^{-1}$ do
2      $max\_elt \leftarrow 0$;
3      for $P \in Parents(N)$ do
           // test this parent P with its new weight
4          $\alpha_P \leftarrow \alpha_P + load\_step$;

           // track the min ELT with this new weight
5          $min\_elt \leftarrow \min_{B \in Bottlenecks(P)} \{ELT(B)\}$;
6          $min\_elt \leftarrow \min\{min\_elt, ELT(N)\}$;

           // is this parent the best one?
7          if $max\_elt < min\_elt$ then
8              $max\_elt \leftarrow min\_elt$;
9              $parent\_max \leftarrow P$;
10         end

           // test each parent before taking a
              decision
11         $\alpha_P \leftarrow \alpha_P - load\_step$;
12     end
13     $\alpha_{parent\_max} \leftarrow \alpha_{parent\_max} + load\_step$;
14 end
```

A small $load\_step$ balances more finely the energy in the network but increases the computation complexity, since a node has to execute the assignment $\lceil load\_step^{-1} \rceil$ times. Some optimizations are possible in the implementation. In particular, for $i > 1$, a node has to recompute the minimum ELT (lines 4-11) only for the parent which was the best one at the previous iteration ($i - 1$). Indeed, the possible weight of all the other parents has already been considered in the previous step.

VII. Simulation Results

We simulated RPL using WSNet, an efficient event-driven simulator dedicated to WSN, which has been extensively evaluated [16]. The results are averaged over 10 simulations with different random topologies. For the traffic, we considered usual CBR convergecast flows.

At the PHY layer, we used a realistic path-loss shadowing model, calibrated with the scenario FB6 (indoor real deployment) presented in [17]: shadowing, path loss = 1.97, standard deviation = 2.0, $Pr(2m) = -61.4dBm$.

We configured RPL as illustrated in Table II. We compared our multipath proposal against the standard, single-path RPL, where the DODAG is constructed using both the residual energy and ELT as routing metrics.

A. Performance evaluation

1) Reliability: Fig. 2 illustrates the complementary cumulative distribution function (CCDF) of the end-to-end Packet Delivery Ratio (PDR) for all the flows. The residual energy presents the lowest PDR since it does not consider the link reliability when selecting the routes. ELT improves significantly the end-to-end reliability: it exploits among others the

the implementation is freely available at https://forge.imag.fr/projects/wsnet-802154/

TABLE II: Default parameters for the simulations

Parameter	Value
Simulation duration	$3600s$
Number of nodes	50
Nb. of bottlenecks advertised	10
Simulated area	300m x 300m
Traffic type, rate	CBR, 1 pkt/min
Data packet size	127 bytes (incl. MAC headers)
RPL	MinHopRankIncrease = 256
Trickle	$Imin = 2^7 ms$, $Imax = 16$, $k = 10$
MAC layer	IEEE 802.15.4 beacon-enabled mode
MAC parameters	BO=7, SO =2
Energy consumption	CC2420 datasheet

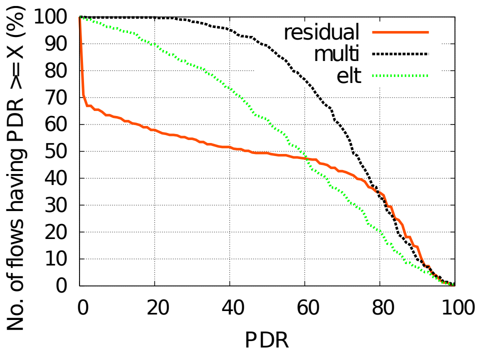


Fig. 2: CCDF of the end-to-end PDR

ETX, which reflects both the energy budget of a link and its reliability. Finally, the multipath version of RPL efficiently takes advantage of the load balancing property to provide the best reliability.

2) Network lifetime: We measured the network lifetime (time before the first node dies) when varying the number of nodes while maintaining the simulation area constant.

Our proposal clearly outperforms the standard RPL, even when ELT is used as the routing metric (Fig. 3). Multipath routing helps balancing more accurately the energy: routing decisions are not binary, and the traffic is spread to all the bottlenecks. The weights accurately *smooth* the traffic redirections.

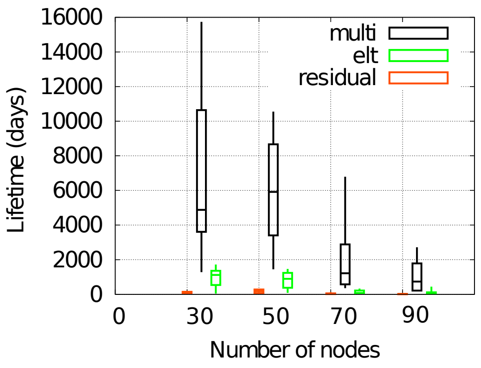


Fig. 3: Network Lifetime (time until the first node dies) in function of the density

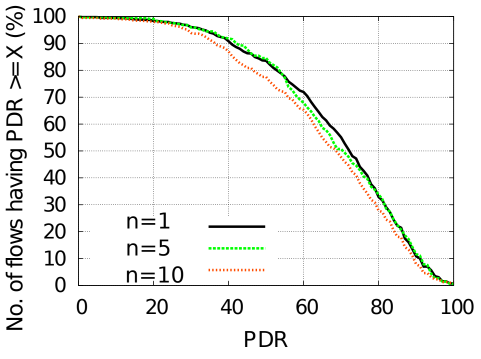


Fig. 4: CCDF of the end-to-end PDR in function of the maximum no. of bottlenecks advertised (n)

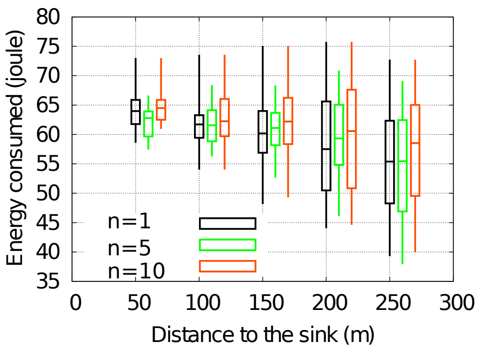


Fig. 5: Energy consumption of the nodes in function of their physical distance from the sink

B. Number of bottlenecks advertised

In the second part of the evaluation, we have investigated the impact of the number of bottlenecks included in the `DIO`.

1) Reliability: In Fig. 4 we plotted the CCDF of the end-to-end PDR for all the flows, in function of the maximum number of bottlenecks advertised by a node. We can see that the PDR is almost the same, no matter how many bottlenecks a node advertises. Indeed, a node takes into account both its ELT and the ELT of the bottlenecks. Since ELT accounts for the ETX, it implicitly takes into account the reliability.

2) Energy consumption: In Fig. 5 we plotted the energy consumed by the nodes during the whole simulation, against their physical distance from the sink (i.e., the border router). No matter the number of bottlenecks advertised, our solution manages to construct an energy balanced topology. Our solution is hopefully not very sensitive to the exact number of bottlenecks to advertise.

A small number of bottlenecks (e.g., 1) tends to increase the energy consumption because it underestimates the consumption of the second most loaded bottleneck. Oppositely, too many bottlenecks (e.g., 10) means the `DIO` size becomes larger, consuming more energy.

Here, we see that adverting 5 bottlenecks is sufficient to balance the energy while having a negligible overhead.

3) Network lifetime: Finally, Fig. 6 illustrates the impact of the number of bottlenecks on the network lifetime. It only depends loosely on the number of bottlenecks.

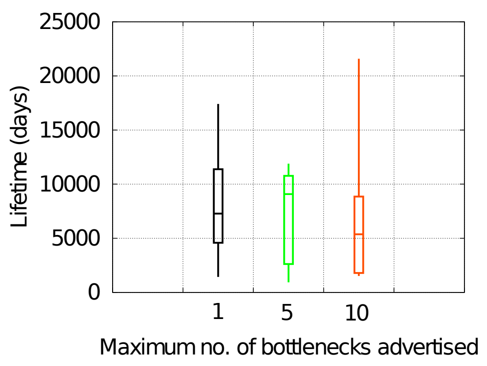


Fig. 6: Expected Lifetime (time until the first node dies) in function of the maximum no. of bottlenecks advertised (n)

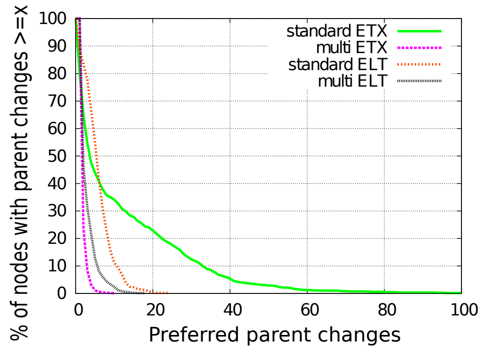


Fig. 7: Stability of the DODAG when using ELT and ETX in standard/multipath

With only one bottleneck, the network lifetime is shorter: the lack of information leads to less energy balanced paths. Oppositely, too many bottlenecks consume more energy on average, impacting negatively the lifetime (10 bottlenecks).

C. Stability

We finally measured the number of parent changes, comparing the standard approach (one preferred parent) and our multipath version. To see the effect of the stability algorithm, we also added results with the multipath approach applied to the ETX metric (Fig 7).

Thanks to the new algorithm for choosing the preferred parent, the multipath solution manages to considerably reduce the dynamics of the network, when both metrics are used. With the standard version of RPL applied to ETX, 30% of the nodes have 10 different parents during the simulation. WIth the multipath approach on the other hand, no node has more than 5 different parents during the whole simulation. The network efficiently converges to a stable set of parents.

VIII. Conclusion and Perspectives

We proposed here a pragmatic design of an energy-balanced RPL. First, we construct a DAG based on the ELT metric which accurately estimates the lifetime of the bottlenecks. Second, we propose a multipath approach to fully exploit the DAG structure. A node exploits all its parents, assigning a weight of traffic to each of them. In this way, a node distributes fairly the energy consumption among all the bottlenecks. Finally, our strategy maintains a stable set of active parents and improves the stability.

We are currently investigating how RPL should integrate inaccuracies in the metric estimation. Indeed, the radio link quality is stochastic, and the routes constructed by RPL should not change if the radio link quality has not *significantly* changed. Furthermore, we will make a generalization of the metric that will account for the energy spent during retransmissions, so that it can be used with other MAC protocols, e.g., the emerging IEEE 802.15.4e-TSCH. We plan also to experimentally evaluate this new multipath energy-balancing version of RPL, to verify it operates efficiently *in vivo*.

References

[1] T. Winter et al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, IETF, 2012.

[2] JP. Vasseur et al. Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks. RFC 6551, IETF, 2012.

[3] P. Kamgueu et al. Energy-Based Routing Metric for RPL. Research Report RR-8208, INRIA, 2013.

[4] M. Radi et al. Multipath routing in wireless sensor networks: Survey and research challenges. *Sensors*, 12(1), 2012.

[5] O. Iova et al. Stability and efficiency of RPL under realistic conditions in wireless sensor networks. In *PIMRC*. IEEE, 2013.

[6] B. Pavković et al. Multipath Opportunistic RPL Routing over IEEE 802.15.4. In *MSWiM*. ACM, 2011.

[7] K.-S. Hong and L. Choi. DAG-based multipath routing for mobile sensor networks. In *ICTC*, Sept 2011.

[8] D. Ganesan et al. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *MC2R*, 5(4), 2001.

[9] Y. Chen and N. Nasser. Energy-balancing multipath routing protocol for wireless sensor networks. In *QShine*. ACM, 2006.

[10] T. Ming-hao et al. Multipath routing protocol with load balancing in WSN considering interference. In *ICIEA*. IEEE, 2011.

[11] N.A. Pantazis, S.A. Nikolidakis, and D.D. Vergados. Energy-efficient routing protocols in wireless sensor networks: A survey. *Communications Surveys Tutorials, IEEE*, 15(2):551–591, Second 2013.

[12] H. Liu et al. PWave: A multi-source multi-sink anycast routing framework for wireless sensor networks. In *Networking*. IFIP, 2007.

[13] H. Yoo et al. GLOBAL: A Gradient-based routing protocol for load-balancing in large-scale wireless sensor networks with multiple sinks. In *ISCC*. IEEE, 2010.

[14] J.-H. Chang et al. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM TON*, 12(4):609–619, August 2004.

[15] O. Iova et al. Improving network lifetime with energy-balancing routing: Application to RPL. In *WMNC*. IFIP/IEEE, 2014.

[16] E. Ben Hamida et al. On the Complexity of an Accurate and Precise Performance Evaluation of Wireless Networks using Simulations. In *MSWiM*. ACM, 2008.

[17] Y. Chen and A. Terzis. On the implications of the log-normal path loss model: an efficient method to deploy and move sensor motes. In *SenSys*. ACM, 2011.