

Received July 26, 2021, accepted August 11, 2021, date of publication August 26, 2021, date of current version September 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3107981

# RPL Cross-Layer Scheme for IEEE 802.15.4 IoT Devices With Adjustable Transmit Power

RAFAEL ESTEPA<sup>1</sup>, ANTONIO ESTEPA<sup>1</sup>, GERMAN MADINABEITIA<sup>1</sup>, (Member, IEEE),  
AND ERNESTO GARCÍA<sup>2</sup>

<sup>1</sup>Department of Telematics Engineering, University of Seville, 41092 Seville, Spain

<sup>2</sup>NTT Ltd., 41020 Sevilla, Spain

Corresponding author: Antonio Estepa (aestepa@trajano.us.es)

**ABSTRACT** We propose a novel cross-layer scheme to reduce energy consumption in wireless sensor networks composed of IEEE 802.15.4 IoT devices with adjustable transmit power. Our approach is based on the IETF's Routing Protocol for Low power and lossy networks (RPL). Nodes discover neighbors and keep fresh link statistics for each available transmit power level. Using the product of ETX and local transmit power level as a single metric, each node selects both the parent that minimizes the energy for packet transmission along the path to the root and the optimal local transmit power to be used. We have implemented our cross-layer scheme in NG-Contiki using the Z1 mote and two transmit power levels (55mW and 31mW). Simulations of a network of 15 motes show that (on average) 66% of nodes selected the low-power setting in a 25 m × 25 m area. As a result, we obtained an average reduction of 25% of the energy spent on transmission and reception of packets compared to the standard RPL settings where all nodes use the same transmit power level. In large scenarios (e.g., 150 m × 150 m and 40-100 motes), our approach provides better results in dense networks where reducing the transmit power of nodes does not translate into longer paths to the root nor degraded quality of service.

**INDEX TERMS** RPL, IEEE 802.15.4, energy efficiency, IoT, Contiki, WSN.

## I. INTRODUCTION

The IPv6 Routing Protocol for Low power and Lossy Networks (RPL) [1], [2] is the IETF standard for routing in the Internet of Things (IoT) widely used in wireless sensor networks. This routing protocol provides a baseline architecture to deal with constrained-resource nodes that communicate using short-ranged noisy links with high variability of data loss, which is a challenging endeavor [3]. Among the design requirements for RPL is to be fit for low-end battery-powered devices severely constrained in memory and processing capabilities. The need to limit battery replacement sets a lifetime goal of several years. As such, energy efficiency becomes a crucial restriction to be carefully considered [4], [5].

This work seeks to improve energy efficiency in IEEE 802.15.4 wireless sensor networks (IETF's choice for IoT), which is not new. RPL already counts with several enhancements geared toward improving energy efficiency, including load balancing [6], [7], multi-path [8], [9], and all sorts

of combined metrics, which usually include a link-quality metric such as the Expected Transmission Count (ETX) [4], [10] plus some energy-related term (e.g., typically node's residual energy) [11], [12]. However, despite the intense research effort, improving energy efficiency persists as a challenge [3], [11], [13]. In the authors' view, two aspects contribute to the complexity of this problem. First, energy efficiency optimization is a broad term that can refer to different goals such as maximizing the overall energy consumed by the network or maximizing nodes' residual energy (i.e., network lifetime) [4], [12], and optimizing one may prevent the optimization of the other. For instance, minimizing the total energy consumed by the network may result in an optimal path in which nodes with high delivery rate drain their battery faster than the rest and, consequently, it fails to attain the maximum possible network lifetime. Second, and following the previous example, there is usually a trade-off between optimizing energy and quality of service (QoS). In order to deal with conflicting requirements in the same application, authors typically suggest the use of composite routing metrics [14]–[16].

The associate editor coordinating the review of this manuscript and approving it for publication was Rakesh Matam<sup>1</sup>.

In this paper, however, we do not set the focus exclusively on the routing function but also on the transmit power setting of nodes. In a typical deployment, all nodes operate with a fixed transmit power setting. This assumption is habitually made in the existing literature, even though it is not uncommon that sensor nodes can use different transmission power levels. For example, the CC2420 transceiver used by the Crossbow TelosB and Zolertia Z1 motes counts with eight different power levels ranging from  $-25$  to  $0$  dBm, the TI CC1200 transceiver used by Zolertia Re-Mote counts with 31 different power levels ranging from  $-16$  to  $14$  dBm (in steps of 1dBm).

Having various levels of transmit power brings up a new degree of freedom to the routing problem. The transmit power impacts node's range (i.e., ability to be seen as a neighbor) and link statistics (e.g., RSSI and ETX observed by neighbors), thereby influencing the routing topology created. We leverage this fact to propose a cross-layer mechanism that operates in three layers: network, data-link, and physical. Nodes maintain fresh link statistics for the set of neighbors discovered with each available transmit power level and self-adjust their transmit power according to the parent chosen by the routing function. We use a routing objective function that minimizes the cumulative value of the one-hop product of ETXs and transmit power of nodes along the upward route from the node to the sink. This single metric compares favorably with the default use of ETX as it encompasses both reliability and transmission energy in the path. But, as a single metric, it inherits its limitations (e.g., it does not maximize the network lifetime). Still, our cross-layer scheme could be easily adapted to other single or composite metrics, such as those suggested in [14], [17], which is left for further work.

*Contribution:* The contributions of this work are:

- We extend RPL's parent selection procedure to the multi-level transmit power scenario.
- We propose a cross-layer approach that dynamically adapts the sensors' transmit power to reduce energy expenditure in the network without impairing QoS.
- Via simulation, we show the benefits of our scheme and analyze the circumstances in which our method is most advantageous.

Observe that our approach can also be used in heterogeneous networks (i.e., a mix of nodes that exhibit different transmit power) even though multi-level transmit power was not featured.

The remainder of this paper is as follows. Section II provides the differences between our work and other cross-layer algorithms for improving energy efficiency in our context. Section III provides a brief overview of the basics of RPL. Section IV describes our proposal, the objective function used and the cross-layer operation. Section V describes the simulation environment created to validate our contribution using Cooja. Simulation results are provided and discussed in Section VI. Section VII highlights the limitation of this work,

and Section VIII concludes the paper and points to further research for continuation.

## II. RELATED WORKS

Cross-layer schemes between the network and the physical layer are not novel. In the wireless network literature, some authors have proposed a joint approach to routing and data rate adaptation [18], [19] where the TX rate is adjusted to minimize energy consumption combined with routing metric. This interesting cross-layer approach addresses the trade-off between route quality and energy consumption via packet service duration. Still, rate adaptation schemes are only feasible in multi-rate wireless technologies such as IEEE 802.11, which fall out of the scope of this work. The standard IEEE 802.15.4 was originally designed for low-power low-rate WSN that produce small-sized packets and operate with a single data rate fixed at 250 kb/s in the 2.4 GHz ISM band (though variants suited to regional regulations or specific applications have been also defined [20]). And although some authors have prospectively explored dynamic link rate adaptation schemes [21], this is not supported by the IEEE 802.15.4 standard. In our work, we choose transmit power instead of data-rate as physical-layer control variable.

A network-mac cross-layer mechanism, named RPL-SCSP was proposed in [22] which combines the ETX and Queue Load to optimize both end-to-end delay and the energy, but it differs from our proposal as it is aimed at providing the network with QoS support and the transmit power of nodes is ignored. More recently, Safaei *et al.* have proposed ELITE [23] a cross-layer scheme that operates at the link layer and RPL by using a routing metric that accounts for information about the radio duty cycle. This scheme achieves a significant energy reduction, but it is restricted to the radio duty cycling protocol ContikiMAC and, like the previous one, it fails to consider nodes with different levels of transmit power.

Autonomous adjustment of node's transmit power has been proposed by Chipara *et al.* [24]. The authors proposed to dynamically adapt the sensors' transmit power to reduce delay in real-time communications. Packets that do not have an urgent deadline are transmitted with lower power to reduce energy consumption. However, this work is different from ours in the following aspects: (a) it creates a new routing protocol rather than using standard RPL, (b) the criterion used to set the node's transmit power level is related to delay and not to energy, (c) it requires to have prior knowledge of the nodes layout and the application-layer traffic pattern. Finally, in [25], Rukpakavong *et al.* pursued, as we do, automatic configuration of nodes' transmit power employing modifications of RPL. However, their work differs from ours in several aspects. First, they only consider link statistics for the maximum transmit power while we keep link statistics for every available transmit power level. Secondly, in their work, RPL selects the best path based solely on ETX, ignoring the transmission power of nodes. As such, the optimal path is identical to that of mf0. Once the topology is set, each

node self-adjusts its transmit power using the received signal strength from the parent's DIOs messages. So it is not a cross-layer scheme as the routing objective function is independent and unaware of nodes' transmit power. In our work, however, the routing metric is aware of both ETX and the node's transmit power (which could lead to routing solutions different from that of mf0), and it is the routing function that chooses the optimal local transmit power for the node.

To the best of our knowledge, our approach is completely original as in the literature is overwhelmingly assumed that nodes operate at a fixed default transmit power level, and the few works that pursue autonomous adaptation of transmit power differ from ours in the aspects mentioned above.

### III. OVERVIEW OF RPL

RPL is a routing protocol that creates a topology using a destination-oriented directed acyclic graph (DODAG) typically oriented towards a sink, border router, or simply put, root node. Although RPL supports dual traffic pattern (i.e., from the root to the devices), this work is focused on multipoint-to-point communication where sensor nodes (moten) send data packets to the root node.

RPL uses control packets called DODAG Information Objects (DIO) so that nodes discover the links that exist between them (i.e., neighbors) and select a preferred one (parent) for forwarding data packets upwards toward the root, hence creating a routing topology. The content of DIOs includes a rank that indicates the relative position of every node with respect to the root and a routing policy in the form of objective function (OF) that specifies how the rank is calculated and how a node selects its preferred parent among its neighbors. For route maintenance, DIOs are broadcasted by nodes according to the Trickle timer [26] that offers a tradeoff between reactivity to topology changes and energy efficiency. As such, DIOs are advertised more frequently when the network is unstable, and instead rebroadcast at an increasingly slow pace while the network is stable.

The RPL standard [1] requires the use of a common OF by all network nodes, but it does not include any OF definition. Thus, nodes running RPL can employ user-defined metrics to describe the properties of a link or node (e.g., link quality, residual energy) and make these metrics available for route selection in DIO messages with a Metric Container sub-option at the choice of the implementer. The IETF, however, have defined (in separate standards) two types of objective functions for use with RPL [27]:

- (a) Objective function zero (OF0) [28], in which a node selects the parent that provides the shortest path to the root, which is calculated by adding a positive scalar that represents link properties (typically the hop count or the ETX) to the rank received from the neighbor. OF0 is designed as default OF allowing interoperation between implementations in a broad spectrum of use cases.

- (b) Minimum Rank Hysteresis OF (MRHOF) [29] that selects routes that minimize a metric but prioritizes stability by using hysteresis to reduce change the preferred parents in response to small metric changes. In the absence of a metric in the DIO Metric Container, MRHOF defaults to using ETX to compute rank, allowing RPL to find the stable minimum-ETX paths from the nodes to the root.

The separation of OF from the core protocol specification allows RPL to be adapted to meet the different optimization criteria required by the wide range of deployments, applications, and network designs. Indeed, there is a plethora of metrics and objective functions proposed in the literature to overcome the natural limitations of RPL and the interested reader is referred to [12] and [11]. Kindly note that our goal is not to compete with these metrics. Indeed, our cross-layer mechanism can be easily adapted to other metrics suggested in the literature.

#### A. RPL PROBING

During DODAG formation, ETX is typically initiated using the RSSI observed during packet reception. But after joining the DODAG, nodes update ETX according to retransmissions (i.e., transmission attempts until ACK is received). This implies that only the ETX with the preferred parent is refreshed, and in the absence of regular traffic, adjacencies could not be tested and repaired if broken. For this reason, some RPL implementations (such as Contiki) allow probing adjacencies by sending periodically unicast DIOs to every RPL neighbor with outdated statistics. Probing also impacts the objective function as outdated neighbors are not eligible to be preferred parent [30], [31]. The periodicity of probing is implementation-dependent, but the node is expected to invoke probing only then there is no data traffic after a certain period (e.g., between 45 and 135 secs in Contiki-NG) or there is no data-link layer feedback. In this work, probing has been enabled.

### IV. DESCRIPTION OF THE PROPOSED SCHEME

The rationale behind our cross-layer scheme is based on the idea that nodes' transmit power and RPL's preferred parent election are not independent. Indeed:

- The transmit power impacts the set of reachable neighbors, or equivalently, the set of eligible RPL parents. Figure 1 illustrates this fact by showing the set of neighbors discovered by node 4 as a result of its transmit power level (four neighbors with maximum power level and three neighbors with the minimum).
- The transmit power of a node may impact the ETX observed by receptors. So, indirectly, the transmit power may also impact the preferred parent selection in OFs that include link-quality metrics such as ETX.

In general, low-power settings may save energy but may degrade QoS (e.g., more retransmissions, connectivity loss, longer paths), which might increase energy consumption.

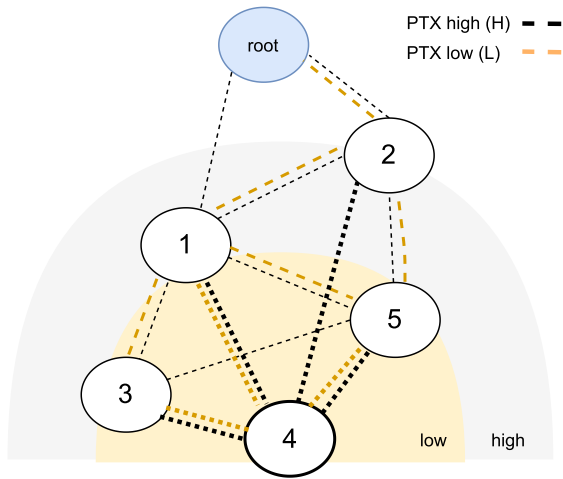


FIGURE 1. Neighbors discovered for two different transmit power level (PTX).

As such, it is not clear whether high or low transmit power will be the most beneficial as it all depends on the prevailing network conditions and nodes' layout. This trade-off between the power settings at the physical and network layers is the bolts and nuts of our proposal.

**A. OVERVIEW**

Our cross-layer mechanism operates in three layers: network, data-link, and physical as illustrated in Figure 2.

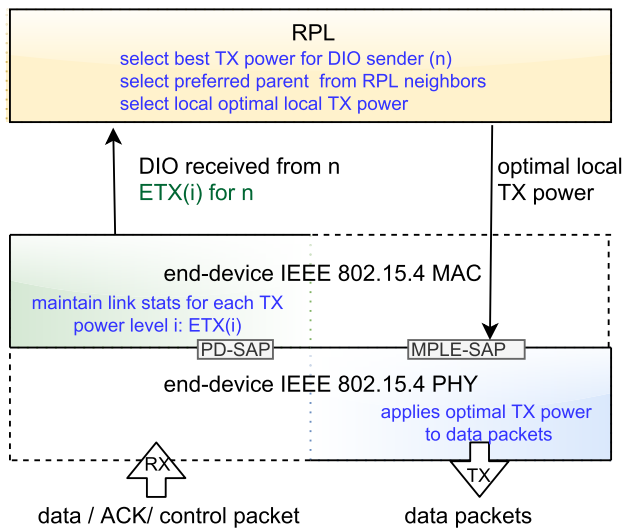


FIGURE 2. End-device cross-layer scheme.

At the physical level, we assume that nodes have a pre-defined set of transmit power levels  $i = 1, 2, 3, \dots, N$ . The physical layer provides data services using the PHY data (PD) service access point (SAP), while the management services are provided using the PHY layer management entity (PLME) SAP. The mac sublayer maintains link quality statistics for every neighbor (i.e., mac address) discovered. In our scheme, the mac sublayer will keep  $N$  values of ETX for every

neighbor (i.e.,  $ETX(i), i = 1 \dots N$ ) since nodes have  $N$  available transmit power levels.

When a DIO is received from a neighbor  $n$ , it is handed to RPL (along with the set of  $ETX(i)$  values associated with the link). Then, we calculate the optimal transmit power level associated with the DIO sender by computing the lowest expected transmission energy expenditure for the link with node  $n$  (i.e.,  $\min_i\{ETX(i) \cdot PTX(i)\}$ , where  $PTX(i)$  is the actual transmit power for level  $i$ ). Let us name this value as the best TX power level (BPTX) for neighbor  $n$ .

Then, the DIO is processed and the rank of neighbor  $n$  is updated by adding the previous lowest expected transmission energy expenditure (i.e., local link metric) to the value received in the DIO metric container (MC).

Finally, the lowest-ranked neighbor is chosen as the preferred parent. Once a preferred parent is chosen, the node's forwarding table is updated accordingly, and our cross-layer algorithm instructs the PHY layer<sup>1</sup> to use the BPTX level associated with the preferred parent for the transmission of upper-layer data packets.

Figure 3 shows the concerned layers of the IoT protocol stack and the supporting tables maintained by each layer when only two levels of transmit power are considered: high (H) and low (L). Observe that node 1 would be elected as preferred parent and the BPTX for that node would be H.

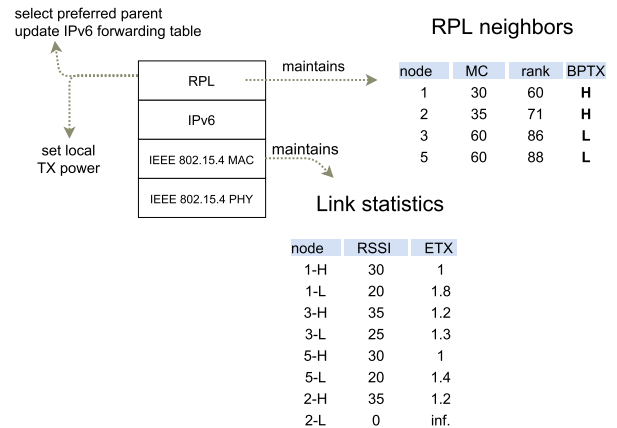


FIGURE 3. Tables and layers in our cross-layer approach.

**B. MAINTAINING LINK STATISTICS FOR VARIOUS POWER LEVELS**

This section expands on how link statistics are collected and maintained for every neighbor and transmit power level available. As with the previous example, and for the sake of simplicity, we will only consider two levels of transmit power for the remainder of this paper: high (H) and low (L). The extension to more levels is straightforward.

<sup>1</sup>It is the mac layer that instructs the PHY layer by using the MPLE-SAP. But for the sake of clarity, we followed the flow of the diagram in our description.

### 1) ENABLERS

Some changes to the standard behavior of the following elements are required in our scheme:

- IEEE 802.15.4 Mac. When receiving a packet, the mac sublayer needs to be aware of the transmit power level used by the sender. For this, we define an Information Element (IE) in the 802.15.4 header that indicates the power level used in the transmission of the frame (i.e., H or L). This is a flexible way to extend IEEE 802.15.4 in an interoperable manner. This sublayer also needs to be able to instruct the PHY layer to use a specific transmit power for data packets as a result of a new parent election.
- RPL DIO announcements. Nodes have to send DIOs at various transmit power levels so that per-level statistics can be kept. Multicast DIOs announcements are still sent according to the Tickle timer, but instead of sending a batch of consecutive DIOs at various power levels, we alternate the transmission power between H and L each time the timer timeout expires. This saves unnecessary overhead at the cost of slower reactivity to changes. In our scenario, however, nodes are static.
- RPL Probing (if active). ETX statistics are periodically refreshed by sending a unicast DIO packet to every outdated neighbor. In our scheme, the target to be refreshed will be not only the link but also the transmission power level (i.e., H and L).

For the remainder of the paper we assume that the previous enablers are in place.

### 2) UPDATING STATISTICS

When a new frame is received, the PHY layer generates a PD-DATA.indication primitive, which contains the *ppduLinkQuality* of the received frame and sends it to the MAC sublayer, which is in charge of maintaining link quality statistics for the mac address and transmit power used by the sender. This tuple (*mac address, transmit power level*) will be used as index in the table of statistics maintained by this sublayer.

After receiving the packet, the mac sublayer processes the IEEE 802.15.4 header and reads both the source mac address and the transmit power level used to transmit the frame (carried in the IE part of the header). Then, ETX(i) is either initialized (the first time a neighbor and power level is discovered) or updated (for existing entries) with every packet sent to that neighbor based on ACK frames (and a retransmission counter) [15]. So the mac sublayer maintains a table such as the one shown in Figure 3.

### C. RPL CROSS-LAYER ALGORITHM

Our cross-layer mechanism is driven by the routing function. In this section, we describe the objective function and RPL operation required by our scheme.

### 1) METOF OBJECTIVE FUNCTION

The parent is elected using following objective function:

$$\min \left( \sum_{j \in \text{path to root}} \text{ETX}_j \cdot \text{PTX}_j \right) \quad (1)$$

where  $\text{PTX}_j$  stands for the transmit power used by node  $j$  for sending data packets. Since ETX accounts for the number of expected transmissions, the physical interpretation of (1) is the transmission power expected in the upward path to the root node. Thus, the OF described in (1) will be referred to as Minimum Expected Transmission Power OF (METOF). Since each node includes its own transmission power settings, METOF is valid for heterogeneous networks (i.e., a mix nodes with different transmit power level).

### 2) ALGORITHM OPERATION

Upon the reception of a DIO from node  $n$ , RPL follows the steps in Algorithm 1.

---

#### Algorithm 1 Cross-Layer Algorithm

---

```

Input: DIO, source address(n), RPL-neighbors
table,PTX(H),PTX(L)
Output: local transmit power, updated table RPL
neighbors, updated ipforwarding-table
/* read link stats for node n */
1 ETX(H) = get-link-stats(n,H);
2 ETX(L) = get-link-stats(n,L);
/* best TX power level for node n
*/
3 if ETX(H)·PTX(H) < ETX(L)·PTX(L) then
4 | BPTX = 'H';
5 else
6 | BPTX = 'L';
7 end
8 local-metric = ETX(BPTX)·PTX(BPTX);
/* update entry for node n in RPL
neighbors table */
9 MC = metric container from DIO;
10 rank = max(MC + Δ, MC + local-link-metric);
11 update rank and BPTX in entry n of RPL-neighbors
table;
/* select preferred parent */
12 for each node N in RPL-neighbors do
13 | best-parent = min (N.rank);
14 | local-PTX = best-parent.BPTX;
15 end
16 update.ipforwarding.table(best-parent);
/* select local optimal TX power */
17 configure.local.TX.pow(best-parent.BPTX)

```

---

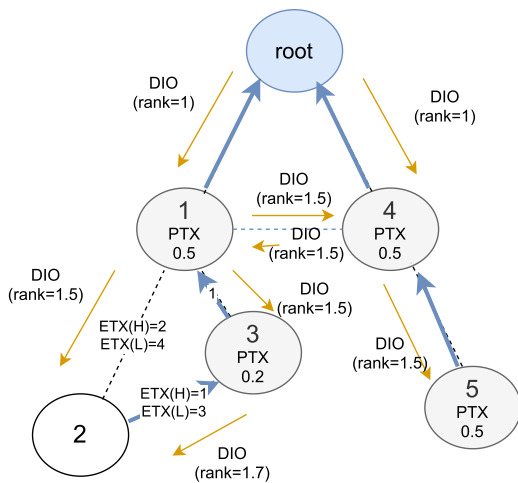
The input of our algorithm is the DIO (that includes the MC with the rank according to METOF), the table of RPL neighbors, and the actual transmit power for levels H and L. As shown in Figure 3, the RPL neighbors table is indexed

by node address, and includes a new column (BPTX) that is the optimal transmission power to be used when transmitting packets to that neighbor. We first read the link statistics corresponding to the DIO sender for all possible transmit power levels. Then, we calculate the BPTX for that neighbor (i.e., the one providing the lowest local metric  $ETX \cdot PTX$ ). Then, we update the rank and BPTX of the corresponding entry in the RPL neighbor table. The rank is calculated as the addition of the value in the metric container plus the local link metric unless this value is less than a minimum increment per hop ( $\Delta$ ), which in our case is set<sup>2</sup> to  $128 \cdot PTX(L)$ .

After updating the corresponding entry, RPL selects the lowest-ranked neighbor as the preferred parent, updating the ipv6 forwarding table accordingly. The local optimal TX power to be used locally will be the BPTX associated with that parent. Finally, the algorithm instructs the mac layer to use this local optimal TX power for transmitting application-level data packets.

**D. EXAMPLE OF OPERATION**

Figure 4 illustrates the operation of our cross-layer scheme in a scenario where sensor nodes have two levels of transmit power:  $PTX(H) = 0.5$  mW, and  $PTX(L) = 0.2$  mW. Links’ ETX are assumed to be 1 except for those values shown in the figure. We assume that all nodes but node 2 have already joined the DODAG and we show the local transmit power chosen by each node (PTX). In this example we analyze the operation of node 2.



**FIGURE 4.** Operation of the objective function.

After receiving the DIO from node 1, node 2 reads the link statistics ( $ETX(H) = 2$  and  $ETX(L) = 4$ ) and calculates its optimal BPTX and local link  $2 \rightarrow 1$  metric (i.e.,  $BPTX = L; ETX(L) PTX(L) = 0.8$  since  $ETX(H)PTX(H)$  would be  $1 > 0.8$ ) for node 1. Having received a rank value in the MC of 1.5, the rank for neighbor 2 would be 2.3. Let us say that then node 2 receives the DIO from node 3. Again, it read the

<sup>2</sup>ETX 128 is equivalent to one hop.

values  $ETX(L) = 3$  and  $ETX(H) = 1$  for the link  $2 \rightarrow 3$  from the table of link statistics. It calculates BPTX for link  $2 \rightarrow 3$  and the local link metric (i.e.,  $BPTX = H; ETX(H)PTX(H) = 0.5$  since  $ETX(L)PTX(L)$  would be  $0.6 > 0.5$ ). The ranking received in the MC was 1.7, so the rank for neighbor 3 would be 2.2. Therefore, node 3 would be chosen as preferred parent and the local TX power would be set to H (0.5).

In our example, node 2 chooses a longer path because the cumulative expected transmission energy is smaller than in the shorter path. Obviously, this depends on both links quality (i.e., ETX) and the actual levels of transmit power used  $PTX(H)$  and  $PTX(L)$ .

**E. IMPLEMENTATION IN CONTIKI-NG**

We have implemented our cross-layer mechanism in Contiki-NG [32]. Contiki-NG started as a major rewrite of the 2017 version of Contiki, with a focus on the most important and stable functionality. As default, Contiki-NG implements a lightweight version of RPL named RPL-lite [33] that lacks support for storing mode and removes the complexity of handling multiple instances and DODAGs.

For the sake of readability, we have moved implementation issues to Appendix A. The interested reader can find there a description of the main elements involved in Contiki’s network architecture and details of the modifications made to accommodate our cross-layer scheme in RPL-Lite, ETX module, IEEE 802.15.4, and the mechanism created to control the transmit power.

**V. SIMULATION SET-UP**

Contiki-NG features Cooja, a java-based simulator of applications that allows one to interact with devices’ code via specialized plugins. We have used Cooja to validate our cross-layer scheme and quantify its benefits versus standard RPL with default ETX metric and transmission power.<sup>3</sup>

The following properties are common to all simulations:

*Node’s hardware:* Cooja supports the Sky and Z1 motes. Both offer multiple transmission power levels, use the radio transceiver CC2420, and are widely referenced in the literature. The power levels chosen for the Z1 mote<sup>4</sup> were:

- $PTX(H) = 0$  dBm (55mW) (default)
- $PTX(L) = -15$ dBm (31mW)

*Terrain size and motes layout:* We simulate two main scenarios. The first one has 15 client nodes (motes) located randomly over a squared area of a small-medium size (e.g., from  $25\text{ m} \times 25\text{ m}$  to  $50\text{ m} \times 50\text{ m}$ ). The second one uses between 40 and 100 motes located randomly over a squared area of  $150\text{ m} \times 150\text{ m}$ . In all cases, a root node (additional to the motes) is always located at the centroid of the square.

*Traffic generation:* nodes run the default *udp-server* and *udp-client* example application, sending a message (‘hello’)

<sup>3</sup>Remember that our cross-layer mechanism can be combined with other metrics suggested in the literature. As such, our approach does not compete with them.

<sup>4</sup>We finally chose the mote Z1 as the flash memory in Sky mote was insufficient for our project.

to the root node periodically. In the small-medium scenario, we simulate 10 hours of operation with data transmissions produced randomly within 10 sec (a total 3 597 data packets generated by each mote). In this way, data traffic will prevail over control traffic. In the large scenario, we simulate 1 hour of operation with transmissions every 60 secs (180 data packets per mote).

**Radio link model:** We use the Unit Disk Graph Medium (UDGM) model with linear path loss (e.g., RSSI and error probability increase with distance). We disabled errors in TX or RX.

#### A. INFORMATION COLLECTED FROM NOTES

In the startup process, each mote initializes a set of specialized plugins to keep track of statistics related to QoS, energy, or RPL behavior. The following plugins have been used:

- **Energest plugin.** This is a Contiki-NG module [15] for gauging how long hardware components are on. Some variables available off-the-shelf are CPU (in active and low-power mode<sup>5</sup>), or how long the radio has been transmitting or listening (which could be either receiving or idle). We have added variables to recognize transmission power levels.
- **PowerTracker plugin.** This Contiki-NG plugin let us find how long the radio is on, transmitting, or receiving. We use it to complement Energest plugin and find the actual time spent in RX and idle (as Energest does not discriminate them).
- **Event-count plugin.** We have developed this plugin for recording events or statistics of interest. These statistics are related to RPL performance (e.g., delay to join the DODAG, number of changes of preferred parent), RPL messages sent by the node at each power level, application-level counters (e.g., application packets sent at each power level), and general traffic counters such as the number of frames sent and received, number of retransmissions, etc. The final list of indicators generated per simulation (and a brief description) is listed in Table 1.

The statistics related to energy consumption have been calculated by multiplying the time spent on each state of the hardware component (as measured by the plugins) by the corresponding values of current intensity and voltage found in the CC2420 and T1 datasheets shown in Table 2. As such, the energy of the mote is calculated as:

$$E_{mote} = V \cdot (T_{CPU} \cdot I_{CPU} + T_{idle} \cdot I_{idle} + T_{TX-H} \cdot I_{TX-H} + T_{TX-L} \cdot I_{TX-L} + T_{RX}) \quad (2)$$

where the time spent on each state from Equation 2 is collected from the simulator.

<sup>5</sup>A low-power mode of CPU exists  $E_{LPM}$ , but its contribution to the overall energy budget is negligible due to a very low current of 0.00015mA, being about the same in all scenarios and representing less than 0.1% of the overall energy.

**TABLE 1. Main statistics collected in our simulations.**

	indicator	description
traffic and QoS	APP	# 'hello' packets successfully sent and received
	Delay(ms)	average end-to-end delay for 'hello' packets
	TOTAL TX	# frames sent by node
	TOTAL RX	# frames received by node
	TX_NO_ACK	# unicast packets unacknowledged
	NUM_RTX	# retransmissions
	FORWARDED	# packets forwarded (i.e. relay function)
RPL	M-DIO	# multicast DIOs sent
	U-DIO	# unicast DIOs sent
	DIS	# DIS sent
	DIOs proc.	# DIOs received and processed
	Parent_switch	# times a node switches its preferred parent
	JoinedDelay(s)	average delay to join the DODAG
Energy	Energy-CPU (mJ)	energy spent by the CPU
	Energy-TX (mJ)	energy spent by radio in transmission
	Energy-RX (mJ)	energy spent by radio in reception
	Energy-IDE (mJ)	energy spent by radio in idle mode
	Energy-TOTAL	overall energy expenditure (CPU + radio)

**TABLE 2. power coefficients from CC2420 and T1 datasheets.**

voltage	3.2volts
$I_{CPU}$	4.5mA
$I_{IDLE}$	0.26mA
$I_{TX-H}$ (0dBm)	17.4mA
$I_{TX-L}$ (-15dBm)	9.9mA
$I_{RX}$	18.8mA

#### 1) SIMULATION PROCESS

We used the plugin Simulation Script Editor to automate the simulation process and collect the statistics of interest from motes via UART at the end of each simulation run. Basically, our script creates a square area of a specific size, distributes motes randomly over the area, initializes the simulation, collects the results from each mote using the plugins mentioned above, and calls a post-processing script to get the final results.

Simulations are repeated 25 times for better significance of results. On each repetition, the motes are randomly distributed over the square area. Thus, all results provided in the remainder of this paper represent the average values of these 25 simulation runs. We have also saved and reviewed the results obtained after each simulation run to validate exceptional cases such as disconnected motes.

## VI. RESULTS

Using the settings described above, we have simulated the operation of our cross-layer scheme x-RPL(METOF) and standard RPL(MRHOF) with default (H) transmit power. The comparison of both will allow one to validate, gain insight and quantify the benefits of our proposal.

#### A. SMALL SCENARIO: 15 NODES IN 25 m × 25 m

Simulation results for the small scenario are shown in Table 3. Indicators express cumulative values (the sum of the 15 motes) except for delay-related statistics, which represent average values. Indicators ending in H or L stand for packets transmitted using the corresponding power level. Detailed (i.e., per mote) results are provided in Appendix B for the

TABLE 3. standard RPL vs cross-layer RPL in 25 m × 25 m.

	RPL(MRHOF)	x-RPL(METOF)
APP-H	53929.6	18224.8
APP-L	0.0	35704.2
Delay(ms)	28.3	28.3
TOTAL TX	60848.7	60820.0
TOTAL RX	15874.6	12287.3
TX_NO_ACK	1.0	1.6
NUM_RTX	311.1	445.2
FORWARDED	102.9	53.4
M-DIO-H	642.4	339.2
M-DIO-L	0.0	302.2
U-DIO-H	6173.1	4918.1
U-DIO-L	0.0	1278.1
DIOs proc.	15664.2	12126.2
Parent_switch	16.0	17.2
DIS	16.4	16.4
JoinedDelay(s)	47.6	47.9
Energy-CPU	136540.2	135148.7
Energy-TX	6597.4	4970.8
Energy-RX	123962.0	92345.3
Energy-IDLE	733143.6	733859.1
Energy-TOTAL	1000243.2	967172.8

interested reader. A description and analysis of the results from Table 3 follows.

1) OBSERVED NETWORK BEHAVIOR

An area of 25 m × 25 m is likely to produce a star topology since the root is in range for all motes. Examining the traffic and QoS counters from standard RPL, it can be concluded that traffic load is low, links are stable, and the routing topology is shaped like a star. Some evidence supporting these conclusions are scarce retransmissions -affecting less than 0.07% of the packets-, no changes of preferred parent, and lack of relay operation in motes according to the number of packets forwarded (which is residual and primarily due to the initial transient period of topology formation). Furthermore, the packet delay is low (28 ms), and in Appendix B the reader can verify that it is similar in all nodes.

The behavior observed in x-RPL(METOF) is not different than in standard RPL. Scarce retransmissions (less than 1%), stable links (few changes of preferred parent), lack of relay operation, and a similar low delay are signs of a star-like routing topology in an uncongested network. The counters in x-RPL reveal that (on average) almost 66% of packets are transmitted in low-power (L), which can be interpreted as 2 of 3 nodes (on average) operate in low-power (L) in the simulated scenario. Precisely, due to this, nodes in x-RPL receive on average 23% less packets than nodes running standard RPL due to fewer neighbors in range.

2) ENERGY PERFORMANCE

Examining the energy-related statistics in Table 3, it can be concluded that our cross-layer approach compares favorably with standard RPL. Not only the cumulative energy spent in packet transmission is reduced 25%, but also in reception (a 26%) due to some motes using the L transmit

power level.<sup>6</sup> The latter is particularly important in the light of the energy breakdown shown in Figure 11 since the energy spent in reception is significantly greater than in transmission in our simulations (12.4% vs less than 1% of the total energy expenditure in standard RPL).

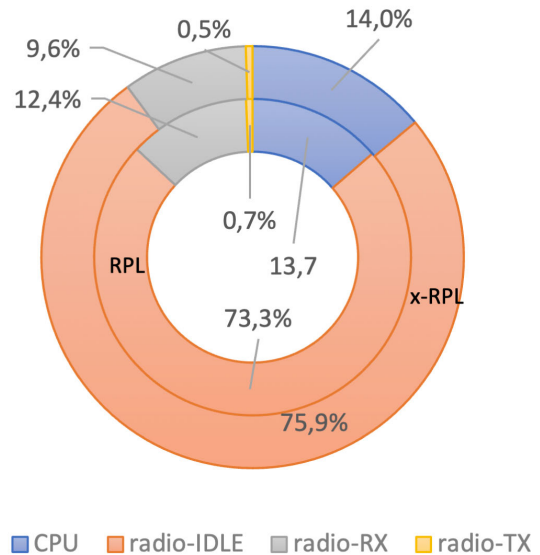


FIGURE 5. Energy breakdown in both schemes.

As shown in Figure 11, most of the energy is devoted to the idle radio state (over 73% in both schemes). This is why the total energy savings with our scheme are more modest (around 3%) in the absence of a radio duty cycle mechanism. Note, however, that if TSCH was in place, most of the energy spent in idle state would be eliminated.

Another result attributable to having fewer neighbors is processing fewer DIO packets which translates to less CPU usage. As such x-RPL exhibits CPU energy savings of 1% with respect to standard RPL.

So we can conclude that in the 25 m × 25 m scenario our cross-layer scheme reduces energy consumption while maintains QoS with respect to standard RPL with default transmit power H in all nodes.

B. OTHER ASPECTS IN SMALL-MEDIUM SCENARIOS

In this section, we analyze additional aspects not addressed in our previous scenario by performing additional simulations (all with 15 motes). For the sake of conciseness we only show the results relevant for the analysis provided.

1) IMPACT OF PROBING

Probing has been enabled in the simulated scenarios. Although probing improves stability and convergence speed, it also increases traffic by sending unicast DIO packets to

<sup>6</sup>Note that Cooja implementation of CC2420 does not perform mac filtering at PHY and as such, all packets are received.



outdated neighbors, a traffic increase of about 10% in our simulations (41 DIO packets per hour and node).

Results obtained without probing are somehow similar to that with probing. Logically, the lack of unicast DIO packets reduces the energy spent in transmission, reception, and CPU accordingly, which is shown in Figure 6.

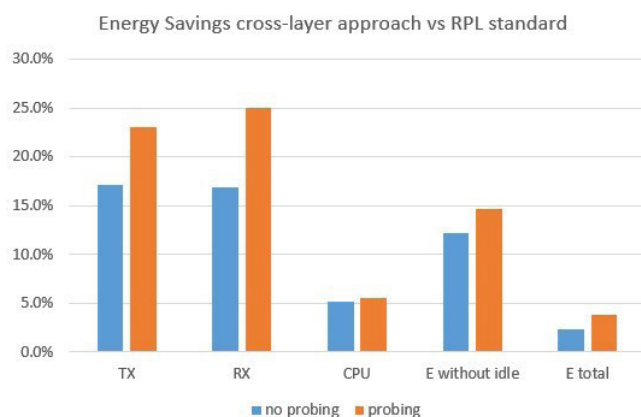


FIGURE 6. Power savings with and without probing.

Since the traffic load simulated is low, the benefit of using probing is diluted. In more complex scenarios, or under poor network performance, however, probing should be examined in the light of cost (more traffic and energy consumption) and benefit (i.e., better adaptation to topology changes).

### 2) 15 MOTES IN A LARGER AREA (FROM SMALL TO MEDIUM)

In larger terrains, more nodes in x-RPL are expected to set their transmission power to H to avoid disconnection. For this reason, we have gradually increased the terrain size in our simulations up to 50 m × 50 m. Figure 7 shows the percentage of packets sent using the low-power setting and the energy savings achieved by our scheme for each size simulated.

Results in Figure 7 confirm our hypothesis. With greater sparsity, more nodes set their transmission power to H to avoid disconnection. As such, differences between our cross-layer scheme and standard RPL tend to diminish. All nodes were connected in all simulation runs, and the delay achieved was similar to that of 25 m × 25 m (i.e., about 28ms), suggesting tree-like topologies. This result is also due to the effect of the power levels chosen, which produces that nodes only select L if it does not add an extra hop in the path.

### 3) CHANGING DEFAULT TRANSMISSION POWER TO LOW

Using H as default transmit power is more cost-effective than L as it provides better coverage, most applications are not traffic-intensive, and it also improves mobility support. Nevertheless, we have explored the results that would have been obtained had the default transmit power been set to low (L). Lower transmission power is expected to produce shorter coverage, which may lead to longer paths. In the worst

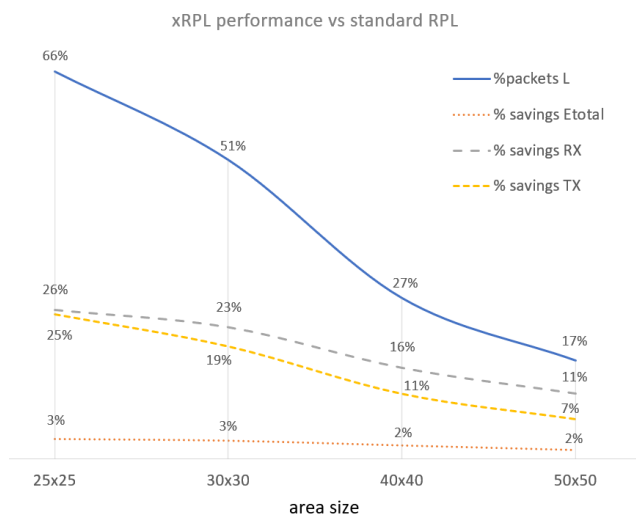


FIGURE 7. Percentage of packets transmitted in low-power setting and energy savings for different area sizes.

case, isolated nodes will not be able to send data packets (depending on the nodes' layout and area size). For this reason, we have checked the number of times that nodes are disconnected (i.e., unable to send 'hello' packets) after every simulation run for different area sizes. Figure 8 provides the rate of disconnection for every node for different area sizes.

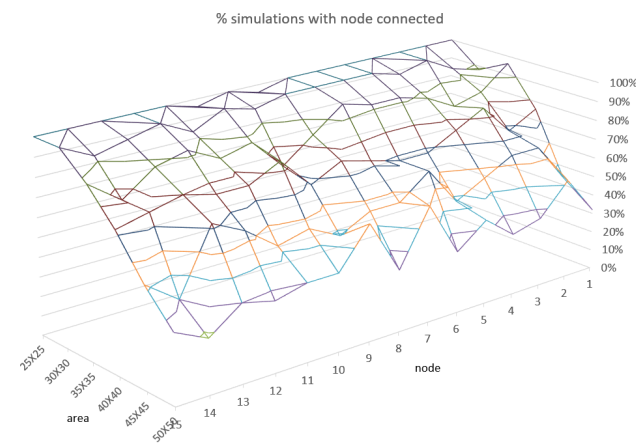


FIGURE 8. Percentage of simulation runs without nodes disconnected.

Results in Figure 8 show nodes disconnected in some simulation runs for all area sizes simulated. In the scenario of 25 m × 25 m two nodes were disconnected in 8% of the simulations. The number and frequency of disconnected nodes increase with node sparsity. Indeed, only 53% of nodes were always connected (i.e., over the 25 simulation runs) in a 30 × 30 m scenario. For larger areas, we could not find a single node connected through all simulations. As expected, QoS degrades with lower transmission power, which is evidenced not only by disconnected nodes but also by increased packet delay (from 28ms to 37ms on average) and increased

retransmissions. So we can conclude that a default transmission power of L is not generally feasible due to the risk of unconnected nodes.

**C. LARGE SCENARIO: 150 m × 150 m**

Our small-medium (15 nodes in star topology) provided insight into our cross-layer approach, but it is small in size and number of nodes compared with most related works.<sup>7</sup> For this reason, we simulate a larger scenario of 150 m × 150 m increasing the number of nodes from 40 to 100. Recall that in this scenario we simulate one hour of operation and the 'hello' packet generation has been set to 1 packet per minute.

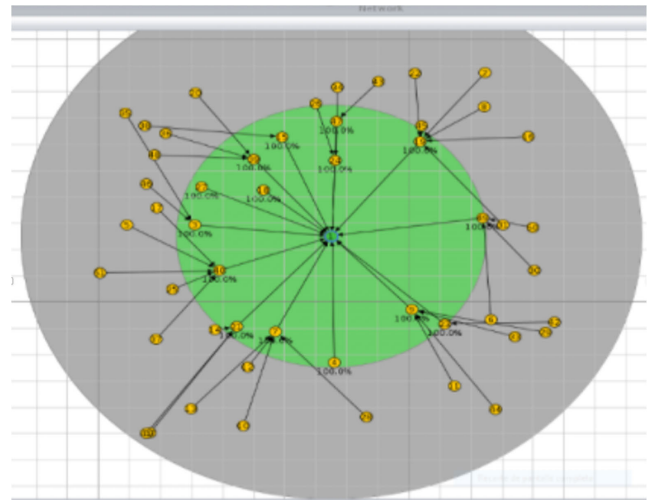
**TABLE 4. x-RPL(METOF) performance 40-100 nodes (area 150 × 150 m<sup>2</sup>).**

nodes	40	50	60	80	100
APP-H	6980.3	8581.6	10154	13126	15877
APP-L	182.8	373.9	579.6	1193.7	1961.9
%APP-L	2.55%	4.18%	5.40%	8.34%	11.00%
Delay(ms)	49.83	50.99	53.74	56.45	63
TOTAL TX	18316	23263	28731	39108	52292
TOTAL RX	15246	20512	27161	45267	79891
TX_NO_ACK	2.9	7	9.6	26.2	52.8
%forwarded	29.97%	30.38%	32.19%	32.81%	36.23%
M-DIO-H	446.9	543.4	656.1	1035.6	1715.2
M-DIO-L	336.1	415.4	494.4	715.6	920.2
U-DIO-H	4669.7	6032.4	7261.2	9722.5	12166
U-DIO-L	209.9	248.7	336.2	480.8	704.2
DIOs proc.	9521.6	13086	17382	30859	54601
Parent_switch	1.83	2.22	2.52	3.3	3.83
joindelay(ms)	54.9	54.02	54.33	63.23	108
Energy-CPU	1459	1613	1747	1961	2160.1
Energy-TX	63.45	65.16	67.73	70.3	75.76
Energy-RX	843.2	1038.1	1290.2	1778.4	2387.5
Energy-Idle	14701	14697	14690	14680	14666
Energy-TOTAL	17067	17413	17795	18490	19289

Table 4 shows the results obtained in x-RPL. The first group of rows shows cumulative values except for delays (which show average values). We have added the percentage of low-power packets over the total number of application-level packets sent (%APP-L), and the percentage of packets forwarded over the number of packets transmitted (% forwarded). The percentage of nodes operating at low-power ranges from 2.5% to 11%. When nodes are more sparse, only those with very close parents (notice that our L setting has a range of about 6 m) use the low-power setting. This can be better observed in a scenario taken from one of our simulations shown in Figure 9 (where the grid represents 10 m × 10 m).

As expected, traffic increases with the number of nodes, as it does the number of nearby nodes, which tends to cause longer paths. Some evidence supporting this is the increment in % of forwarding (from 30% to 36%) and packet delay (from almost 50 ms to 63 ms). The packet delivery rate is very close to 100%. Looking at the routing performance, the delay in joining the DODAG and the number of parent switching

<sup>7</sup>Most papers use an area size between 100 m × 100 m and 200 m × 200 m, and according to [23] a study of 39 papers showed that the average number of nodes simulated was 49.4.



**FIGURE 9. One of the scenarios simulated for 50 nodes.**

**TABLE 5. standard RPL performance 40-100 nodes (area 150 × 150 m<sup>2</sup>).**

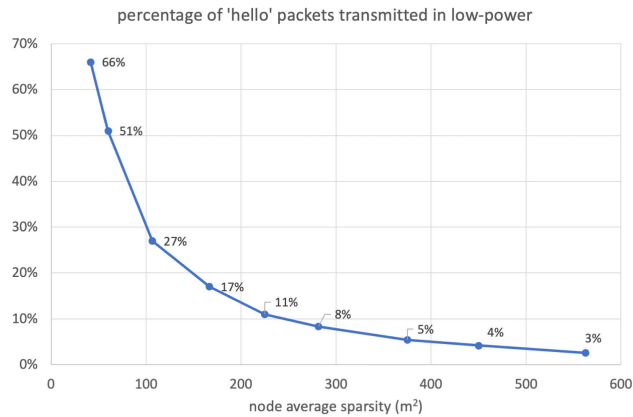
nodes	40	50	60	80	100
APP	7162	8956	10734	14314	17876
Delay(ms)	50.45	51.18	52.28	57.72	59.75
TOTAL TX	18386	23231	28131	39152	50125
TOTAL RX	18527	25415	33255.4	55734.9	90357
NUM_RTX	300.8	580	870.4	2142.2	3401.8
TX_NO_ACK	1.9	3.9	7.6	22.3	32.3
%forwarded	30.54%	30.60%	31.12%	33.48%	34.46%
M-DIO	776.2	965.8	1138.4	1585.8	2320
U-DIO	4831.5	6201	7502.1	10140	12653
DIOs proc.	12675	17943	23991	41015	69214
prnt-switch	1.83	2.17	2.36	3.49	3.52
joindelay(ms)	55.2	53.26	54.18	64.85	79.12
Energy-CPU	1509.1	1682.3	1829.1	2080.8	2308.2
Energy-TX	64.99	66.78	68.11	72.52	75.64
Energy-RX	879.0	1080.3	1321.7	1927.6	2521.8
Energy-Idle	14700	14696	14690	14676	14663
Energy-TOTAL	17153	17525	17909	18757	19568

made by nodes (i.e., instability) grows with the number of nodes ranging from 1.83 to 3.83. Indirectly this increases the odds of nodes switching their transmit power setting. The third group of rows shows the per-node (not cumulative) power expenditure. Logically, the energy consumption grows with the number of nodes mainly due to the energy spent in receiving and processing more packets.

Table 5 shows the results obtained with RPL for comparison. The QoS performance is very similar to x-RPL for a similar number of nodes. Packet delay increases with nodes (ranging from 50.45 ms to 59.75 ms) as it does the path length according to both delay and the percentage of relay. The number of unacknowledged packets is negligible, providing almost 100% of packet delivery rate. The most notable difference is that with standard RPL the number of packets received and DIOs processed is significantly greater than with x-RPL, which translates into more energy consumption in these two components. From Tables 4 and 5 it can be concluded that QoS performance in x-RPL is similar to standard RPL in the simulated scenarios.

#### D. INFLUENCE OF SPARSITY

Using the simulations from the previous sections, we have calculated the sparsity of nodes for each scenario dividing the area size by the number of nodes simulated. This section studies the percentage of application-level packets sent using the low-power setting  $L$  according to node sparsity.



**FIGURE 10.** Percentage of low-power application packets with node density.

Taking the rate of low-power packets transmitted in Figure 10 as a rough indicator of the percentage of nodes self-configured in low ( $L$ ) transmit power setting, we can observe that this number lessens with sparsity in a non-linear way. This confirms our initial judgment that our scheme provides better results in highly dense scenarios.

#### VII. LIMITATIONS

This work is only a first step in exploring energy efficiency through a cross-layer PHY-RPL approach. As such, some limitations prevent the excessive generalization of our results and pave the way for further research. The first and most obvious is that our scheme only makes sense in networks where nodes can operate with different transmit power levels. Other limitations include:

- Our choice for  $H$  and  $L$  power levels in the Z1 mote (55mW and 31mW respectively) and our OF have resulted in nodes choosing  $L$  only if it does not add more hops to the path. As such, different power levels or a different minimum hop penalty in the calculus of the rank could have resulted in other behavior.
- Our OF considers a single metric that may not be suited for some applications. RPL counts with a good deal of enhancements [34], including single and combined metrics suggested in the literature [12]. Our cross-layer scheme could be enhanced with some of these metrics (e.g., offering a balance between network life and energy efficiency).
- Bi-level vs Multi-level. Using two levels of transmit power has produced a simple behavior where only very close nodes set to  $L$ . If multiple levels of transmission power were available (e.g., 8 or 16) the behavior would

be different and there would be a cost-benefit analysis to be studied.

#### VIII. CONCLUSION AND FURTHER WORK

This paper has explored a novel scheme for improving energy efficiency in IEEE 802.15.4 IoT devices with adjustable transmission power levels. In our cross-layer approach, each node maintains fresh statistics for each neighbor and power level and self-adjusts its transmission power according to RPL's choice of the preferred parent.

Our cross-layer scheme reduces energy consumption without impairing QoS in nodes when compared to standard RPL. This energy reduction is based on savings in transmission, but mostly in fewer packets being received and processed. The percentage of nodes configured in low-power setting strongly depends on the nodes layout, decreasing notably with nodes sparsity.

This preliminary work still has open questions that require complementary research. Some of the issues left for further work follow. An extension to multiple power levels (i.e., more than 2), including an analysis of the sensitivity of the levels to the objective function, nodes' layout, resource consumption, and network performance. Another area of interest is in the objective function. Many metrics suggested in the literature can be combined with our cross-layer approach providing enhanced performance. Another subject of interest for research purposes could be exploring the feasibility of a joint RPL approach that combined dynamic transmit power and data-rate adaptation in IEEE 802.11 WSN. Finally, mobility introduces a major challenge for RPL [35]–[37] and the applicability of our scheme should be the subject of further study in this area.

#### APPENDIX A IMPLEMENTATION OUTLINE USING CONTIKI-NG

Figure 11 shows the main modules involved in the *netstat* pile used by Contiki-NG as well as the relationship between these modules. They can be described as follows:

- **Network.** This block is in charge of the transport-level and network-level service, excluding the routing function. It creates, sends, and receives data packets for the application. It uses a shared buffer (*UIPbuff*) with the routing function. Its modules are:
  - *udp-socket*. This module allows one to use UDP connections to send and receive messages between nodes. It is serviced by the *ipv6* module.
  - *icmp6*. Is in charge of sending and receiving control messages for IPv6. For our purposes, it will send and receive RPL's DIO, DIS, and DAO packets interfacing with the routing module.
  - *ipv6*. This core module is responsible for the creation and processing of datagrams and the forwarding function. It also maintains a list with the network and link addresses of each neighbor and the forwarding table (assisted by the routing module).

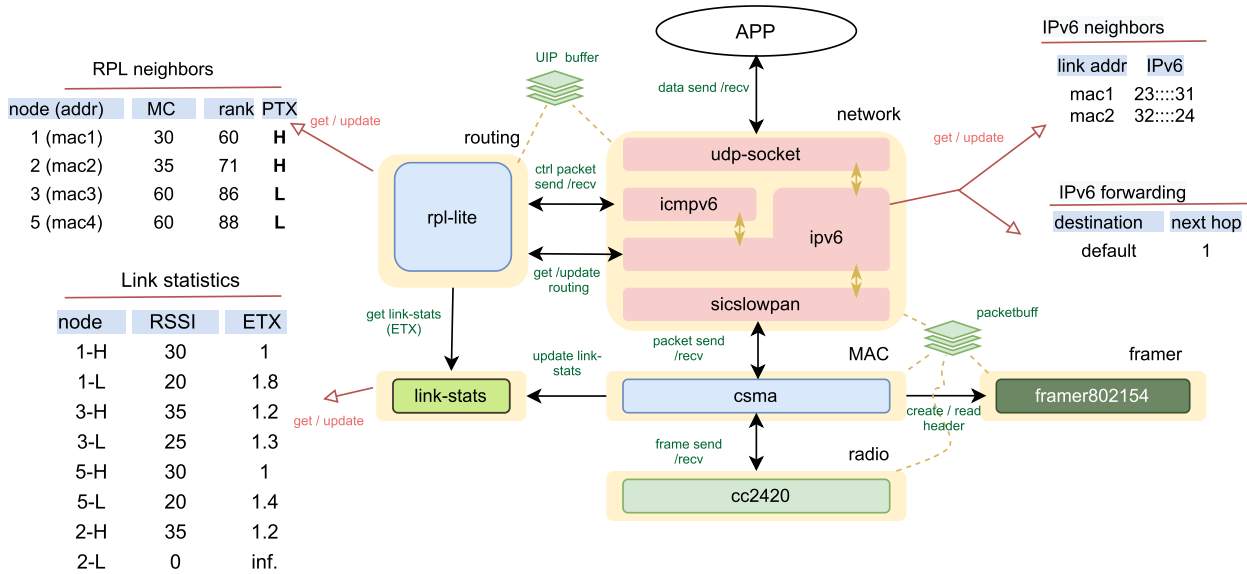


FIGURE 11. Main components of the network protocol stack in Contiki-NG.

– sicslowpan. It performs header compression /decompression for IPv6 packets, featuring fragmentation / reassembly if necessary. Once a packet is ready to go, it reads it from the *UIPbuff* buffer and places it as payload in a buffer named *packetbuff* accessible to the mac sublayer.

- **Routing.** This module is in charge of routing, including the processing of source routing headers [38]. It also generates and processes RPL control messages (DIO, DIS, and DAO) which are passed to/from the *icmpv6* module via the *UIPbuff* buffer. When a preferred parent is selected, this module instructs the *ipv6* module to update its forwarding table. This module maintains a neighbors table with relevant information from each neighbor (e.g., rank, routing metric) who are identified by their link-layer address. For local statistics such as ETX, this module queries the link-statistics module.
- **MAC.** This module is a basic implementation of IEEE 802.15.4 with CSMA/CA. It reads packets placed by *sicslowpan* at the *packetbuff* and composes IEEE 802.15.4 frames using the *framer* module. Then, it sends the frame to the radio layer (using the shared buffer *packetbuff*) according to the CSMA/CA algorithm, waiting for the reception of ACK and returning the status of the transmission. Every time that a frame is sent or received, it calls the link-statistics module to update RSSI and/or ETX.
- **link-statistics.** This module calculates and maintains link statistics using information conveyed by the MAC layer with every packet sent (e.g., number of retransmissions, the status of the transmission) or received (e.g., RSSI). It stores statistics for every mac address

discovered in a table that includes the value of RSSI, ETX, and additional information such as counters of the number of transmissions and acknowledgments for a link or a freshness indicator.

ETX is either initialized (the first time a neighbor is discovered) or updated (for existing entries) with every packet sent to that neighbor as follows:

- ETX initialization. As default ETX is initiated to a default value (e.g., 2) or else via RSSI of a received packet, in which case ETX takes a value between 1 and 3 if RSSI is between  $-60$  and  $-90$ .
- ETX update. ETX can be configured to be carried out in two possible ways: using Exponential Weighted Moving Average (EWMA)(which is the default) or using the counters of transmissions and acknowledgments. In both methods, a packet unacknowledged adds a penalty of 12 to the number of transmissions counter. In the EWMA method, ETX for the  $i$ -th update is calculated as:

$$ETX(i) = ETX(i - 1) \cdot (100 - \alpha) + numtx \cdot \alpha \tag{3}$$

where  $\alpha$  is 10 if link statistics are fresh and 25 if not (according to default values).

- **Radio** the chip radio used in this project is CC2420. It sends and receives frames to/from *packetbuff* and provides PHY statistics to the mac layer. Packets are transmitted using the power level indicated in a register.

A. IMPLEMENTING OUR SCHEME IN CONTIKI-NG

This section outlines the main modifications made to the original netstack.

TABLE 6. Results 25 m × 25 m standard RPL(MRHOF).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
APP-H	3594.6	3595.7	3594.6	3595.4	3594.7	3595.3	3594.1	3595.6	3594.9	3594.9	3596.8	3595.8	3596.5	3595.4	3595.7
APP-L	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Delay(ms)	28.4	28.0	29.0	28.0	28.6	28.2	28.9	28.1	29.1	28.0	28.1	29.1	27.9	28.6	27.7
TOTAL TX	4051.5	4058.3	4048.0	4050.3	4048.9	4051.1	4050.8	4051.7	4076.5	4050.3	4073.2	4049.5	4091.0	4050.7	4046.9
TOTAL RX	986.9	1263.4	959.7	1097.1	916.3	1152.1	937.4	1078.7	887.2	1061.9	1160.1	1040.6	1287.2	1026.7	1019.3
TX_NO_ACK	0.1	0.1	0.1	0.1	0.1	0.0	0.1	0.2	0.0	0.0	0.1	0.1	0.0	0.0	0.0
NUM_RTX	15.6	18.2	28.0	17.0	16.5	25.1	21.8	15.8	34.0	18.8	17.0	34.8	15.8	17.1	15.5
FORWARDED	0.9	1.6	1.4	3.4	1.0	3.5	1.0	1.1	29.8	1.0	18.2	1.0	37.7	1.1	1.0
M-DIO-H	42.8	43.8	42.4	42.6	42.6	43.1	42.3	42.3	42.8	42.4	44.3	42.3	44.0	41.9	42.7
M-DIO-L	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
U-DIO-H	413.2	417.2	409.6	408.9	410.6	409.2	413.4	412.7	409.5	412.0	413.9	410.4	412.8	412.3	407.5
U-DIO-L	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PROCESSED_DIO	981.6	1251.6	955.7	1083.4	913.1	1139.4	935.1	1069.5	857.6	1053.5	1127.7	1033.3	1236.4	1020.5	1006.0
PRNT_SWITCH	1.1	1.0	1.1	1.1	1.0	1.3	1.0	1.2	1.0	1.0	1.0	1.0	1.0	1.0	1.1
DIS	1.0	1.6	1.4	1.0	1.0	1.0	1.0	1.1	1.0	1.0	1.1	1.0	1.2	1.1	1.0
JoinedDelay(s)	54.7	45.9	54.1	45.0	54.7	45.6	55.7	46.0	59.1	48.8	33.9	46.3	34.1	46.7	44.3
Energy-CPU	9131.8	9530.7	9058.3	9309.6	8921.7	9373.0	8799.1	9272.6	8811.2	9188.9	8791.9	9008.5	9151.2	8885.3	9306.3
Energy-TX	435.2	439.3	436.3	436.2	434.2	438.3	435.1	436.4	444.7	442.4	445.4	444.0	447.7	441.9	440.4
Energy-RX	8267.3	8266.6	8266.0	8270.8	8265.2	8268.5	8263.2	8269.9	8250.3	8263.2	8261.4	8261.5	8257.5	8262.5	8268.1
Energy-IDLE	48876.2	48876.2	48876.1	48876.3	48876.5	48876.2	48876.3	48876.2	48876.4	48876.2	48876.2	48876.2	48876.1	48876.2	48876.3
Energy-TOTAL	66767.1	67169.3	66693.3	66949.5	66554.2	67012.6	66430.3	66911.7	66439.3	66827.3	66431.6	66646.8	66789.0	66522.5	66947.5

### 1) MECHANISM TO HANDLE THE TRANSMIT POWER LEVEL

We have added a global variable to RPL-Lite that indicates the transmit power level to be used by the radio. The mac layer will read this variable and change the radio power register accordingly right before instructing the PHY layer to transmit a new frame that encapsulates application data, restoring it to default after the transmission.<sup>8</sup>

### 2) CHANGES IN IEEE 802.15.4

Frames that need to inform about power settings include a new Information Element (IE) header (5 bytes) that extends the IEEE 802.15.4 to indicate the transmission power level used.<sup>9</sup>

Besides the frame, the data structure of *Packetbuff* buffer stores additional information such as RSSI (of a received frame) or header fields such as sequence number, etc. We take advantage of this buffer to exchange power-related information between layers by introducing two new variables that indicate whether to transmit at a specific power level or not, and the power level to be used to transmit the frame. These variables are examined for handling IE header content after a frame reception or before frame transmission.

### 3) ETX MODULE MODIFICATIONS

ETX has to be initiated and updated for every tuple: mac address and transmission power level. We have modified the default data structure to include a new column in the statistics table for the power transmission level. Other modifications in the ETX module are:

- before calculating ETX for an outgoing packet or updating RSSI for an incoming packet, we obtain the transmission power level of such packet to update the right entry (i.e., mac address and power level) in the table.

<sup>8</sup>Some chipsets like CC2420 perform hardware-sent ACKs using a default transmit power of H. Disabling hardware ACKs causes delay due to software processing and overload under heavy traffic, so we decided to enable the hardware ACKs and keep the radio always high except when it is required.

<sup>9</sup>Files frame802154e-ie.h and frame802154e-ie.c show how to define and use custom IEs.

- freshness and *last\_tx\_time* are now considered for every power level possible.

### 4) RPL-LITE MODIFICATIONS

Changes to RPL-Lite are geared toward controlling the transmission power used for sending DIOs and adapting probing targets to the tuple address and power level.

- Controlling the power level of DIOs. DIOs are sent by the function *rpl\_icmpv6\_dio* from the *icmpv6* module. We have modified this function to add power as a new input parameter. Before sending the DIO, *icmpv6* reads the *packetbuff*'s power-related variables to set the desired power level. Similarly, we have modified the function that sends DIS to use always the default power level (H).
- Sending periodic multicast DIOs. DIO multicast packets are generated periodically according to the Trickle algorithm alternating the power level between H and L each time. For this, we have added a new variable to the DAG information data structure *rpl\_dag\_t* that indicates the desired transmission power level and switches after every transmission.
- Probing (unicast DIOs). Modifications in probing are geared toward considering the tuple mac address and power level as potential target instead of only the mac address. This includes the maintenance of the freshness indicator and the function in charge of selecting the next target to be polled (*get-probing-target*).

### 5) OBJECTIVE FUNCTION

The implementation of METOF is based on MRHOF and can be summarized as follows.

- Link metric implementation. Contiki-NG uses the function *nbr\_link\_metric* to calculate link metric for a neighbor. We have edited this function to calculate the best the link metric for a specific neighbor *i* as:

$$\begin{aligned} nbr\_link\_metric \\ = \min (PTX(H) \cdot ETX(H)^i, PTX(L) \cdot ETX(L)^i) \quad (4) \end{aligned}$$

TABLE 7. Results 25 m × 25 m x-RPL(METOF).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
APP-H	1596.8	1744.5	751.9	1026.2	1759.1	883.0	1323.2	1190.3	1176.7	1325.8	887.8	1166.2	1187.2	1176.4	1029.7
APP-L	1997.5	1851.2	2842.6	2569.2	1835.7	2712.1	2270.9	2405.4	2417.6	2269.1	2708.8	2429.7	2409.1	2419.1	2566.1
Delay(ms)	28.4	27.7	29.0	27.9	28.8	27.9	29.2	27.8	29.3	27.7	28.0	28.2	27.8	28.6	27.4
TOTAL TX	4053.8	4058.4	4049.9	4049.2	4052.4	4049.6	4051.8	4053.4	4050.7	4081.1	4058.6	4049.8	4058.8	4052.6	4050.0
TOTAL RX	691.7	840.3	753.8	823.2	653.6	965.8	719.6	826.6	714.7	851.0	984.2	788.4	1013.8	802.2	858.2
TX_NO_ACK	0.2	0.1	0.1	0.0	0.1	0.1	0.3	0.1	0.1	0.0	0.0	0.0	0.2	0.1	0.2
NUM_RTX	43.7	32.3	24.0	46.2	26.7	21.9	30.9	17.9	37.4	28.7	21.5	25.9	17.0	27.5	43.8
FORWARDED	1.0	1.5	1.3	1.1	1.0	0.8	0.9	1.1	1.0	30.9	4.6	1.1	5.0	1.1	1.0
M-DIO-H	22.6	23.7	22.0	22.4	22.3	23.1	22.1	22.2	22.3	22.4	23.9	22.0	23.7	22.1	22.6
M-DIO-L	20.2	20.0	20.3	20.1	20.1	20.0	20.2	20.1	20.4	20.1	20.2	20.1	20.1	20.1	20.0
U-DIO-H	330.8	335.5	320.4	319.9	331.0	322.6	325.2	332.8	327.8	328.0	329.0	327.8	332.0	339.6	315.9
U-DIO-L	84.9	82.0	91.4	90.3	83.2	88.0	89.3	81.5	84.9	84.8	84.3	82.9	81.7	74.2	94.7
PROCESSED_DIO	686.3	828.3	749.7	811.8	650.5	955.3	717.3	817.3	713.7	812.8	965.6	780.9	995.9	795.9	844.8
PRNT_SWITCH	1.0	1.2	1.1	1.4	1.0	1.4	1.1	1.2	1.0	1.1	1.0	1.1	1.0	1.1	1.3
DIS	1.0	1.5	1.4	1.0	1.0	1.0	1.0	1.1	1.0	1.0	1.2	1.0	1.2	1.1	1.0
JoinedDelay(s)	56.3	45.9	55.0	44.5	54.9	45.3	55.7	46.3	60.3	49.1	34.8	45.2	34.3	46.7	44.7
Energy-CPU	9030.8	9225.2	9024.9	9206.2	8707.6	9296.9	8674.1	9172.6	8686.1	9169.4	8793.1	8938.2	9040.7	8866.8	9316.2
Energy-TX	345.1	352.7	308.5	321.8	351.3	316.2	333.1	328.5	331.1	339.6	321.3	330.3	333.9	332.1	325.3
Energy-RX	5811.7	5809.5	6502.8	6224.6	6118.8	6168.7	6043.9	6117.5	6386.4	6175.5	6063.2	6154.4	6200.8	6116.6	6451.1
Energy-IDLE	48931.8	48931.7	48916.2	48922.5	48924.9	48923.7	48926.7	48924.8	48918.9	48923.3	48925.9	48923.9	48922.8	48924.8	48917.1
Energy-TOTAL	64175.9	64375.7	64809.0	64731.7	64159.2	64762.1	64034.3	64600.0	64379.1	64664.4	64160.1	64403.5	64554.7	64296.9	65066.2

similarly to Algorithm 1. We also store the best power setting (BPTX field of the RPL neighbors table) for that neighbor in this function. This function will return 0xFFFF if the local metric cannot be calculated due to uninitialized values of ETX.

- DAG metric container update. We have modified the *rpl\_metric\_container\_t* data structure to include the best link metric. After receiving a DIO from a neighbor, the metric container is updated for the DODAG instance (*rpl\_instance\_t*) according to the preferred parent.
- transmission power settings. Every time the DAG state is updated with a preferred parent, the global variable that control the transmission power level is updated.

## APPENDIX B

### SIMULATION FOR SCENARIO 25 × 25m

This appendix provides the per-mote simulation results obtained as the average of the 25 simulation runs for the main scenario of 25 m × 25 m.

## REFERENCES

- [1] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, document RFC 6550, RFC Editor, Internet Requests for Comments, Mar. 2012. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc6550.txt>
- [2] J. Ko, A. Terzis, S. Dawson-Haggerty, D. E. Culler, J. W. Hui, and P. Levis, "Connecting low-power and lossy networks to the internet," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 96–101, Apr. 2011.
- [3] O. Iova, P. Picco, T. Istomin, and C. Kiraly, "RPL: The routing standard for the Internet of Things... or is it," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 16–22, Dec. 2016.
- [4] S. A. Alvi, F. U. Hassan, and A. N. Mian, "On the energy efficiency and stability of RPL routing protocol," in *Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2017, pp. 1927–1932.
- [5] O. Gaddour, A. Koubaa, N. Baccour, and M. Abid, "OF-FL: QoS-aware fuzzy logic objective function for the RPL routing protocol," in *Proc. 12th Int. Symp. Model. Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2014, pp. 365–372.
- [6] D. Pancaroglu and S. Sen, "Load balancing for RPL-based Internet of Things: A review," *Ad Hoc Netw.*, vol. 116, May 2021, Art. no. 102491.
- [7] A. Sebastian and S. Sivagurunathan, "A survey on load balancing schemes in RPL based Internet of Things," *Int. J. Sci. Res. Netw. Secur. Commun.*, vol. 6, no. 3, pp. 43–49, 2018.
- [8] O. Iova, F. Theoleyre, and T. Noel, "Exploiting multiple parents in RPL to improve both the network lifetime and its stability," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 610–616.
- [9] L. Zhu, R. Wang, and H. Yang, "Multi-path data distribution mechanism based on RPL for energy consumption and time delay," *Information*, vol. 8, no. 4, p. 124, Oct. 2017.
- [10] K. F. Haque, A. Abdelgawad, V. P. Yanambaka, and K. Yelamarthi, "An energy-efficient and reliable RPL for IoT," in *Proc. IEEE 6th World Forum Internet Things (WF-IoT)*, Jun. 2020, pp. 1–2.
- [11] H. Kharrufa, H. A. A. Al-Kashoash, and A. H. Kemp, "RPL-based routing protocols in IoT applications: A review," *IEEE Sensors J.*, vol. 19, no. 15, pp. 5952–5967, Apr. 2019.
- [12] B. Ghaleb, A. Y. Al-Dubai, E. Ekonomou, A. Alsarhan, Y. Nasser, L. M. Mackenzie, and A. Boukerche, "A survey of limitations and enhancements of the IPv6 routing protocol for low-power and lossy networks: A focus on core operations," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1607–1635, 2nd Quart., 2018.
- [13] S. Tadigotla and J. K. Murthy, "A comprehensive study on RPL challenges," in *Proc. 3rd Int. Conf. Adv. Electron., Comput. Commun. (ICAEECC)*, Dec. 2020, pp. 1–6.
- [14] H. Pereira, G. L. Moritz, R. D. Souza, A. Munaretto, and M. Fonseca, "Increased network lifetime and load balancing based on network interface average power metric for RPL," *IEEE Access*, vol. 8, pp. 48686–48696, 2020.
- [15] S. S. Solapure and H. H. Kenchannavar, "Design and analysis of RPL objective functions using variant routing metrics for IoT applications," *Wireless Netw.*, vol. 26, no. 6, pp. 4637–4656, Aug. 2020.
- [16] S. Capone, R. Brama, N. Accettura, D. Striccoli, and G. Boggia, "An energy efficient and reliable composite metric for RPL organized networks," in *Proc. 12th IEEE Int. Conf. Embedded Ubiquitous Comput.*, Aug. 2014, pp. 178–184.
- [17] N. N. Hadaya and S. A. Alabady, "Improved RPL protocol for low-power and lossy network for IoT environment," *Social Netw. Comput. Sci.*, vol. 2, no. 5, pp. 1–11, Sep. 2021.
- [18] D. Passos and C. V. N. Albuquerque, "A joint approach to routing metrics and rate adaptation in wireless mesh networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 4, pp. 999–1009, Aug. 2012.
- [19] D. G. Narayan and U. Mudanagudi, "A cross-layer framework for joint routing and rate adaptation in infrastructure wireless mesh networks," *Comput. Electr. Eng.*, vol. 56, pp. 113–129, Nov. 2016.
- [20] L. Alkama and L. Bouallouche-Medjkoune, "IEEE 802.15.4 historical evolution versions: A survey," *Computing*, vol. 103, no. 1, pp. 99–131, Jan. 2021.
- [21] Y. Jang, Y. Kim, S. Park, and S. Choi, "Link adaptation strategies for IEEE 802.15.4 WPANs: Protocol design and performance evaluation," *J. Commun. Netw.*, vol. 21, no. 4, pp. 376–384, Aug. 2019.
- [22] R. B. Abdessalem and N. Tabbane, "RPL-SCSP: A network-MAC cross-layer design for wireless sensor networks," in *Proc. 9th Int. Conf. Wireless Commun. Sensor Netw.* New Delhi, India: Springer, 2014, pp. 27–35.

- [23] B. Safaei, A. M. H. Monazzah, and A. Ejlali, "ELITE: An elaborated cross-layer RPL objective function to achieve energy efficiency in Internet-of-Things devices," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1169–1182, Jan. 2021.
- [24] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher, "Real-time power-aware routing in sensor networks," in *Proc. 14th IEEE Int. Workshop Qual. Service*, Jun. 2006, pp. 83–92.
- [25] W. Rukpakavong, I. Phillips, L. Guan, and G. Oikonomou, "RPL router discovery for supporting energy-efficient transmission in single-hop 6LoWPAN," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 5721–5725.
- [26] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, *The Trickle Algorithm*, document RFC 6206, RFC Editor, Internet Requests for Comments, Mar. 2011. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc6206.txt>
- [27] J. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, *Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks*, document RFC 6551, RFC Editor, Internet Requests for Comments, Mar. 2012, [Online]. Available: <http://www.rfc-editor.org/rfc/rfc6551.txt>
- [28] P. Thubert, *Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)*, document RFC 6552, RFC Editor, Internet Requests for Comments, Mar. 2012. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc6552.txt>
- [29] O. Gnawali and P. Levis, *The Minimum Rank With Hysteresis Objective Function*, document RFC 6719, RFC Editor, Internet Requests for Comments, Sep. 2012. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc6719.txt>
- [30] R. Jadhav, R. Sahoo, and Y. Wu. (Nov. 2020). *RPL Observations*. Working Draft, IETF Secretariat, Internet-Draft draft-ietf-roll-rpl-observations-05. [Online]. Available: <http://www.ietf.org/internet-drafts/draft-ietf-roll-rpl-observations-05.txt>
- [31] S. Dawans, S. Duquennoy, and O. Bonaventure, "On link estimation in dense RPL deployments," in *Proc. 37th Annu. IEEE Conf. Local Comput. Netw. Workshops*, Oct. 2012, pp. 952–955.
- [32] A. Kurniawan, *Practical Contiki-NG: Programming for Wireless Sensor Networks*, 1st ed. New York, NY, USA: Apress, 2018.
- [33] A. Parasuram, D. Culler, and R. Katz, "An analysis of the RPL routing standard for low power and lossy networks," *Elect. Eng. Comput. Sci.*, Univ. California Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2016-106, 2016. [Online]. Available: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-106.html>
- [34] H. Kim, J. Ko, D. E. Culler, and J. Paek, "Challenging the IPv6 routing protocol for low-power and lossy networks (RPL): A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2502–2525, 4th Quart., 2017.
- [35] R. K. Yadav and N. Awasthi, "A survey on enhanced RPL: Addressing the mobility in RPL," in *Proc. 4th Int. Conf. I-SMAC (IoT Social, Mobile, Anal. Cloud) (I-SMAC)*, Oct. 2020, pp. 1189–1195.
- [36] B. Safaei, A. Mohammadsalehi, K. T. Khoosani, S. Zarbaf, A. M. H. Monazzah, F. Samie, L. Bauer, J. Henkel, and A. Ejlali, "Impacts of mobility models on RPL-based mobile IoT infrastructures: An evaluative comparison and survey," *IEEE Access*, vol. 8, pp. 167779–167829, 2020.
- [37] J. Kniess and V. Figueiredo Marques, "MARPL: A crosslayer approach for Internet of Things based on neighbor variability for mobility support in RPL," *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 12, Dec. 2020, Art. no. e3931.
- [38] J. Hui, J. Vasseur, D. Culler, and V. Manral, *An IPv6 Routing Header for Source Routes With the Routing Protocol for Low-Power and Lossy Networks (RPL)*, document RFC 6554, RFC Editor, Internet Requests for Comments, Mar. 2012.



**RAFAEL ESTEPA** received the M.S. and Ph.D. degrees in telecommunication engineering from the University of Seville, in 1998 and 2002, respectively. He worked as a Product Engineer with Alcatel, Spain, for two years. He has also been a Visitor with the Department of Applied Mathematics, Instituto Superior Técnico (IST), Lisbon, and Dublin Institute of Technology (DIT). He is currently an Associate Professor with the Department of Telematics Engineering, University of Seville. His research interests include networking, voice over IP (VoIP) quality of service, wireless networks, unmanned aerial vehicles (UAVs), and cybersecurity.



**ANTONIO ESTEPA** received the M.S. and Ph.D. degrees in telecommunication engineering from the University of Seville, in 1998 and 2004, respectively. From 1998 to 2000, he was a software and network engineer with a software development company. In 2004, he was a Visitor with the Department of Electrical Engineering and Computer Science, University of Minnesota, Minneapolis, MN, USA. He is currently an Associate Professor with the Department of Telematics Engineering, University of Seville. He has authored or coauthored several conferences or journal papers. His research interests include telecommunication networks, with particular emphasis in networking protocols, wireless networks, and cybersecurity.



**GERMAN MADINABEITIA** (Member IEEE) received the M.S. and Ph.D. degrees in telecommunication engineering from the Universidad Politécnica de Madrid, in 1986 and 2004, respectively. He worked as a product engineer in the industry, for over ten years. He is currently an Assistant Professor with the Department of Telematics Engineering, University of Seville. His research interests include networking, the Internet of Things, cybersecurity, and traffic engineering.



**ERNESTO GARCÍA** received the B.S. degree in telecommunication engineering from the University of Seville, in 2020. He worked as a Research Assistant with a scholarship in the Department of Telematics Engineering, University of Seville, and developed the early stages of this research. In 2020, he started his career as an Engineer with the Research and Innovation Department, NTT Ltd., Spain. His research interests include networking, the Internet of Things (IoT), and telecommunications.

...