

Evolutionary Software Defined Networking-Inspired Routing Control Strategies for the Internet of Things

GEORGE VIOLETTAS^{ID}, (Member, IEEE), SOPHIA PETRIDOU^{ID}, (Member, IEEE),
AND LEFTERIS MAMATAS^{ID}, (Member, IEEE)

Department of Applied Informatics, University of Macedonia, 54636 Thessaloniki, Greece

Corresponding author: George Violettas (georgevio@uom.edu.gr)

This work was supported in part by the EU's Horizon 2020 Research and Innovation Programme through the fourth open call scheme of the FED4FIRE+ under Grant 732638, and in part by the EU-BRA Horizon 2020 NECOS Project funded by the European Commission and the Brazilian Ministry of Science, Technology, Innovation, and Communication (MCTIC) through RNP and CTIC under Grant 777067.

ABSTRACT The Routing over Low Power and Lossy Networks (RPL) protocol is an important standardized routing solution for the Internet of Things (IoT), characterized by significant benefits, including IPv6 support and efficient low-power operation over noisy channels, especially for many-to-one communication scenarios (i.e., a set of deployed devices gathering periodic measurements destined to a single sink node). However, the RPL faces performance issues in networks with mobility or *point-to-point communication* requirements, raising applicability constraints in a number of real-world IoT applications, spanning from environmental monitoring to industrial and urban networks. In this paper, we approach RPL from the Software-Defined Networking (SDN) perspective, exploiting its high customization features to address the above inefficiencies. We apply an evolutionary methodology, i.e., building over the widely deployed RPL protocol, while maintaining compliance with its standard. More precisely, we investigate two routing control strategies exploiting the global view of the network: (i) the *Moderate RPL control* that enables dynamic reconfigurability of crucial protocol parameters to improve its operation in mobile environments; and (ii) the *Deep RPL control* that utilizes a new RPL Objective Function (OF) we proposed that enforces direct *point-to-point* paths through link-coloring. We implemented and evaluated the two strategies based on a novel centralized routing control facility. Our experimental analysis considers hybrid scenarios with both fixed and mobile nodes, as well as many-to-one and *point-to-point communication*. Compared to the standard RPL in mobile topologies, the proposed solution achieves improved packet delivery ratio of up to 33 percent and 21 percent for the mobile nodes and for the whole network, respectively, while maintaining RPL compliance. In case of *point-to-point communication* in a random topology, the improvements rise up to 32.7 percent for the trip-time and 42 percent for the round-trip time, while the packet loss ratio for the same experiment is improved up to 75 percent in the non-storing mode.

INDEX TERMS Internet of Things, software-defined networks, RPL, objective function, link coloring.

I. INTRODUCTION

Augmenting Wireless Sensor Networks (WSNs) with Internet connectivity, enabled an important class of IoT applications on area, machine or people monitoring, such as for the protection of the environment, industrial processes control, and personalised healthcare systems [1]–[3]. In this context, *efficient routing* is a challenging issue, primarily due to the

The associate editor coordinating the review of this manuscript and approving it for publication was Arun Prakash.

power, storage, memory, processing, and signal limitations of the connected devices [4].

The most important relevant proposal is RPL, which is a distance-vector IPv6 routing protocol for Low-power and Lossy Networks (LLNs). In practice, RPL organizes network nodes, e.g., motes of a WSN, as a Destination-Oriented Directed Acyclic Graph (DODAG) routed to a single destination called sink [5], [6]. The sink is the only node that can launch the DODAG's (re)construction, which is based upon a periodic exchange of routing control messages.

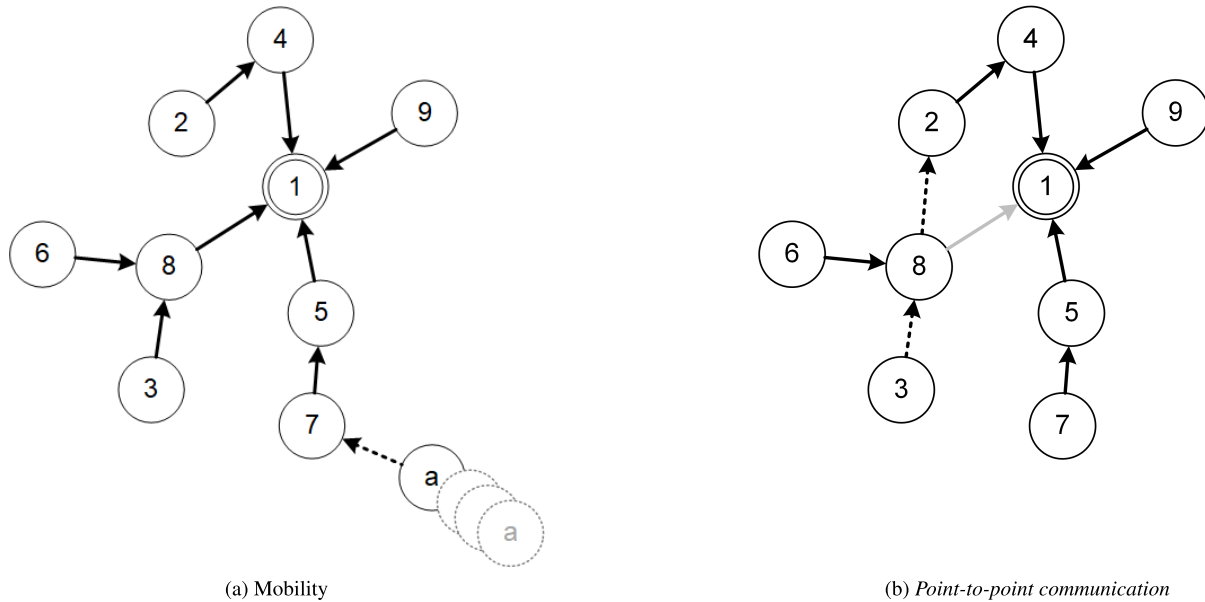


FIGURE 1. An abstract view of the RPL's performance issues.

The DODAG's maintenance is placed at the very core of the RPL's functionality and hence, the following two important aspects impact significantly the network performance: (i) an efficient scheduling of the topology control messages from a dedicated algorithm called the trickle timer where I_{min} and $I_{doubling}$ are the particular RPL configuration parameters which tune the messages' time interval from I_{min} up to $I_{min} * 2^{I_{doubling}}$; and (ii) an appropriate node-parent selection and link quality estimation based on a relevant Objective Function (OF).

A. MOTIVATION

The more complex IoT scenarios and applications become, e.g., involving *mobility* and communication beyond measurement collection from a single node, such as *point-to-point (P2P)*, the more they are questioning the applicability boundaries of the existing protocols' solutions, inherited by the WSNs. For example, the RPL topology maintenance features are particularly adjusted to energy efficiency and long-term periodic sensing over fixed nodes' deployment [7].

In the case of the topology shown in Fig. 1a, the default trickle timer starts from roughly 4 ms (i.e., $I_{min} = 2^{12}$) and gradually increases (actually doubled) defining accordingly the interval time of the topology discovery control messages. This is a typical approach when a fixed topology of nodes (e.g., 2 to 9) gather and send measurements to the sink-node (e.g., node 1). Upon the appearance of a new mobile node (e.g., node *a*), the above "conservative" configuration of gradually increased interval time would lead to extensive delays in discovering the newcomer, since the trickle timer reaches rather high values, e.g., up to 17.5 min (i.e., $2^{12} * 2^8$). On the other hand, flatly decreasing the nodes' trickle timer, e.g., independently of their mobility behavior, can lead to pointless energy consumption. We argue that the

fixed nodes require different treatment regarding RPL configuration compared to the mobile ones. Hence, a centralized control approach defining RPL's parameters per-node can be beneficiary for the network.

Apart from data gathering, the same nodes' deployment could temporarily serve the need for direct communication between two nodes, as shown in Fig. 1b. For example, a node that monitors a threshold crossing value may urgently require to trigger an alarm to a node that is not the typical sink; for instance, we assume that the node 3 has to communicate with the node 2. RPL supports two approaches in implementing *P2P communication*: (i) the storing mode, where each node stores locally the DODAG; and (ii) the non-storing mode, where each node knows its parent and only the sink node maintains the DODAG. In both cases, the DODAG is constructed with the path towards the sink as a target, that is, $3 \rightarrow 8 \rightarrow 1$; paths between nodes are neither stored nor discovered in full. In such a case, forcing an emergency data flow from $3 \rightarrow 2$ simply requires that the node 8 should switch from the "parent" 1 to the new "parent" 2. This "surgical" switching improves significantly the $3 \rightarrow 2$ communication, while it has minor impact on the rest nodes' communications, e.g., in Fig. 1b only the nodes 6 and 8 are affected from the change. We argue that many WSN deployments have temporal requirements for *P2P* apart from the typical many-to-one communication. Such requirements can be served by centralized routing control mechanisms which, exploiting the underlying RPL graph, could proceed with local amendments.

RPL actually supports a number of configuration parameters and OF customization(s) that cover a wide range of alternative deployments [5], [6]. However, customizing such configurations is basically manual, global, and often unpredictable in terms of the outcome. The above two exam-

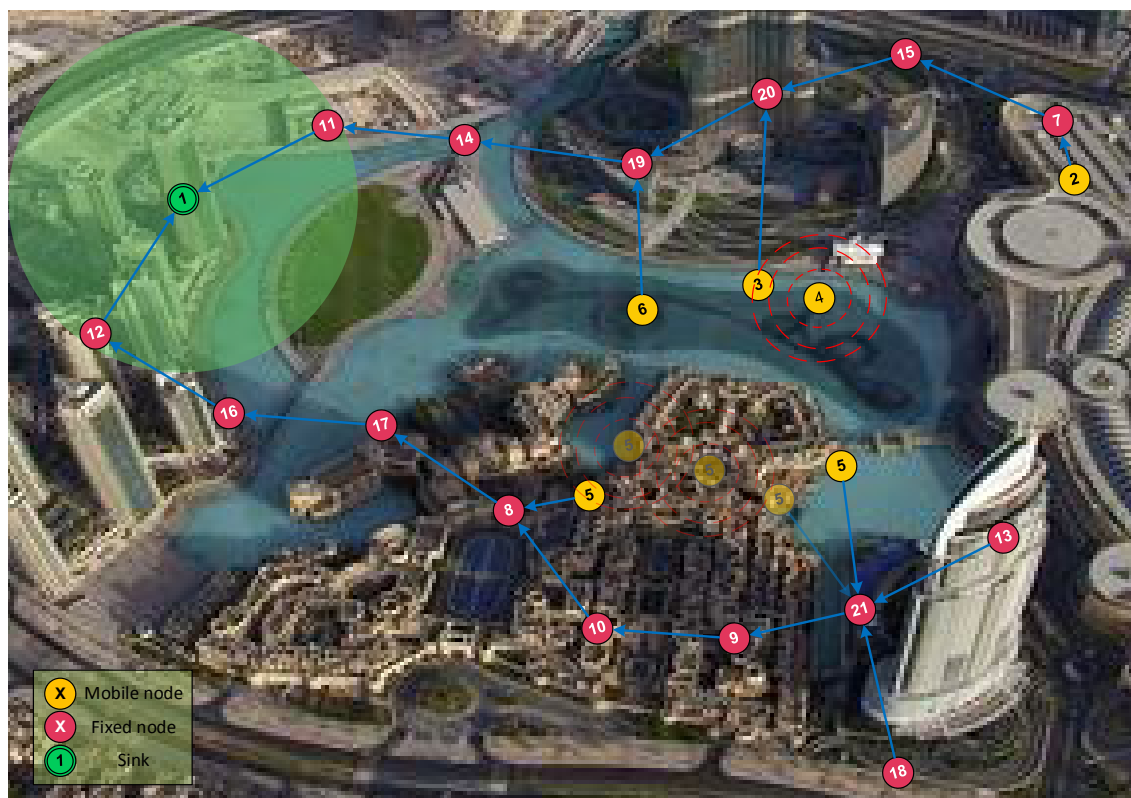


FIGURE 2. Urban environmental monitoring.

ples highlight the need for routing control mechanisms that handle appropriately the requirements defined by the application, i.e., configure network nodes individually, or “surgically” change the topology graph, an approach aligned to the Software-Defined Networking (SDN) paradigm. Such concepts are discussed below.

B. CONTRIBUTION

Here, we suggest that the high customization capability of the RPL protocol can enable the appropriate SDN-inspired routing control strategies to extend the applicability of the protocol beyond the typical WSN scenarios. Hence, we propose two SDN-inspired network control mechanisms improving the RPL’s behavior in mobile and *P2P communication* contexts, reflecting different depths in the protocol adaptation:

- The *Moderate RPL control* that enforces appropriate protocol configurations in an on-the-fly manner to improve mobile communication. For example, it allows mixing RPL configurations, i.e., treating differently mobile and fixed nodes. In a typical WSN data collection application, where battery-powered mobile nodes may be employed to widen the monitoring area, fixed nodes should broadcast topology discovery messages frequently to provide connectivity chances for the mobile ones, while the latter relax their discovery intervals to save energy.

- The *Deep RPL control* that establishes *P2P communication* in consistence to the RPL topology graphs, exploiting the feature of changing the OF. More specifically, it utilizes the idea of link coloring to appropriately color the network nodes, and enforce direct communication paths based on those. When a measurement of a sensor-node crosses a threshold it may trigger a rule, in an alert and action application, that defines an urgent message transmission to a particular recipient-node instead of the regular post to the sink-node. A centralized controller can specify the relevant direct path that can be enforced by a suitable OF.

Our proposed mechanisms do not require adaptations in the RPL standard and are being integrated into CORAL [8], [9], our novel centralized control facility supporting: (i) network control mechanism extensibility; (ii) abstracted protocol configuration APIs; and (iii) GUI interface and experimentation features for protocol configuration and measurement visualization. We implemented CORAL based on the WiSHFUL architecture [10], offering suitable abstractions and interfaces for dynamic protocol adaptations. An initial investigation of CORAL and the *Moderate RPL control* strategy can be found in [9], while a demonstration of this early work is described in [8]. Our experimental results in the current paper confirm that extending the RPL with novel SDN-inspired routing control features can tackle its inefficient performance in the cases of heterogeneous topologies consisting of both



FIGURE 3. The harsh workplace of a mine.

fixed and mobile nodes, as well as applications occasionally requiring *P2P communication*.

To highlight the impact of our proposal, we discuss two representative use-case scenarios in Section II. The architecture of the centralized routing control facility along with the two proposed mechanisms are described in Section III. Section IV presents a detailed experimental analysis. The related works are discussed in Section V. Finally, we provide concluding remarks and future work insights in Section VI.

II. USE-CASE SCENARIOS

To further motivate our work, we elaborate on two use-cases with characteristics aligned to the capabilities of the proposed SDN-like routing control schemes: (i) an *urban environmental monitoring* deployment based on data collection; and (ii) a *harsh workplace* scenario with occasional alert and action communication requirements.

A. DATA COLLECTION IN URBAN ENVIRONMENTS

In the first use-case, we consider an urban environment with a number of motes scattered downtown gathering measurements (e.g., pollution or weather-related). We assume that

fixed nodes connected to a mains power supply coexist with battery-powered mobile ones, for more complete area coverage. Fig. 2 shows that the yellow-colored mobile nodes communicate through the red-colored fixed nodes, given that there is radio connectivity among them.

According to our previous work [9], in cases like this, an optimal routing control mechanism should take into account the different resource-capabilities of the fixed and mobile nodes. For example, RPL should adapt the fixed nodes to probe more frequently for mobile ones into their vicinity, while the latter should be more conservative to preserve energy. Practically, a platform like CORAL should be able to tune, from a global viewpoint, the RPL routing configuration in respect to the nodes' mobility profile. This will allow an extended network coverage since the mobile nodes can be detected and incorporated in the monitoring network without delays.

B. ALERTS AND ACTIONS IN HARSH WORKPLACES

In the second use-case, we consider a harsh workplace environment like the mine shown in Fig. 3. A set of nodes

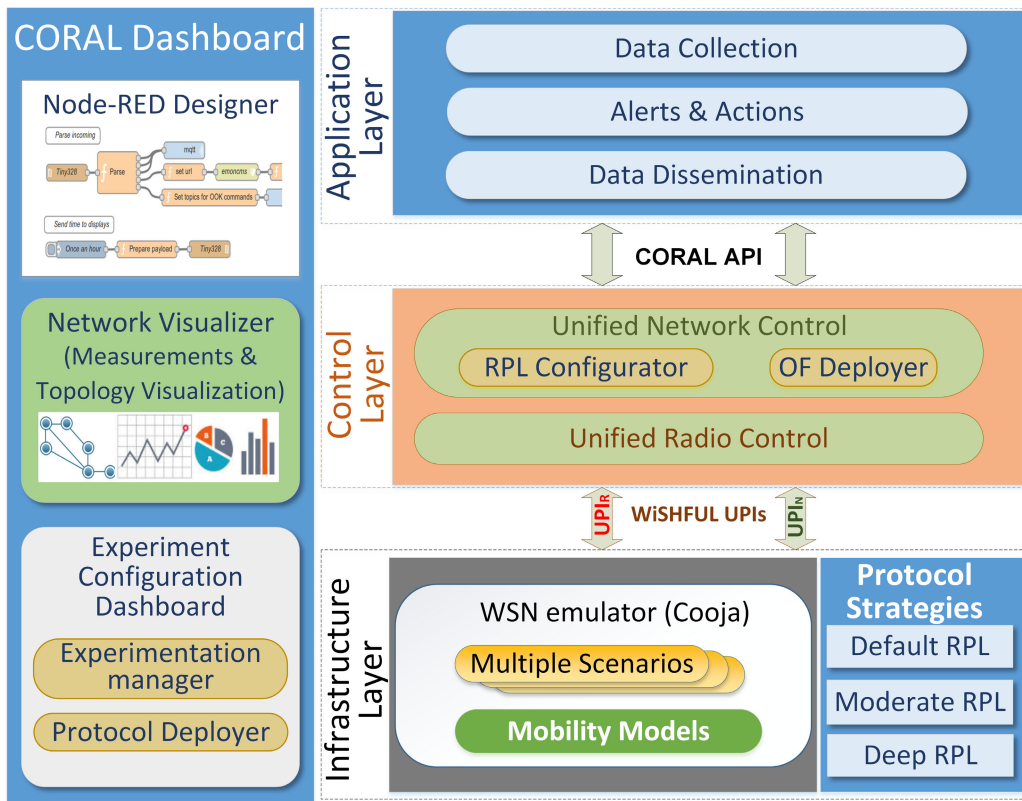


FIGURE 4. The CORAL architecture.

are deployed in the area, such that the fixed ones monitor the environment (e.g., oxygen level) while the mobile wearables collect the miners' vital signs. Under typical operation, this deployment exhibits similar requirements with the first use-case.

However, if a miner suddenly faces an emergency, indicated by his vital signs monitoring device, there is a need for an emergent communication between him and the operation center on the ground surface, or alternatively to another worker with medical training. In such a case, high priority should be given to the messages originated from the miner in danger, even sacrificing the rest devices' communication. As identified in a number of works [11], [12] and shown in our experimental analysis below, RPL is inefficient for the direct communication between two end-points. Relevant scenarios have been described in [13]–[15], focusing on industrial cases of emergency, a large-scale industrial environment, and an oil refinery scenario with real-time constraints, respectively.

Here, we argue that a centralized routing control facility like CORAL can enforce direct communication between nodes while maintaining the typical RPL operation for the rest network. To tackle this issue, our approach employs a new OF that enables link coloring to enforce direct *P2P* paths. The centralized controller appropriately colors the nodes to intentionally change a single or more nodes' parents, that is to enable the direct communication while being aligned to the RPL's RFC [5].

III. CENTRALIZED ROUTING CONTROL

To improve the performance of RPL in mobile or *P2P communication* contexts, we propose CORAL, a novel routing control facility accommodating two alternative centralized control approaches, i.e., the *Moderate* and *Deep RPL control*. Here, we provide a high-level overview of the CORAL architecture and describe the novel routing control strategies we propose.

A. THE CORAL ARCHITECTURE

The CORAL facility implements SDN-inspired routing control to evolutionary enable the applicability of RPL to alternative IoT use-cases, such as those described in Section II. Fig. 4 illustrates the CORAL architecture, which follows the typical three-tier SDN paradigm. We describe the three layers in a bottom-up fashion below:

- The *Infrastructure Layer* accommodates multiple scenarios with diverse topologies and network settings, including support of realistic mobility models for the nodes. Such scenario configuration can be both loaded at compile-time or dynamically updated at real-time, i.e., to implement dynamic IoT scenarios. In practice, we employ Ansible scripting for the offline Cooja emulator configurations and utilize the *Control Layer* for the on-line adaptations.
- The *Control Layer* offers abstracted and logically-centralized control of the network. The network control

abstraction accommodates two main components: (i) the *RPL Configurator* being responsible to configure the RPL protocol (e.g., adapting parameters like the I_{min}) and running OF specific features (e.g., the link coloring in our case) to adhere with rapidly changing network demands; and (ii) the *OF Deployer* able to change the OF in real-time, when alternative communication requirements emerge. This layer currently supports protocol deployment via device-specific Ansible scripts updating the IoT devices' firmware (i.e., low memory-footprint approach with moderate deployment time). Our plans also include experimentation with over-the-air protocol deployment approaches. The configuration / measurements of network protocol functionalities and radio are provided to the *Control Layer* through the *Universal Network and Radio Control Interfaces*, i.e., the UPI_N , for the RPL protocol, and UPI_R for the radio channel, developed by the WiSHFUL project [16]. UPIs are also utilized by the *Infrastructure Layer* in order to provide via the serial port or CoAP protocol to the *Control Layer* real-time measurements on the protocols' and network performance (e.g., the round-trip time).

- The *Application Layer* offers high-level protocol configuration that matches different application types (i.e., for data collection, alerts & actions and data dissemination). The applications express their demands (e.g., the need for P2P communication, or to support mobility) to the *Control Layer*, while the latter matches them with particular protocol configurations, i.e., employing either *Moderate* or *Deep RPL control*. Such matching is currently hard-coded, but we consider relevant intelligent algorithms as future work. The applications' requirements are being communicated to the *Control Layer* over the *CORAL API* through JSON messages.

The control facility interacts with the user through the *CORAL Dashboard*, a flexible and modular Graphical User Interface (GUI), implemented in Node-RED (<http://nodered.org>) and depicted in Fig. 5. We can either configure RPL parameters, such as the I_{min} and $I_{doubling}$, or the OF to rely upon, along with setting the network nodes appropriate coloring. Node-RED flows are wired with JSON messages which pass updated RPL's configuration to the *Control Layer*. Live visualization of the outcome illustrates the experiments' progress and the impact of particular SDN-inspired protocol strategies on the application and network performance.

An initial version of CORAL¹ focusing on the *Moderate RPL control* approach can be found at [8], [9]. We now elaborate on the proposed routing control strategies.

B. MODERATE RPL CONTROL

The first approach to SDN-inspired network control of RPL focuses on its dynamic protocol configuration based either

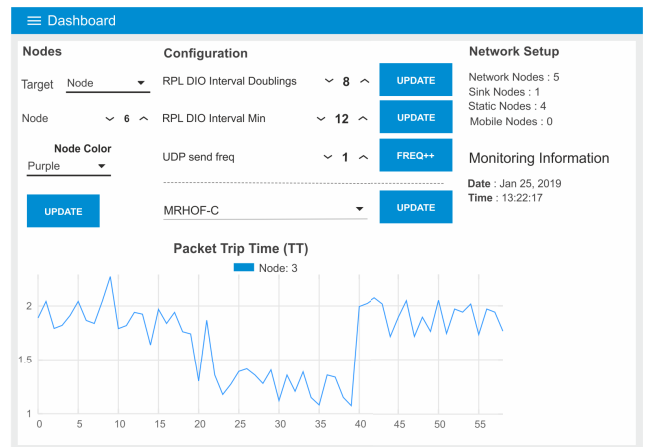


FIGURE 5. On-the-fly RPL configuration with the CORAL dashboard.

TABLE 1. The DODAG's setup time for different network settings as a function of the RPL's I_{min} parameter.

#	No. of nodes	Heterog.	Topology	I_{min}	Setup time (sec)
1	10	Y	Fig 1a	8	11.4
				12	42.3
2	15	N	chain	8	6.2
				12	50.9
3	15	N	lambda (Λ) - sink on top	8	4.8
				12	26.8
4	30	N	as in [7]	8	5.1
				12	23.0
5	30	N	chain	8	11.3
				12	107.4
6	50	N	random	8	10.2
				12	27.3
7	100	N	random	8	32.4
				12	68.1

on environmental conditions or node heterogeneity (e.g., whether they are mobile or not). This network control scheme considers the RPL as a black-box, it does not change its main mechanisms, but it exploits the available protocol configuration options to adapt it to particular conditions, especially towards supporting mobility. This strategy is in line with the requirements of the urban environment scenario described in Section II-A.

To further detail the *Moderate RPL control* approach, we now present how the RPL constructs the network topology. The sink launches the DODAG's construction based on the exchange of routing control messages, i.e., DODAG Information Object (DIO), Destination Advertisement Object (DAO) and DODAG Information Solicitation (DIS). A first DIO message is issued by the sink, and then plenty of them are sent in multicast by nodes getting connected to the graph. The DAO messages are used by all nodes, except the sink, to propagate reverse route information. Finally, DIS messages are sent by disconnected nodes in order to solicit DIO messages from their connected neighbors and join the graph.

The DODAG's maintenance is placed at the very core of the RPL's functionality and hence, a dedicated algorithm—the trickle timer—synchronizes the propagation of DIO messages

¹The relevant source code and video can be found at <https://github.com/SWNRG/wishful-coral>.

upon which the graph's convergence time is based. The critical aspect in DIO multicasting process is to achieve a short period of the graph's setup time and thus, to reinforce network's metrics, e.g., the packet delivery ratio, while restricting the control overhead towards lowering node's power consumption [7]. To achieve the aforementioned trade-off, the DIO messages are sent periodically, but their interval ranges from I_{min} up to I_{max} , where $I_{max} = I_{min} * 2^{I_{doubling}}$. For example, the default RPL configuration specifies $I_{min} = 2^{12} = 4,096 \text{ ms}$ and $I_{doubling} = 8$ which entails $I_{max} = 2^{12+8} = 17.5 \text{ min}$. Actually, the timer's duration is doubled each time it fires. Any change in the DODAG, e.g., an unreachable parent or a new parent selection, resets the trickle timer to I_{min} .

Table 1 reports the impact of the RPL's I_{min} parameter on the graph's setup time for different network settings, i.e., the number of nodes, heterogeneity in nodes' behavior (fixed and/or mobile) and topology type. For example, we consider the topology of Fig 1a with 10 randomly positioned nodes, including both fixed and mobile ones. In the case of an accident close to node 7 at the 20 sec (e.g., an isolated miner loses his senses), the default RPL configuration fails to route its signal since the node is not connected yet to the graph. We notice that it is important to begin with an "aggressive" graph setup policy to ensure that all nodes are being connected to the graph (e.g., within less than 12 sec), and successfully report their data even at the cost of control overhead. Actually, this cost at the given time is not an issue for the successful data delivery, since data cannot be collected before the routing graph is fully constructed.

An important aspect hindering the applicability of RPL in challenging IoT use-cases is the following: the topology probe interval gradually increases and produces a delayed response to the topology changes caused by mobility [17]. This is attributed to the inherent focus of the RPL design on static networks with limited local adaptability [18], since the RPL specifications do not cover when and how DIS messages should be sent [19]. We argue here that RPL has the potential to improve its behavior in mobile environments by adjusting its main configuration parameters or mechanisms.

In the case of *Moderate RPL control*, a centralized control facility provides the options of dynamic and individual nodes' configuration. More precisely, we start with a minimum I_{min} to set up the DODAG as quickly as possible and then continue with low I_{min} values for the fixed nodes and high for the mobile ones, i.e., to alleviate the control overhead. To save precious time, the CORAL gives the option to enforce such strategies on-the-fly. Furthermore, live network monitoring enables early detection of abnormal or inefficient routing behavior, new configuration decisions, and their dynamic/individual enforcement. Relevant strategies can be supported from the CORAL facility in a straightforward manner, involving all the RPL configuration parameters (e.g., I_{min} , $I_{doubling}$, or OF parameters described afterward).

More details on the *Moderate RPL control* can be found in our conference paper [9]. Next, we introduce the *Deep RPL control* mechanism.

C. DEEP RPL CONTROL

In this second SDN-inspired routing control approach, the controller is more deeply involved in the protocol operation, but again in consistence with the RPL standard. The *Deep RPL control* does not handle the RPL as a black-box, in the sense that it goes beyond its parameters' configuration. In practice, it enables the ability to change the OF upon which the DODAG is constructed. The *OF Deployer* changes the OF either pro-actively or re-actively, in response to communication requirements, and *RPL configurator* proceeds with appropriate customization in line with the OF deployed. The goal of this control mechanism is to improve the *P2P communication* of the RPL protocol in correspondence to the requirements of the mine scenario described in Section II-B.

Before further detailing our proposal, we discuss the most common OFs used by the RPL and the way that the protocol handles the *P2P communication*. The RPL is a link-state routing protocol and its topology (mapped to a DODAG) is constructed by the independent decisions of each node regarding their best parent among potential candidates, based on criteria defined in the particular OF used.

The RPL mainly uses the OF Zero (OF0) [20] and the Minimum Rank with Hysteresis OF (MRHOF) [21] (which can be found implemented in IoT environments, such as the Contiki OS [22]). The OF0 returns an output (set of selected parents) that optimizes (i.e., minimize) the number of hops to the sink node. It provides the network topology quickly consuming the minimum resources, although the solution is non-optimal in respect to more sophisticated criteria. For example, the MRHOF selects node parents optimizing (i.e., minimizing) the Expected Transmission Count (*ETX*), which is the expected number of transmissions from a node to a destination to successfully deliver a packet [5]. The default formula is $ETX = 1/(Df * Dr)$, where Df is the probability for a packet to be received by a particular neighbor, and Dr is the probability that the acknowledgment packet is received successfully. The MRHOF allows the addition of new metrics and uses the hysteresis mechanism [21] to avoid frequent switching between parents due to minor metric changes. All parents' selections are being communicated up to the tree topology until the sink node, which is though aware of the full network graph.

This setup facilitates the many-to-one communication since each node passes the packets to its parent until they reach to the sink. However, parent information is not enough for the communication towards a non-sink node, since this may require communication down the routing graph. To address this issue, the RPL follows two alternative approaches: the *storing* and the *non-storing mode*. In the *storing mode*, each node maintains the portion of the DODAG starting from its-own rank and towards the sink, whereas in the *non-storing mode*, only the sink holds the full

Algorithm 1 Parent Selection Towards Establishing a *P2P* Path**Result:** Establish a *P2P* path in RPL**Input :** Color *node_color*, candidate parent *p1*, candidate parent *p2***Output:** Selected parent (*p1* or *p2*) for a node colored *node_color*

```

1 begin
2   //if a receiver is found by a red
  node, make it a parent
3   if p1.node_color==orange then
4     | if this.node_color==red then
5     |   return p1;
6   if p2.node_color==orange then
7     | if this.node_color==red then
8     |   return p2;
9   switch this.node_color do
10    case orange
11    |   //receiver chooses parent
     |   based on the ETX
12    |   return p1.ETX< p2.ETX? p1:p2;
13    end
14    case purple
15    |   //sender chooses a red parent
     |   if p1.node_color==red then
16    |     | if p2.node_color==red then
17    |     |   return p1.ETX< p2.ETX? p1:p2;
18    |   end
19    case red
20    |   //red nodes choose other red
     |   parents
21    |   if p1.node_color==red then
22    |     | if p2.node_color==red then
23    |     |   return p1.IP< p2.IP? p1:p2;
24    |     | else
25    |     |   return p1;
26    |     | end
27    |   else
28    |     | if p2.node_color==red then
29    |     |   return p2;
30    |     | else
31    |     |   return p1.ETX< p2.ETX? p1:p2;
32    |     | end
33    |   end
34    end
35    case white
36    |   //white nodes are choosing
     |   parents based on the ETX
37    |   return p1.ETX< p2.ETX? p1:p2;
38    end
39  endsw
40 end

```

topology information. These approaches tune the involved performance trade-offs differently: the *storing mode* trades memory state for lower communication delays, while the *non-storing* adopts the opposite strategy. However, both of them are inefficient concerning the need for a potential *P2P* path.

To enable such an option, we added a new mechanism in the CORAL controller, as part of the *RPL Configurator*, which calculates the direct path between any two nodes: the controller is running a second version of the DODAG, where the receiver acts as the sink, and stores the path created between the sender and the receiver. This path is now “implanted” into the existing DODAG with the least possible impact (performance-wise) on the rest network nodes’ communication.

To enforce the desired *P2P* path in real-time, we utilize a new OF, namely the MRHOF-C (coloring). This OF exploits the Link Coloring (LC) feature of the RPL that can be implemented either as a metric or constraint, i.e., to attract or avoid specific links [23]. In practice, the *RPL Configurator*, which is part of the CORAL controller, is utilizing the UPIs to set the nodes’ color, i.e., manipulating the LC, along the desired *P2P* path. It utilizes four different sets of nodes/colors: sender/purple, receiver/orange, nodes-in-path/red, and nodes-non-in-path/white. Once the coloring phase is completed, the controller refreshes the topology in real-time either globally (i.e., a global-repair takes place when the whole network resets the DODAG) or locally (i.e., local-repair enables a reset mechanism for a node and its sub-DODAG only). Global or local-repairs in the topology are being handled by the Algorithm 1 which is executed in each node; some nodes should alter their parent selection to facilitate the establishment of the desired *P2P* path. The algorithm takes as input a node along with its color, and two candidate parents (i.e., *p1* and *p2*), and outputs the preferred parent according to the proposed MRHOF-C OF; in particular, it forces the purple and red nodes to select a red parent (red nodes lie along the path), if there is one available (lines 14 – 19, 20 – 35), while it lets the orange and white nodes to go through the default *ETX* metric selection (lines 10 – 13, 36 – 39). The orange (receiver) node is prioritized as a parent only by a red node (lines 3 – 5, 6 – 8), hence white nodes are not changing parents, minimizing the disturbance in the network. In case that multiple candidate parents exist, the Algorithm 1 is repeatedly executed by the RPL until the best among them is selected.

This centralized routing control interferes in the topology graph with few local adjustments: it improves the performance of a particular *P2P communication* changing the parents of a small subset of nodes. In addition, the performance overhead for the many-to-one communication of the rest nodes is minor, since the new control mechanism is also based on a topology derived by a DODAG. Section V contrast our approach to other relevant works handling the *P2P communication* when the RPL protocol is used.

1) IMPLEMENTATION DETAILS

To realize the *Deep RPL control*, our proposal enables the dynamic deployment of alternative OF(s) (*OF Deployer* component), along with the run-time configuration of such OFs by the controller mechanisms (*RPL Configurator*). The parameters of each OF are globally accessible through the UPIs. The dynamic OF deployment is an existing RPL protocol feature. In particular, the RFCs 6550 [5] and 6551 [23] state the need for RPL improvements and adaptations to certain conditions, although such feature was not yet part of a centralized control facility. Some proposals [17], [19] proceed with OF changes but only in compile-time and not dynamically. Our implementation exercise takes place in a Contiki-based experimentation environment. For the LC feature, we extended the DAG Metric Container (MC), defined in [5] and [23], as the way that nodes use to report alternative or additional metrics along the DODAG. The LC parameter expressed as a 10-bit link constraint (assigned the value 8 by IANA) can be either adjusted statically or dynamically [23]. A new *node_color* parameter has been added in each node to be manipulated by the CORAL centralized controller through the UPIs. Such parameter is then propagated through the LC in the corresponding MC. To avoid loops, the nodes' id (reflected to the last number of the IP address) is decreasing from the sender to the receiver, so the Algorithm 1 chooses the smallest id if two red parents are found.

In cases where the MRHOF-C is employed re-actively, the CORAL platform triggers the global and local topology repair mechanisms, described in [5] and [7]: each node nullifies all its connections and starts again sending solicitation messages looking for potential parents; nodes in the vicinity (i.e., within radio coverage) are evaluated through the OF as potential parents. We noticed that if we proceed with a global-repair, it takes too long for the sender node to recalculate routes with the new OF, a phenomenon well connected with the frequency of sending solicitation messages (DIS), as described in Section III-B. A direct result of triggering such repairs is an anticipated increase of the control overhead momentarily.

In cases where the MRHOF-C is employed pro-actively, it can be initially used with no coloring customization (actually simulating the MRHOF). Once the need for *P2P* communication emerges, the *RPL Configurator* sets the nodes' colors, the Algorithm 1 outputs the new selected parents and the desired path is established. This way, only local-repairs are needed, hence converging time is smaller.

A proof-of-concept experiment, depicted in Fig. 6, is utilizing RPL and the default OF, i.e., MRHOF, to create the network topology shown in Fig. 6a. The packet delivery time (PDT) for each node is depicted in Fig. 6c. At 20 min the OF changes to the MRHOF-C as a means to improve $3 \rightarrow 2$ communication. A new DODAG with the forced path between the sender-node 3 and the receiver-node 2 is created and depicted in Fig. 6b. The delivery time for all nodes against the sink remains unchanged, except that of node 3, which is slightly increasing, since packets have to travel the

longer path $3 \rightarrow 2 \rightarrow 5 \rightarrow 1$; this increase is depicted by the green line in Fig. 6c during the period [20, 40] min of OF change. On the contrary, the direct $3 \rightarrow 2$ path improves the delivery time for these nodes' communication as presented in Fig. 6d. For the last 20 min of the experiment, the MRHOF is reinstated, reverting the network to its initial behavior (Figs. 6c, 6d). The semantic of this experiment is twofold: firstly, our controller and its components, i.e., *OF Deployer* and *RPL Configurator*, can offer a point-to-point communication keeping consistency with the RPL standard; secondly, our solution is beneficial for the desired communication without causing delays on the rest network.

IV. PERFORMANCE EVALUATION

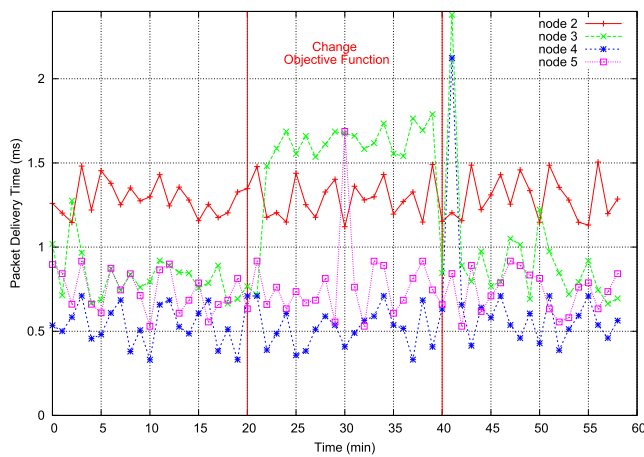
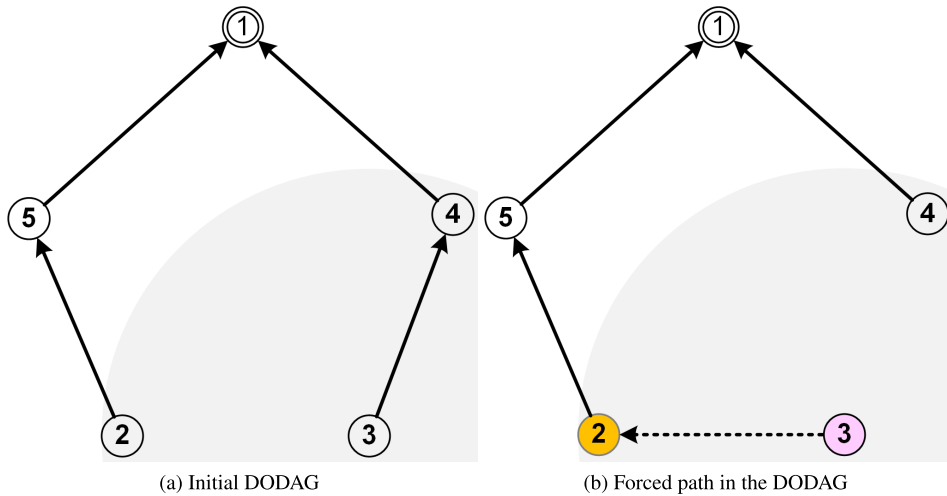
In this section we evaluate the proposed routing control strategies, i.e., the *Moderate RPL* and the *Deep RPL control* against handling *mobility* and *P2P communication*, through a number of experimental setups. The CORAL platform acts as an enabler for the tough experimental task since it accommodates them and facilitates through the dashboard their deployment, configuration, execution, and real-time monitoring.

A. EXPERIMENTAL METHODOLOGY

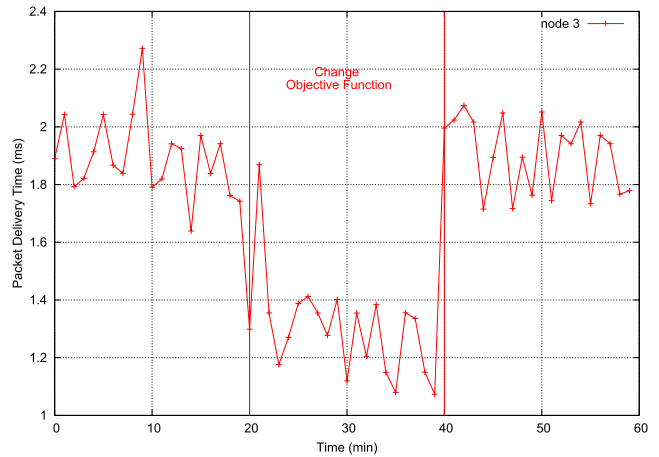
More precisely, for the *mobility* issue we employ the *Moderate RPL* using real mobility traces derived by the MONROE H2020 EU project [24], which provides open access, flexible hardware and software platform for extracting measurements and carrying out custom experimentation on Mobile Broadband (MBB) networks across Europe. As such, the MONROE database includes vehicles' movement trace data (i.e., moving buses, trains, and tracks) from many European cities. We extracted real mobility traces from Stockholm buses, transformed the nodes' GPS coordinates to a $150\text{ m} \times 150\text{ m}$ canvas in Cooja simulator and removed the idle times. Our experiments involve 5 mobile and 16 fixed nodes (including the sink) and last 60 min.

Through the CORAL GUI we employ the *Moderate RPL* mechanisms to extract results regarding two metrics: the packet delivery ratio (*PDR*) defined as the received UDP packets (*rUDP*) over the total number of UDPs being send (*sUDP*), i.e., $PDR = rUDP/sUDP$; and, the control overhead (*OH*) which expresses the ratio of the control packets (*CP*) to the total packets in the network, i.e., $OH = CP/(CP + sUDP)$. All scenarios are using the same deterministic mobility model, so there is no need to confirm the results' statistical accuracy. Cooja TX/RX parameter (i.e., the rate of successfully Transmitting/Receiving a radio message) was set to 100 percent to eliminate randomness, and Transmission/Interference ranges were both set to 50 m according to [19]. All experimental setup parameters are depicted in Table 2. Our comparative analysis for the *Moderate RPL control* uses the standard RPL (indicated as "Default") as a baseline case.

To handle *P2P communication* we exploit the *Deep RPL control* and three distinct network illustrations, namely the Lambda (Λ), Neutral, and Random topologies. The first



(c) PDT per node after path enforcement



(d) PDT in 3 – 2 P2P communication

FIGURE 6. Proof-of-concept experiment.

TABLE 2. Experimental setup.

Layer Setting	Description	Notes
Transport	UDP	
Network	IPv6 / RPL	
Adaptation	6LoWPAN	
MAC	CSMA	
Physical	IEEE 802.15.4	
Cooja ver	3.0	GitHub master branch
TX / RX	100%	TX / INT Range same according to [19]
Transmission Range	50/50 m	
UDP size	60 B	
Node traffic load	60 UDP /h	
Mobility Model	real traces	MONROE Project [24]

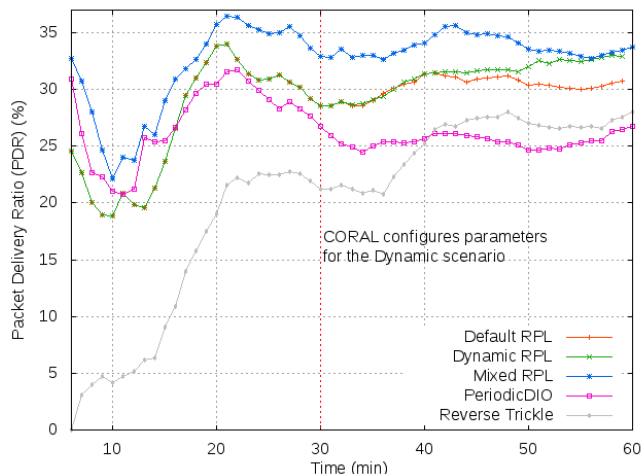
one is positively biased towards highlighting the advantage of forcing a routing path between two specific end-points; the second represents a deployment where our control mechanism does not significantly contribute since the existing paths can serve the desired P2P Communication; the last one depicts a totally random deployment to better simulate a real-world case.

The results in those cases regard two metrics. The packet delivery time (PDT) which expresses the time that a UDP

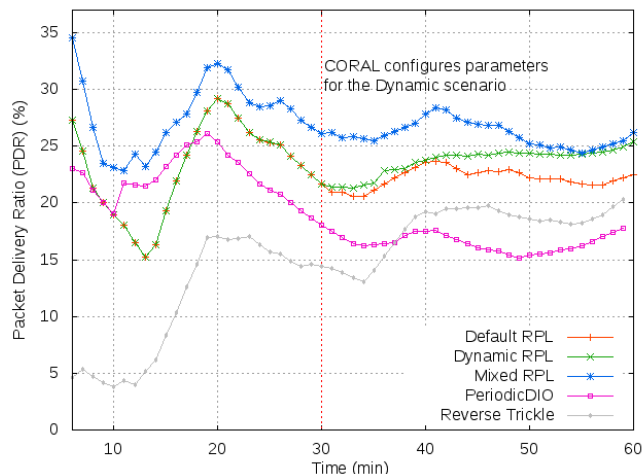
packet requires to travel between the two end-points of a P2P defined path. The PDT is further distinguished to the trip time (TT), if the one-way communication is considered, and to the round-trip time (RTT) if both the forth and back routes are taken into account. The packet loss ratio (PLR) is defined as the ratio of packets lost over the total number of packets send, i.e., $PLR = (sUDP - rUDP)/sUDP$. The experimental analysis for the Deep RPL control contrasts the proposed OF MRHOF-C to the MRHOF which is usually the default choice for the RPL protocol. Below, we present separately the detailed analysis for each control mechanism.

B. MODERATE RPL CONTROL RESULTS

Although we experimented with different combinations of RPL parameters to evaluate the Moderate RPL control, here we focus on the I_{min} configuration since it is the most important RPL parameter for the context under study. Once the I_{min} value changes, the trickle timer is reset. We proceed with two different modification approaches, namely the Mixed (blue line) and Dynamic (green line), both compared to the Default RPL (orange line) in Fig. 7.



(a) The average PDR of all nodes as a function of time



(b) The average PDR of the mobile nodes' subset as a function of time

FIGURE 7. Cooja simulation with real mobility traces.

In the *Mixed* configuration our platform configures the sink and the fixed nodes differently from the mobile ones at the very beginning of the experiment. The first have $I_{min} = 8$ and the latter are configured with $I_{min} = 12$ for the whole period of the experiment, i.e., 60 min (x-axis in Fig. 7a, 7b and 9). This treatment contributes to a fast DODAG setup, while preserves energy for the mobile nodes which send control message with a lower frequency than the fixed ones. In the case of the *Dynamic* setup, the CORAL platform can inject configurations on-the-fly; thus the protocol starts with the default RPL parameters and after 30 min, the I_{min} parameter is dynamically changed to the value 8 only for the sink and fixed nodes. This value entails a higher frequency of the control messages, which in turn provides a higher probability for the mobile nodes' connectivity (compared to $I_{min} = 12$).

Besides the comparison with the Default RPL, our *Moderate RPL control* is also compared with two representative related works [17], [19]. In [17] the authors flatly configure the I_{min} to a fixed value and, thus, completely disable the trickle timer mechanism that proportionally increases the DIO interval time. Their *PeriodicDIO* approach deviates from the RFC specification [5], in that DIO messages are not sent subject to the expiry of the trickle timer, and facilitates the adoption of an aggressive policy upon new parent selection. In our results, the *PeriodicDIO* approach is depicted with the pink line (Fig. 7) assuming that $I_{min} = 12$, since this is the default RPL configuration and one of the settings used by the authors in [17]. The *Reverse Trickle* [19] introduces the reverse rational in the timer mechanism, i.e., reduces to half instead of doubling the DIO interval time, assuming modifications in the DAO control messages which entails non-compliance with the RPL standard. It begins with $I_{min} = 20$ and decreases it down to $I_{min} = 8$ each time a new DIO is sent due to an RPL incident. The idea is that once a mobile node connects to a new parent, it is likely that it remains connected to this parent for a long time. Then, over time, the node is

more likely to move outside the parent's coverage. The gray line in Fig. 7 depicts the performance of the *Reverse Trickle* approach.

Fig. 7a and 7b demonstrate the performance advantages of the *Mixed configuration* in terms of PDR, the former for all network nodes, and the latter for the mobile ones exclusively. The *Moderate RPL* outperforms the default routing protocol regarding the PDR, especially in the case of the mobile nodes. More specifically, it shows an improvement of up to 21 percent for the whole network (Fig. 7a), which rises up to 33.3 percent for the mobile nodes (Fig. 7b). It is also superior to both the *PeriodicDIO* and *Reverse Trickle* as much as 30 percent and 10 percent, respectively, when all network nodes are considered (Fig. 7a), and as much as 18 and 11 percent, respectively, when only mobile nodes are taken into account (Fig. 7b). The *Reverse Trickle* begins with very low performance due the I_{min} configuration at a maximum value, and then it converges with the other solutions, offering nevertheless lower PDR compared to the proposed strategy. The *PeriodicDIO* exhibits stable and comparable to other solutions performance, but it also provides lower PDR since its control mechanism is not aware of the network environment. This is even more clear in Fig. 7b and 8 where we observe that the flat consideration in control messages' interval time is not beneficial for mobility.

Furthermore, in both figures, the performance of the *Dynamic configuration* of RPL is identical with the *Default* case (something that works as a proof-of-concept for our experimentation) and starts to converge with the performance of the *Mixed* configuration after the 30 min, i.e., once the configuration for the fixed nodes has been modified. Since the mobility pattern for the nodes 2 – 6 is an emulation of real moving buses, there are quite long periods of time that the mobile nodes have no connectivity, because of radio limitations. This explains the fact that the average PDR does not exceed 30 percent in Fig. 7b and 8 for the mobile nodes.

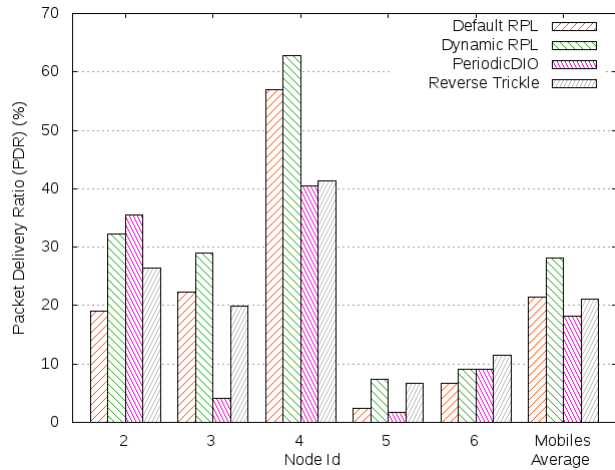


FIGURE 8. PDR per mobile node (2..6) in contrast to the total average values.

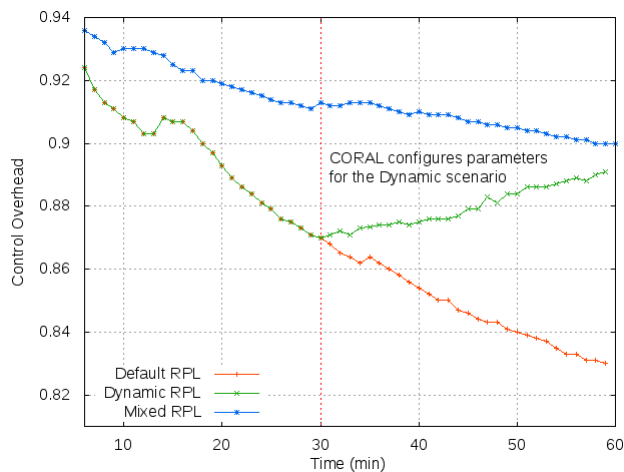


FIGURE 9. The network’s control overhead.

This outcome also highlights the benefits of offloading the control overhead to the fixed nodes.

Fig. 9 shows that PDR improvements come with an increase in control overhead, especially in the case of *Mixed configuration*. However, such an overhead increase may be traded for the PDR improvement in case of an emergency. These results highlight that the sooner the appropriate parameter setting, the better for the PDR. At the same time, this calls for further future improvements in the centralized platform accommodating the protocol’s mechanisms, e.g., implementing rapid intelligent detection of the network conditions. We omit a straight overhead comparison with the *Periodic-DIO* and the *Reverse Trickle*, since these approaches deviate from the RPL specification [5] and, thus, they produce extra, non-RPL-standard control packets.

C. DEEP RPL CONTROL RESULTS

In this section, we evaluate the Algorithm 1 and the *Deep RPL control* mechanism towards supporting *P2P communication*. As we previously explained (Section IV-A), we assume three distinct network topologies to show that the impact of the

proposed algorithm is independent of the nodes’ arrangement. Those topologies include one positively and one negatively biased, along with a random one, in order to highlight the degree of the advantage derived in each case.

To exhibit cases that can be most benefited from our mechanism, we start with the positively biased scenario (PBS) of a Lambda (Λ) topology depicted in Fig. 10a. Such a topology could be the case of a city’s backbone road network, where two equal-length (in Km) branches accommodate monitoring equipment (e.g., temperature sensor nodes) that collect environmental data. Typically, in RPL each node receives data from the node below, and it forwards them aggregated with its own measurements to the node one-hop forward to the sink. The sink is located at the top, where the two branches end up (e.g., large traffic lanes leading to the city center exit). In such a topology, nodes at the bottom-end of each branch can communicate with each other only by traversing the one branch up to the sink and all the way back down to the other branch. In Fig. 10a, using the MRHOF, the $3 \rightarrow 2$ *P2P communication* follows the path $3 \rightarrow 8 \rightarrow 6 \rightarrow 1 \rightarrow 7 \rightarrow 9 \rightarrow 2$, which entails slow and unreliable communication. Exploiting MRHOF-C, a small subset of links in the existent DODAG are re-arranged to facilitate the required connection, as shown in Fig. 10b. In practice, the *RPL Configurator* mechanism proceeds with coloring purple the sender 3, orange the receiver 2 and red the intermediate nodes 4, 5. The rest of the network nodes remain white. This color information is exploited by the Algorithm 1 which results with the desired *P2P communications* path: $3 \rightarrow 5 \rightarrow 4 \rightarrow 2$.

The impact of this change is depicted in Fig. 10c and 10d, where the CORAL starts with the default RPL OF (i.e., the MRHOF) in period 0 – 20 min, transits to the proposed MRHOF-C in 21 – 40 min, and concludes the experiment in period 41 – 60 min with the MRHOF again. We use graphs with the time parameter on x-axis only in this first experiment to demonstrate the ability of the CORAL to deploy a new OF dynamically using the *OF Deployer* component. During the whole period of 60 min we monitor and evaluate the TT (red line) and the RTT (green line) for the communication path of interest, i.e., $3 \rightarrow 2$ and we derive improvements up to 63.9 and 64.7 percent for TT and RTT respectively, which are depicted in Fig. 10c. For the same period, Fig. 10d shows that the $5 \rightarrow 1$ (sink) communication is slightly affected in terms of TT and RTT. In both graphs, in each OF change, a period of roughly 2 min passes without reporting TT and RTT values due to the connectivity disruptions caused by the global DODAG’s repair. Such an experiment shows that the CORAL and the *Deep RPL control* enable a *P2P communication*, improve the desired delivery time, and cause minor delays in the rest network, in a topology where the desired communication would be almost impossible in the *non-storing* mode of RPL.

A second negatively biased scenario (NBS) indicates the case where the network topology is not significantly improved from the *Deep RPL control* since the existing nodes’ arrangement can almost serve the desired *P2P* path.

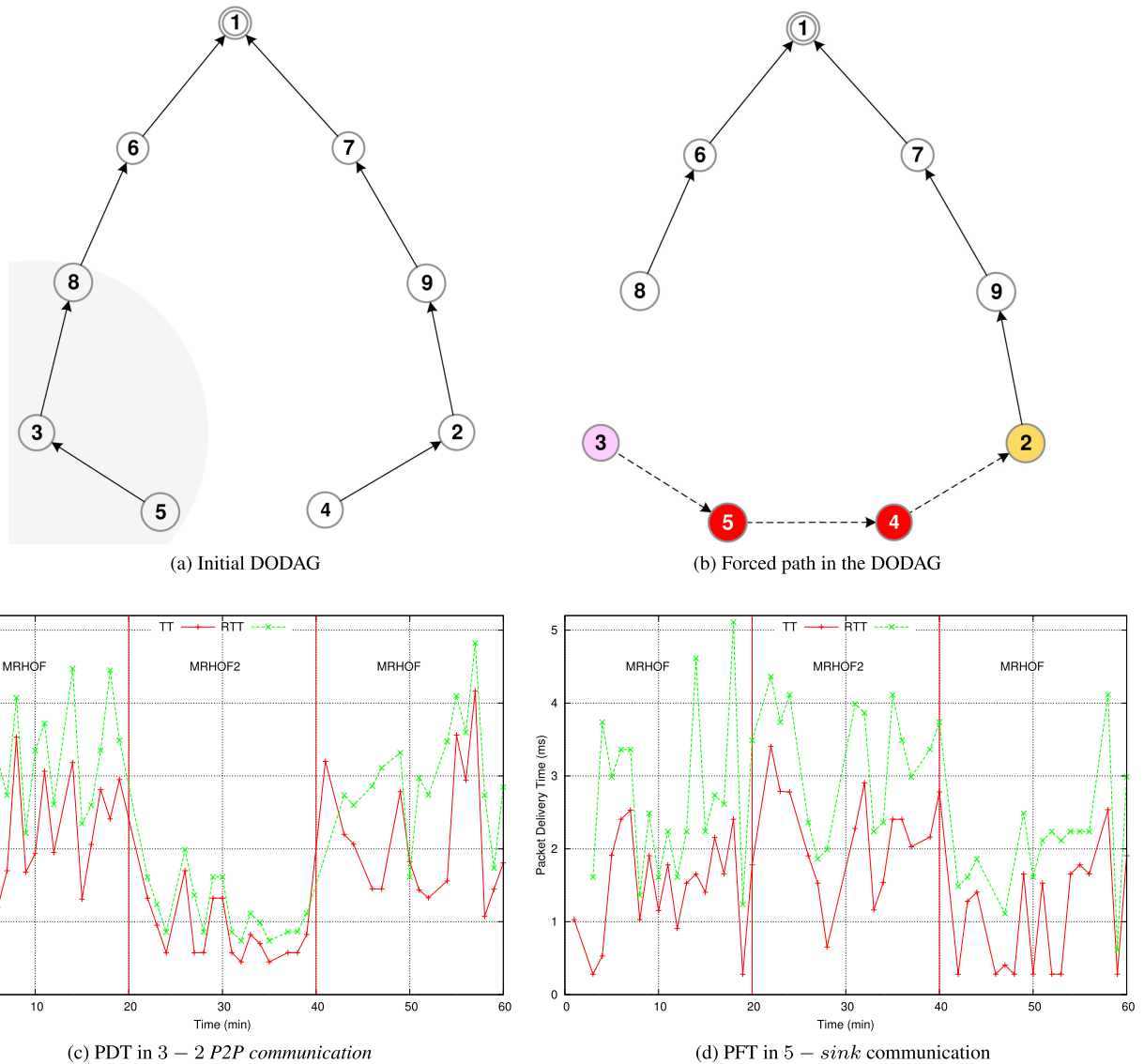
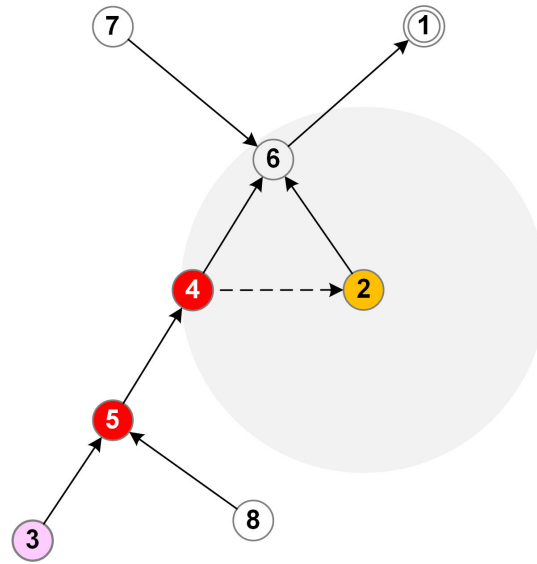


FIGURE 10. Lambda (A) topology in the positively biased scenario.

We use the topology of Fig. 11a, where the end-points of communication, i.e., the nodes 2 and 3, have a common “ancestor” (i.e., node 6), prior their data reach the sink node, i.e., the node 1. This entails the minimum change of creating the $4 \rightarrow 2$ link (dotted line) instead of the existent $4 \rightarrow 6$, in order to speed-up the $3 \rightarrow 2$ communication (when the MRHOF-C is used). The results of this experiment are also derived over a period of 60 min, and in this case, Fig.11b and 11c present the average TT and RTT values along with their standard deviation. We observe that the desired $3 \rightarrow 2$ communication is marginally improved (Fig.11b), mostly in terms of RTT, while the node 8, which is the most affected by the changed link in the DODAG, has a slight deterioration in respect to the TT, while it keeps improved RTT values (Fig.11c). An interesting outcome from both graphs is that MRHOF-C has better behavior in terms of RTT, indicating that the return path is specifically determined and, thus,

the nodes do not waste time looking for the next hop. All in all, this scenario exhibits that even when a local-repair does not end in a much different alternative path for the $P2P$ communication in interest, it is still worth to use the proposed MRHOF-C.

The last random topology scenario (RTS) can show the general applicability of our solution since it can better simulate a real-world case where randomly deployed nodes form a typical IoT network. The DODAG created by the RPL with the MRHOF is depicted in Fig. 12a. Assuming that an emerging network event triggers the need for $3 \rightarrow 2$ communication, the CORAL must find a path to serve them. Either pro-actively to save time or re-actively it launches the new MRHOF-C and the network nodes proceed with the DODAG’s reconstruction. As soon as all repairs have been completed, a new DODAG is constructed as depicted in Figure 12ba, and a direct path between the sender node 3 and the



(a) The initial and final DODAG

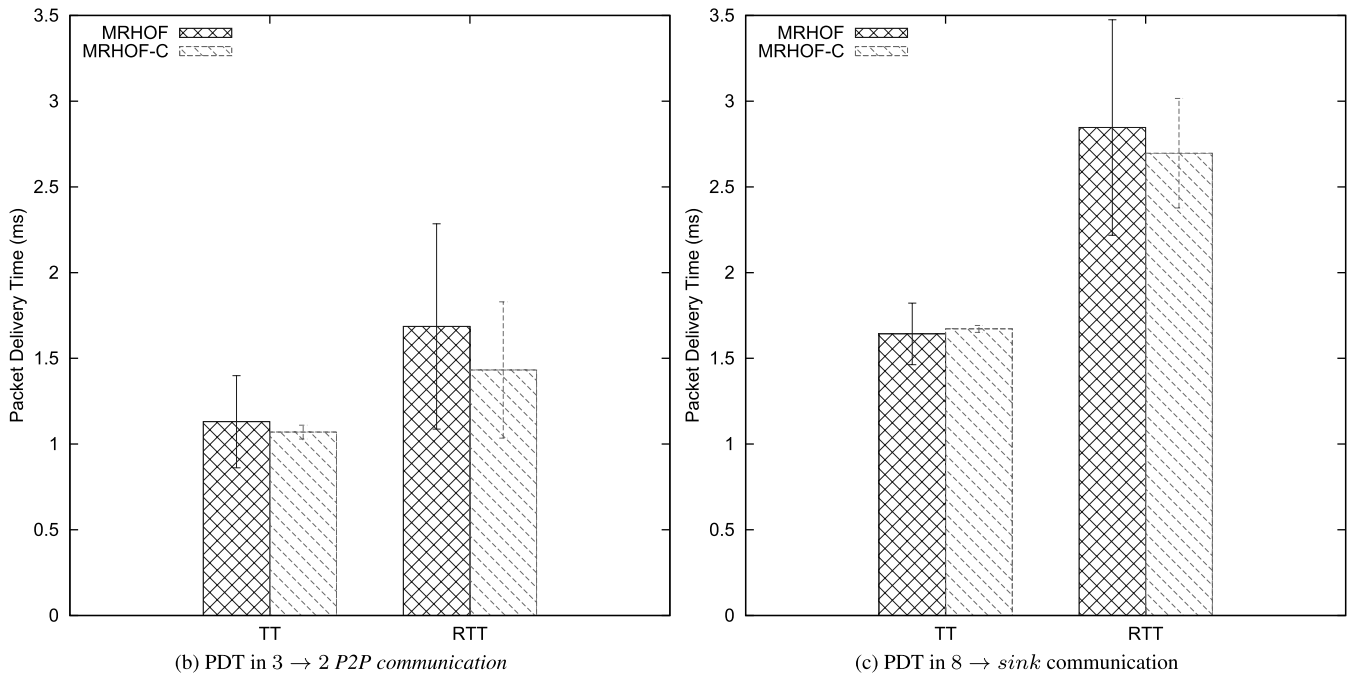


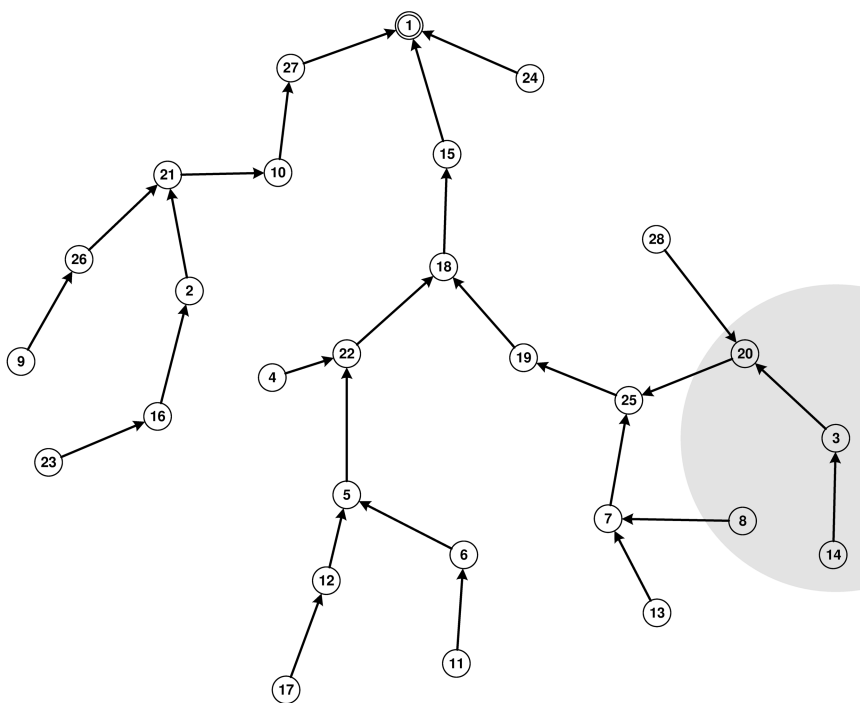
FIGURE 11. Topology of the negatively biased scenario.

receiver node 2 is established. As expected, the communication between the nodes 2 and 3 is significantly benefited; we derive improvements of 32.7 and 42 percent for TT and RTT respectively, as shown in Fig. 13.

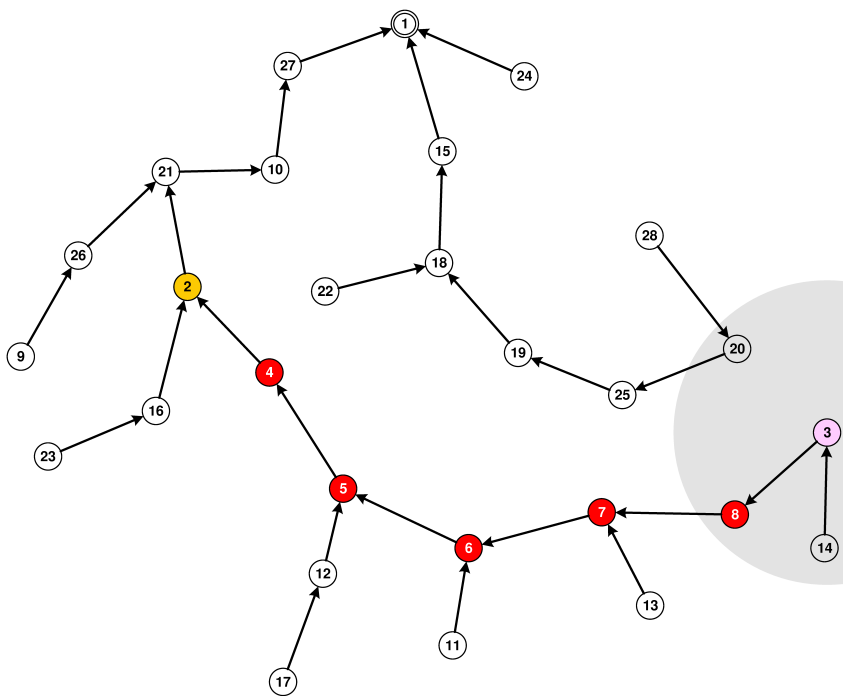
With the default MRHOF, even in storing mode, the round-trip of a 3 → 2 → 3 message is rarely successful—and only when there is no network traffic—because the returning packet goes through the sink node and most probably expires, given the RPL’s known inadequacy for P2P connections [9], [17]. As a matter of fact, Fig. 14 provides the PLR for the two compared OFs both in storing and non-storing mode. At first, it is natural that packet loss in the round-trip would

be higher compared to the one-way paths. The MRHOF has a bad performance of 70 percent loss in the one-way transmissions which rises to 80 percent when a round-trip is considered. Such loss ratio makes it practically inappropriate for P2P communication. MRHOF-C limits the losses at 20 percent and 35 percent, respectively. Then, in non-storing mode, only the loss in the one-way path can be measured and, thus, round-trip values are omitted. Figure 14 shows that MRHOF-C is more reliable than MRHOF in non-storing mode since it can reduce packet losses as much as 75 percent.

To demonstrate the effect of the DODAG’s repairment once the OF Deployer proceeds with the dynamical deployment



(a) Initial DODAG



(b) Forced path in the DODAG

FIGURE 12. Random topology of the neutral scenario.

of the MFHOR-C, we run a last experiment over the RTS in the non-storing mode. The experiment lasts 60 min and starts with the MRHOF. Fig. 15 shows that at the midterm MRHOF-C is employed, and as it is anticipated the ICMP

control packets are temporarily multiplied in the network. The red and green lines correspond to the ICMP send and received packets between the end-nodes 2, 3, and after a 2 min period of sharp increase, they revert to their previous levels.

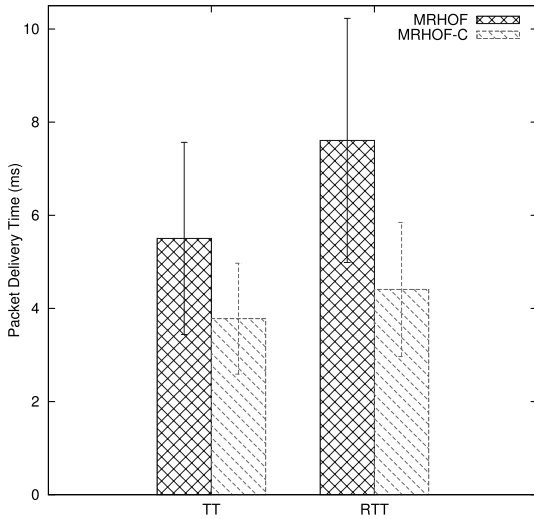


FIGURE 13. PDT in 3 – 2 P2P communication for the RTS.

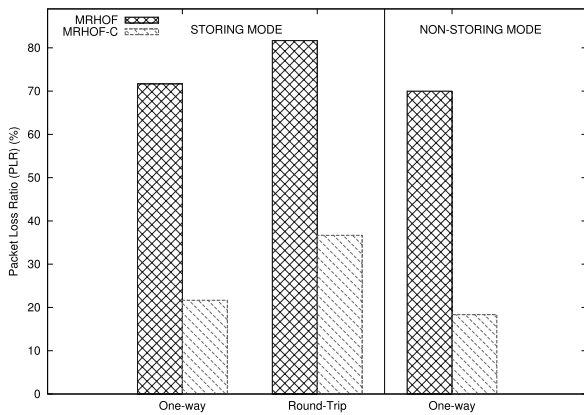


FIGURE 14. Packet loss in storing and non-storing mode for the RTS.

Thus, an interesting outcome is that the control overhead caused by the two OFs is at the same level. However, the useful information is that global and local-repairs demand a 2 – 3 min period to restore the control traffic. This could be a critical or non-critical period concerning the application requirements. In network environments with emergency events and strict safety requirements, a policy of pro-active MRHOF-C deployment is appropriate to save time towards serving a P2P communication. On the contrary, when applications can tolerate the connectivity disruptions during the OF transition, a re-active policy offered by the CORAL mechanisms is a good asset. These disruptions are indicated in Fig. 15 by the frequency of UDP on 30 min (i.e., the star marks on the curves). Finally, to validate our previous findings, Fig. 15 shows that UDP packets delivery time drops significantly from 1 ms to 0.6 ms.

An overview of the results derived by the three experimental setups (i.e., PBS, NBS, and RTS), is provided in Fig. 16. In this summary, it is clear that MRHOF-C is superior compared to the MRHOF in all cases and for both TT and RTT. Apart from the reported average values, it is worth to mention

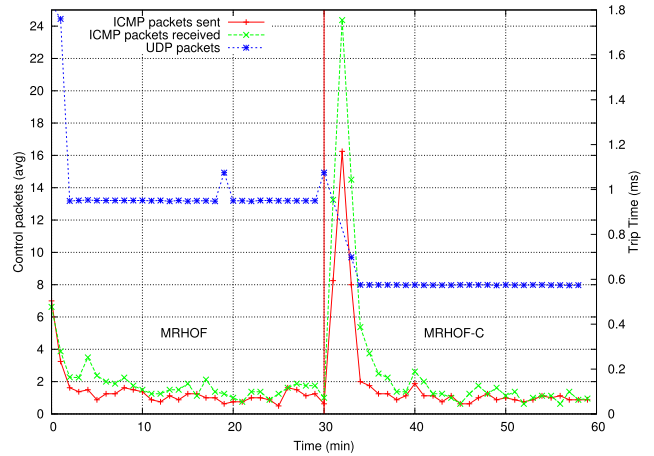


FIGURE 15. Control overhead and UDP delivery time in non-storing mode for the RTS.

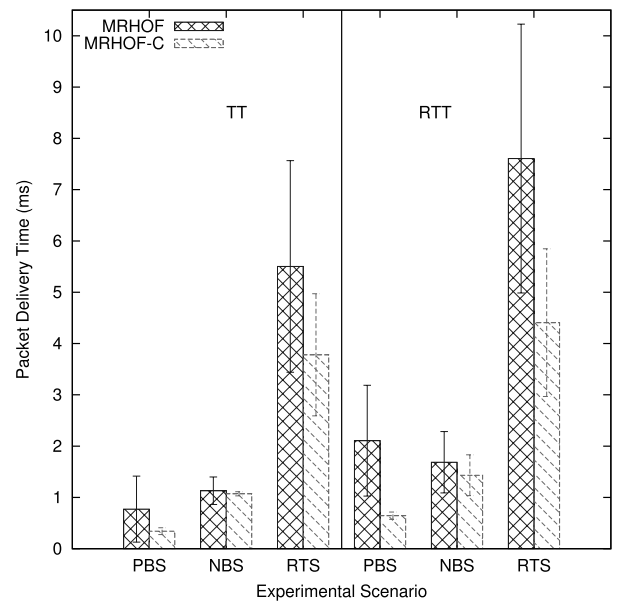


FIGURE 16. A comparison overview of the three experimental setups.

the standard deviations indicate that MRHOF-C is (much) more reliable than its competitor OF.

V. RELATED WORKS

In this section, we contrast our proposal to the related works that particularly focus on solutions tackling the RPL’s limitations for mobility and P2P communication, issues that are being addressed by the proposed Moderate and Deep RPL control strategies, respectively. Furthermore, we discuss other relevant SDN or SDN-inspired platforms.

Several works attempt to improve RPL’s behavior under mobility, mainly through aligning the responsiveness of its topology discovery mechanism to the characteristics of the topology changes, while considering the resource constraints of the devices. Although mobility can be handled through the MIPv6 standard [25], it is a resource expensive approach not matching well the characteristics of WSNs and IoTs, especially for medium to large topologies.

Several RPL adaptations to tackle mobility have been proposed in the literature. In [17] the topology adaptation is based on immediately probing and evaluating the ETX value of a new neighbor, along with this node's parent ETX, and then altering the standard DIO message by stamping it with this neighbor's ID. In order to handle dynamic topologies, the authors in [19] set the I_{min} to a max value and then reduce it to half after each new DIO. Both approaches are compared with our solution in Figs 7a, 7b and 8. We argue that the former solution is suitable for networks where the mobile nodes are mainly connected to the network, switching smoothly between parents, while the latter fits for mobile nodes with a steady pace and predictable behavior.

More works within the same domain include: (i) the adjustment of the DIS transmission times depending on the nodes' status in terms of mobility [18], [26]; (ii) the suggestion that all mobile nodes should be set as leafs, i.e., this way they do not send DIO messages and, thus, they cannot be chosen as parents [27] (this insightful idea was exploited in our experiments in order to exclude mobile nodes from creating "legitimate" paths); and (iii) a number of scenario-specific solutions, such as the autonomous moving of the sink towards the mobile nodes to reduce the number of hops the information traverses [28].

Other proposals augment RPL with mobility-aware features. For example, the recent papers [29], [30] propose mobility-aware adaptations in the RPL protocol that include new OFs. Other RPL extensions introduce hand-off handling mechanisms, such as [31] which elaborate on a relevant proactive mechanism called smart-HOP (i.e., lowering the trickle timer for wearable mobile nodes to improve their hand-off time), and [15] which propose the employment of additional proxy nodes to assist mobile communication. Furthermore, the work in [32] details a hand-off handling mechanism based on the average RSSI value, so the mobile nodes can immediately disconnect from the existing attachment points and connect to more suitable ones [33]. The latter functionality has been controlled by a management framework, underlining the advantages of such an approach. In addition, the following solutions employ mobility prediction mechanisms, such as: (i) adjusting the Trickle timer according to a prediction algorithm based on historical values [18]; (ii) introducing a fuzzy mobility estimator residing at an additional mobility-support layer [34]; or (iii) predicting nodes' mobility based on a Bayesian model [35]. Such last idea is insightful and could be incorporated in our platform to enhance the mechanisms at the *Control Layer*. Our plans also include the utilization of UPIs to collect network parameters, such as the RSSI and LQI, which can be exploited by a decision-making mechanism that once detecting nodes' mobility will correspond according to a predefined set of rules.

The above solutions, i.e., focusing on IoT mobility issues, can be categorized into two groups. The first proposes RPL adaptations (i.e., configuration tweaking or variations of its existing mechanisms, such as the Trickle timer), but targets particular network characteristics or use-cases, while our

solution remains generic and as such can benefit many diverse networks and topologies. The second implements RPL extensions with new architecture layers or functionalities (e.g., on mobility prediction or hand-off handling), but may lead to resource-expensive operation or protocols that are not compatible or consistent to the RPL standard, while our solution remains compliant with RPL, and as such can be easily and rapidly deployed without causing protocol malfunctions and inconsistencies.

Another set of RPL variations focus on *improving the P2P communication issues of RPL*, especially for downward routes [11], [12]. Even though RPL currently allows direct node communication through its storing or non-storing modes of operation, the storing mode is characterized by scalability and memory limitation issues, while the non-storing suffers by significant control overhead near the sink, causing congestion. The DualMOP-RPL protocol [36] supports coexisting modes of operation (i.e., both storing and non-storing) in a single RPL network to improve the downward routing inefficiency. The work in [37] provides a performance analysis quantifying the routing cost difference between DAG-based *P2P* and the shortest potential direct routes, i.e., suggesting that RPL should be improved in terms of direct node communication. In this context, RFCs 6997 [14] and 6998 [38] propose a reactive approach establishing shorter *P2P* paths, through defining temporary DODAGs that consider the destination node of the direct path as a sink. This process is considered as a third mode of operation, called *P2P* route discovery. However, this approach is characterized by additional overhead for the maintenance of multiple, although temporary, DODAGs [11].

The main relevant solutions are either protocols incompatible with RPL or hybrid versions of RPL, adopting alternative routing strategies for the *P2P communication* only. The AODV protocol [39] and its lightweight version LOADng [40] discover reactively bi-directional paths through communicating control packets, i.e., called Route Requests (RREQs). The 6TiSCH standardization initiative [41] proposes adopting hybrid protocols to improve node-to-node communication, such as the Asymmetric AODV-P2P-RPL in Low-Power and Lossy Networks (AODV-RPL) [42]. Another hybrid example is [43] which introduces a protocol combining the advantages of RPL with those of back-pressure routing protocol.

In our case, we are compliant to the RPL standard and allow dynamic changes in the existing protocol features to extend the applicability of RPL to novel use-cases. Additionally, we enable new functionalities to be added in a centralized controller, instead of the resource-constraint nodes. For example, a mobility prediction mechanism is less accurate from a node's viewpoint, compared to an algorithm residing at a centralized controller and taking decisions based on the global network view. Furthermore, our strategy avoids introducing additional overhead into the devices.

Relevant to our proposal control facilities and protocols include: (i) SDN-Wise [44], a logically-centralized IoT

protocol and SDN controller; (ii) our proposal [45] utilizing also the WiSHFUL infrastructure in an OpenFlow-like SDN control environment; (iii) an on-top of SDN-Wise approach for topology discovery [46]; and (iv) a platform implementing basic SDN features, i.e., topology and device management over application, control, and infrastructure layers [47]. These control platforms and protocols bring OpenFlow-like solutions to IoT environments, but they do not preserve the advantages of RPL. In [48], the authors suggest the association of a mote with a particular DODAG to be guided from a centralized controller while in [49] they discuss the synergy between TinySDN (an SDN protocol for IoT) with RPL and how they can assist each other. A recent Internet draft [41] suggests SDN-type centralized routing for time-sensitive flows and RPL for the rest of flows. Another relevant solution is [50], which introduces an SDN-like controller transmitting routing and control messages compatible to RPL to manipulate its operation. This approach works with legacy equipment. In another recent work [51], interoperability between protocol stacks is introduced, to provide controller discovery, and reduce control overhead.

As a bottom line, RPL can cover a wide range of IoT deployments but with manual configurations and without obvious performance outcomes. Here, we argue that a centralized control facility can implement closed control loops, monitoring, deciding and configuring RPL parameters on-the-fly, depending on the mobility status of each node and the application requirements (e.g., requesting direct communication between nodes).

VI. CONCLUSION AND FUTURE WORK

This paper presents two SDN-inspired routing control strategies for the IoT, namely the *Moderate RPL* and the *Deep RPL control*, which evolutionarily tackle the *mobility* and *P2P communication* issues of the RPL in the sense that they remain consistent to the protocol's standard. Our CORAL facility and its components, namely the *OF Deployer* and *RPL Configurator*, act as enablers to dynamically change the OF—along with the RPL operate to construct the DODAG—and appropriately configure either its parameters (e.g., I_{min}) or features (link coloring), both in real-time. Our results confirm that the *Moderate RPL control* strategy can bring improvements in PDR of the order of 33 percent at the cost of increased control overhead. However, offloading this overhead to the fixed infrastructure can eliminate its impact and alleviate the network in emergency cases. On the other hand, the *Deep RPL control* and the newly introduced MRHOF-C Objective Function bring multiple advantages: enable a *P2P* communication both in storing and non-storing mode, improve the packet delivery time between the nodes of interest (up to 42 percent), significantly reduce the packet loss ratio (as much as 75 percent) and keep the rest network almost untouched from the local or global-repairs executed. The obvious cost of control overhead seems to temporarily disturb the network and can be eliminated in case that MRHOF-C is employed pro-actively.

The results of the proposed OF, offer insight for approaching the parents' selection process and the DODAG's construction as a multi-objective optimization problem. Obviously, finding an optimal solution for one OF may require to accept a poor solution for some other(s). Thus, different weights to each OF enable to consider them in conjunction. Such a vision can ideally be supported by the CORAL architecture, whose planes can serve as place-holders for intelligent mechanisms detecting network conditions (e.g., nodes' failures due to connectivity or battery drain issues, or presence of malicious nodes) and tuning as a response a weighted OF or deploy the most appropriate automatically.

REFERENCES

- [1] J. M. Talavera, L. E. Tobón, J. A. Gómez, M. A. Culman, J. M. Aranda, D. T. Parra, L. A. Quiroz, A. Hoyos, and L. E. Garreta, "Review of IoT applications in AGRO-industrial and environmental fields," *Comput. Electron. Agricult.*, vol. 142, pp. 283–297, Nov. 2017.
- [2] J. Qi, P. Yang, G. Min, O. Amft, F. Dong, and L. Xu, "Advanced Internet of Things for personalised healthcare systems: A survey," *Pervasive Mobile Comput.*, vol. 41, pp. 132–149, Oct. 2017.
- [3] S. M. R. Islam, D. Kwak, M. Humaun Kabir, M. Hossain, and K.-S. Kwak, "The Internet of Things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, Jun. 2015.
- [4] Y. B. Zikria, M. K. Afzal, F. Ishmanov, S. W. Kim, and H. Yu, "A survey on routing protocols supported by the Contiki Internet of Things operating system," *Future Gener. Comput. Syst.*, vol. 82, pp. 200–219, May 2018.
- [5] T. Winter, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, document RFC 6550, IETF, 2012. [Online]. Available: <https://tools.ietf.org/html/rfc6550>
- [6] T. Tsvetkov and A. Klein, "RPL: IPv6 routing protocol for low power and lossy networks," *Network*, vol. 59, pp. 59–66, Jul. 2011.
- [7] O. Gaddour and A. Koubâa, "RPL in a nutshell: A survey," *Comput. Netw.*, vol. 56, no. 14, pp. 3163–3178, 2012.
- [8] G. Violettas, T. Theodorou, S. Petridou, A. Tsioukas, and L. Mamas, "An experimentation facility enabling flexible network control for the Internet of Things," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Nov. 2017, pp. 992–993.
- [9] G. Violettas, S. Petridou, and L. Mamas, "Routing under heterogeneity and mobility for the Internet of Things: A centralized control approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.
- [10] P. Ruckebusch, S. Giannoulis, D. Garlisi, P. Gallo, P. Gawlowicz, A. Zubow, M. Chwalisz, E. De Poorter, I. Moerman, I. Tinnirello, and L. DaSilva, "WiSHFUL: Enabling coordination solutions for managing heterogeneous wireless networks," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 118–125, Sep. 2017.
- [11] P. O. Kamgueu, E. Nataf, and T. D. Ndie, "Survey on RPL enhancements: A focus on topology, security and mobility," *Comput. Commun.*, vol. 120, pp. 10–21, May 2018.
- [12] T. Clausen, U. Herberg, and M. Philipp, "A critical evaluation of the IPv6 routing protocol for low power and lossy networks (RPL)," in *Proc. IEEE 7th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2011, pp. 365–372.
- [13] T. Pinney. (Oct. 2013). *RPL Applicability in Industrial Networks Draft-IETF-Roll-RPL-Industrial-Applicability-02*. [Online]. Available: <https://www.ietf.org/archive/id/draft-ietf-roll-rpl-industrial-applicability-02.txt>
- [14] M. Goyal, E. Baccelli, M. Philipp, A. Brandt, and J. Martocci, *Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks*, document RFC 6997, 2013.
- [15] R. Silva, J. S. Silva, and F. Boavida, "A proposal for proxy-based mobility in WSNs," *Comput. Commun.*, vol. 35, no. 10, pp. 1200–1216, Jun. 2012.
- [16] P. Ruckebusch, S. Giannoulis, E. De Poorter, I. Moerman, I. Tinnirello, D. Garlisi, P. Gallo, N. Kaminski, L. DaSilva, P. Gawlowicz, M. Chwalisz, and A. Zubow, "A unified radio control architecture for prototyping adaptive wireless protocols," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2016, pp. 58–63.

- [17] K. C. Lee, R. Sudhaakar, J. Ning, L. Dai, S. Addepalli, J. P. Vasseur, and M. Gerla, "A comprehensive evaluation of RPL under mobility," *Int. J. Veh. Technol.*, vol. 2012, Dec. 2011, Art. no. 904308.
- [18] I. El Korbi, M. Ben Brahim, C. Adjih, and L. A. Saidane, "Mobility enhanced RPL for wireless sensor networks," in *Proc. 3rd Int. Conf. Netw. Future (NOF)*, Nov. 2012, pp. 1–8.
- [19] C. Cobârzan, J. Montavont, and T. Noël, "Analysis and performance evaluation of RPL under mobility," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2014, pp. 1–6.
- [20] P. Thubert, *Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)*, document RFC 6552, 2012.
- [21] O. Gnawali and P. Levis, "The minimum rank with hysteresis objective function," draft-ietf-roll-minrank-hysteresis-of-04 (work in progress), May 2011.
- [22] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki—A lightweight and flexible operating system for tiny networked sensors," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, Nov. 2004, pp. 455–462.
- [23] J.-P. 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, IETF, 2012.
- [24] O. Alay, A. Lutu, D. Ros, R. Garcia, A. F. Hansen, A. Brunstrom, M. A. Marsan, H. Lonsethagen, and V. Mancuso, "MONROE: Measuring mobile broadband networks in Europe," in *Proc. IRTF ISOC Workshop Res. Appl. Internet Measurements (RAIM)*, Oct. 2015, pp. 1–4.
- [25] D. Johnson, C. Perkins, and J. Arkko, *Mobility Support in IPv6*, document RFC 3775, IETF, Jun. 2004.
- [26] C. Cobârzan, J. Montavont, and T. Noel, "Integrating mobility in RPL," in *Proc. Eur. Conf. Wireless Sensor Netw.* Cham, Switzerland: Springer, Feb. 2015, pp. 135–150.
- [27] F. Gara, L. B. Saad, R. B. Ayed, and B. Tourancheau, "RPL protocol adapted for healthcare and medical applications," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Aug. 2015, pp. 690–695.
- [28] L. B. Saad and B. Tourancheau, "Sinks mobility strategy in IPv6-based WSNs for network lifetime improvement," in *Proc. 4th IFIP Intl. Conf. New Technol., Mobility Security (NTMS)*, Feb. 2011, pp. 1–5.
- [29] S. Sanshi and C. D. Jaidhar, "Enhanced mobility aware routing protocol for low power and lossy networks," *Wireless Netw.*, vol. 25, no. 4, pp. 1641–1655, May 2019.
- [30] S. Sanshi and C. D. Jaidhar, "Enhanced mobility routing protocol for wireless sensor network," *Wireless Netw.*, 2018. doi: [10.1007/s11276-018-1816-y](https://doi.org/10.1007/s11276-018-1816-y).
- [31] H. Fotouhi, D. Moreira, and M. Alves, "mRPL: Boosting mobility in the Internet of Things," *Ad Hoc Netw.*, vol. 26, pp. 17–35, Mar. 2015.
- [32] F. Gara, L. B. Saad, E. B. Hamida, B. Tourancheau, and R. B. Ayed, "An adaptive timer for RPL to handle mobility in wireless sensor networks," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Sep. 2016, pp. 678–683.
- [33] H. Fotouhi, D. Moreira, M. Alves, and P. M. Yomsi, "mRPL+: A mobility management framework in RPL/6LoWPAN," *Comput. Commun.*, vol. 104, pp. 34–54, May 2017.
- [34] J. Ko and M. Chang, "MoMoRo: Providing mobility support for low-power wireless applications," *IEEE Syst. J.*, vol. 9, no. 2, pp. 585–594, Jun. 2015.
- [35] F. Soma, I. El Korbi, C. Adjih, and L. A. Saidane, "A modified RPL for wireless sensor networks with Bayesian inference mobility prediction," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Sep. 2016, pp. 690–695.
- [36] J. Ko, J. Jeong, J. Park, J. A. Jun, O. Gnawali, and J. Paek, "DualMOP-RPL: Supporting multiple modes of downward routing in a single RPL network," *ACM Trans. Sensor Netw.*, vol. 11, no. 2, p. 39, 2015.
- [37] W. Xie, M. Goyal, H. Hosseini, J. Martocci, Y. Bashir, E. Baccelli, and A. Durrezi, "A performance analysis of point-to-point routing along a directed acyclic graph in low power and lossy networks," in *Proc. 13th Int. Conf. Netw. Inf. Syst.*, Sep. 2010, pp. 111–116.
- [38] J. Martocci, M. Goyal, A. Brandt, and E. Baccelli, *A Mechanism to Measure the Routing Metrics Along a Point-to-Point Route in a Low-Power and Lossy Network*, document RFC 6998, 2013.
- [39] C. Perkins, E. Belding-Royer, and S. Das, *Ad Hoc On-Demand Distance Vector (AODV) Routing*, document RFC 3561, 2003.
- [40] J. Tripathi, J. C. De Oliveira, and J.-P. Vasseur, "Proactive versus reactive routing in low power and lossy networks: Performance analysis and scalability improvements," *Ad Hoc Netw.*, vol. 23, pp. 121–144, Dec. 2014.
- [41] P. Thubert, T. Watteyne, R. Struik, and M. Richardson, *An Architecture for IPv6 Over the TSCH Mode of IEEE 802.15.4*, document Internet-Draft draft-ietf-6tisch-architecture-14, Working Draft, IETF Secretariat, 2018. [Online]. Available: <https://tools.ietf.org/html/draft-ietf-6tisch-architecture-15>
- [42] M. Zhang, A. Sanga, C. Perkins, and S. Anand, (2016). *Asymmetric AODV-P2P-RPL in Low-Power and Lossy Networks (LLNs) Draft-Satish-Roll-AODV-RPL-03*. [Online]. Available: <https://www.ietf.org/archive/id/draft-satish-roll-aodv-rpl-03.txt>
- [43] Y. Tahir, S. Yang, and J. McCann, "BRPL: Backpressure RPL for high-throughput and mobile IoTs," *IEEE Trans. Mobile Comput.*, vol. 17, no. 1, pp. 29–43, Jan. 2018.
- [44] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for Wireless Sensor networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr./May 2015, pp. 513–521.
- [45] T. Theodorou and L. Mamas, "Software defined topology control strategies for the Internet of Things," in *Proc. 3rd Int. Workshop Orchestration Softw.-Defined Infrastructures, IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2017, pp. 236–241.
- [46] N. Abdolmaleki, M. Ahmadi, H. T. Malazi, and S. Milardo, "Fuzzy topology discovery protocol for SDN-based wireless sensor networks," *Simul. Model. Pract. Theory*, vol. 79, pp. 54–68, Dec. 2017.
- [47] S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat, "Soft-WSN: Software-defined WSN management system for IoT applications," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2074–2081, Sep. 2018.
- [48] C. Jacquenet and M. Boucadair, "A software-defined approach to IoT networking," *ZTE Commun.*, vol. 1, pp. 1–12, Mar. 2016.
- [49] B. T. de Oliveira, R. C. A. Alves, and C. B. Margi, "Software-defined wireless sensor networks and Internet of Things standardization synergism," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Oct. 2015, pp. 60–65.
- [50] E. Municio, J. Marquez-Barja, S. Latré, and S. Vissicchio, "Whisper: Programmable and flexible control on industrial IoT networks," *Sensors*, vol. 18, no. 11, p. 4048, Nov. 2018.
- [51] M. Baddeley, R. Nejabat, G. Oikonomou, M. Sooriyabandara, and D. Simeonidou, "Evolving SDN for low-power IoT networks," in *Proc. 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*, Jun. 2018, pp. 71–79.



GEORGE VIOLETTAS received the bachelor's degree from Hellenic Open University, and the M.Sc. degree in applied informatics from the University of Macedonia, Thessaloniki, Greece, where he is currently pursuing the Ph.D. degree in software-defined mobile networks. He is also the Head of the Department and a Senior Lecturer with the Applied Engineering College (former Technical Trainers College), Riyadh, Saudi Arabia. He has worked for more than 25 years as a Project Manager for big scale Wireless Networks for the private sector and the Greek government and taught in many colleges and vocational institutes in Greece and in the Gulf area. He is also a Senior Researcher with NECOS OC2 (H2020); and also with WiSHFUL OC2 (H2020), and MONROE OC2 (H2020). His current research interests include the Internet of Things (IoT), wireless sensor networks (WSN), edge computing, 5G, and information security.



SOPHIA PETRIDOU received the B.Sc. Diploma and Ph.D. degrees from the Department of Informatics, Aristotle University of Thessaloniki, Greece, in 2008 and 2000, respectively. She is currently an Assistant Professor with the Department of Applied Informatics, University of Macedonia. Her current research interests include optical networks (MAC protocols), wireless networks (wireless data broadcasting), network QoS, clustering algorithms, and the probabilistic model

checking of protocols.

She is a member of the IEEE Computer Society. She received the scholarships from the Greek Telecommunication Company (OTE) for her performance in subjects of networks and telecommunications, in 1999 and 2000. She serves as an Associate Editor for the *International Journal of Communication Systems*.



LEFTERIS MAMATIAS was a Researcher with University College London, U.K.; the Space Internetworking Center, Democritus University of Thrace, Greece; and DoCoMo Eurolabs, Munich. He is currently an Assistant Professor with the Department of Applied Informatics, University of Macedonia, Greece, where he leads the Softwarized & Wireless Networks Research Group. He participated in many international research projects, such as NECOS (H2020),

FED4FIRE+ OC4 (H2020), WISHFUL OC2 (H2020), MONROE OC2 (H2020), Dolphin (FP7), UniverSELF (FP7), and Extending Internet into Space (ESA). He has published more than 60 articles in international journals and conferences. His research interests include software-defined networks, the Internet of Things, 5G networks, and multi-access edge computing. He served as a General Chair for the WWIC 2016 Conference and the INFOCOM SWFAN 2016 Workshop, and as a TPC Chair for the INFOCOM SWFAN 2017, E-DTN 2009, and IFIP WWIC 2012 conferences/workshops. He served as a Guest Editor for the Elsevier *Ad Hoc Networks* Journal.

• • •