Routing under heterogeneity and mobility for the Internet of Things: a centralized control approach

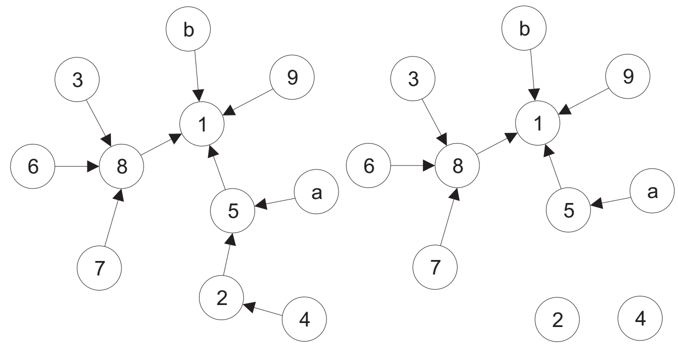
George Violettas, Sophia Petridou, Lefteris Mamatas
Department of Applied Informatics
University of Macedonia, Thessaloniki, Greece
Email: {georgevio, spetrido, emamatas}@uom.edu.gr

***Abstract*—Being part of the Internet of Things (IoT), the Wireless Sensor Networks (WSNs) inherit characteristics such as large-scale deployment, dynamicity, heterogeneity and mobility. These aspects mandate elasticity in many network functions and especially in routing. The IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) is the state-of-the-art routing protocol for resource-constrained devices in environments with signal issues, but it was not designed to support mobility. However, mobility is fundamental in critical WSNs applications, e.g., it improves sensing coverage and brings back connectivity. In this paper, we propose a centralized routing control approach, which exploits the global view of the network inspired by the Software-Defined Networks (SDNs), in order to provide dynamic end-to-end routing service in heterogeneous environments. We briefly present our novel Cross-Layer Control of Data Flows (CORAL) framework, which–in the context of the current work–has been enhanced with management and control features to provide elasticity in the RPL's functionality. In practice we developed and propose dynamic and individual parameters' configuration and adaptation, live monitoring of performance through visualization and Ansible-based protocol's deployment. Our results show that dynamic RPL configuration, which tackles mobility issues on-the-fly, along with individual configuration which handles them separately, can efficiently tune the protocol's performance trade-offs, e.g., between packet delivery ratio (PDR) and routing overhead, bringing up to 35% improvement in PDR and offloading the control overhead from the mobile nodes.**

I. Introduction

Routing is an especially challenging network function in the Internet of Things (IoT), basically due to power, storage, memory, processing and signal limitations of the connected devices, and nowadays, in view of characteristics such as large-scale deployment, dynamicity, heterogeneity and mobility. More and more environmental, agriculture, or smart city applications require extended and reliable sensing coverage. The RPL protocol is a distance-vector IPv6 routing protocol for Low-power and Lossy Networks (LLNs) and so far it is a very good solution, since it has a number of configuration parameters that cover a wide-range of alternative deployments [1], [2]. However, it is particularly adjusted to energy-efficient and long-term periodic sensing over fixed motes' deployment [3].

In practice, RPL organizes network nodes, e.g., motes of a WSN, as a Destination-Oriented Directed Acyclic Graph (DODAG) rooted at a single destination called root or sink [2], [3]. The root is the only node that can launch the DODAG's construction, which is based upon the exchange of routing



(a) Initial graph setup (b) Mobility issue

Fig. 1. DODAG's setup example

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 root, 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 root) 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. Fig. 1a depicts an example of DODAG's initial setup for a WSN consisting of 11 motes, among which the mote with $id = 1$ is the root.

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 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 PDR (Packet Delivery Ratio), while restricting the control overhead towards saving node's power consumption [3]. To achieve the aforementioned trade-off, the DIO messages are send periodically, but their interval ranges from I_{min} up to I_{max}, where $I_{max} = I_{min} * 2^{I_{doubling}}$ (default RPL configuration specifies $I_{min} = 2^{12} = 4096\ ms$ and $I_{doubling} = 8$ which entails $I_{max} = 2^{12+8} = 17.5\ min$). Actually, the timer's duration is doubled each time it fires. Any change in the DODAG, e.g., unreachable parent or new parent selection, resets the trickle timer to I_{min}.

Customizing such configurations is basically manual, global, and often unpredictable in terms of outcome, especially in heterogeneous and mobile environments where fixed motes co-exist with mobile ones and hence, specific requirements should be taken care of individually. For example, when a mobile mote moves outside its parent's range, it is getting disconnected from the graph, e.g., the mote with $id = 2$ in Fig. 1b, affecting the routing process and consequently the network's performance. Handling either fixed or mobile motes manually, or in compile-time, consumes time which in emergency cases can be very critical.

Along these lines, we highlight the need of centralized control in routing so as to address heterogeneity and mobility issues. More precisely, we use and briefly present our novel Cross-Layer Control of Data Flows (CORAL) framework. In the context of this paper, we extend CORAL to decouple complexity from the RPL protocol and offload it to the network control software deployed at the fixed infrastructure. This extension provides management and control features that: i) enforce appropriate protocol's configurations and adaptations in a softwarized and on-the-fly manner; ii) monitor the network and detect changes in real time; and iii) guide dynamic RPL's deployment. Furthermore, we implemented an adaptable version of RPL with the appropriate hooks to the above features. According to our results such centralized approach inherits the advantages of RPL (e.g., resource-efficiency), while tackling heterogeneity and mobility aspects in IoT deployments. A demo paper on this work is [4]; the CORAL was also exploited in an OpenFlow-like protocol for topology control [5], [6].

The rest of the paper is organized as follows: Section II is a brief overview of related studies, while a motivating use-case scenario is discussed in Section III. Section IV presents our integrated IoT control facility and Section V provides an evaluation of our adaptable RPL protocol controlled by the proposed management and control features. We conclude with some future work insights in Section VI.

II. Related works

In this section, we discuss: (i) the motivation of our work, i.e., the RPL's inefficiency in heterogeneous environments (e.g., co-existence of fixed and mobile nodes); (ii) proposed RPL variations trying to improve its behavior in mobile environments; and (iii) related to our solution control platforms and protocols.

A number of works highlight the RPL's weakness in converging to a stable list of routes, even in fixed networks due to the absence of ideal route estimators [7]. Furthermore, nodes may change parents independently of the routing objective function, generating large overhead, while the PDR decreases because of the unsuitable trickle timer value (i.e., the DIO messages' interval). For example, the topology probe interval gradually increases and produces a delayed response to the topology changes due to mobility [8]. There is an inherent focus of the RPL design on static networks with limited local adaptability [9], e.g., the RPL specifications do not cover when and how DIS messages should be sent [10]. However, as we discuss below, it has the potential to improve its behavior in mobile environments through adjusting its main configuration parameters or mechanisms.

Several RPL adaptations to tackle mobility have been proposed in the literature, such as: (i) the immediate topology adaptation for a new neighbor based on stamping the DIO message with its parent's ID and the immediate communication of DIOs and DAOs upon a new parent election [8]; (ii) setting I_{min} to a max value and its reduction to half after each new DIO, in order to handle dynamic topologies [10]; (iii) the adjustment of the DIS transmission times depending on the node status in terms of mobility [9], [11]; and (iv) 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 transverses [12].

Other approaches to handle mobility include dividing of the network in co-centric circles and the usage of multiple routers assisting the mobile nodes to connect [13], or the implementation of a hand-off handling mechanism based on the average RSSI value [14], so mobile nodes can immediately disconnect from the existing attachment points and connect to more suitable ones [15]. The latter capability has been controlled by a management framework, underlining the advantages of such approach. 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.

Relevant to our proposal control facilities include: (i) SDN-Wise [16], an OpenFlow-like IoT protocol and SDN controller; (ii) an on top of SDN-Wise approach for topology discovery [17]; and (iii) the platform [18] implementing basic SDN features, i.e., topology and device management over application, control, and infrastructure layers. These control platforms and protocols bring OpenFlow-like solutions to IoT environments, but they do not preserve the advantages of RPL. In [19], the authors suggest the association of a mote with a particular DODAG to be guided from a centralized controller and the [20] discusses the synergy between TinySDN, an SDN protocol for IoT, with RPL and how they can assist each other. The recent Internet draft [21] suggests SDN-type centralized routing for time-sensitive flows and RPL for the rest of flows.

In this paper, we discuss our integrated SDN-inspired control facility with an adaptable RPL protocol which improves the performance of RPL in heterogeneous network deployments, along with an experimental analysis. To further motivate our approach, we discuss a representative use-case scenario below.

III. Use-case scenario

To demonstrate the need of a centralized routing control facility for heterogeneous and mobile IoT deployments, we consider a mine environment. For human safety reasons, fixed infrastructure in a central control-room monitors the whole place in real-time exploiting both static motes located at mine paths, and wearables measuring vital indicators of miners. Assuming that the graph of Fig. 1a is an abstract view of this scenario, the mote 1-i.e., the sink-collects all measurements, where some motes may be wearables, e.g., $\{2, \ldots, 6\}$, and the rest are fix-positioned devices, e.g., $\{7, 8, 9, a, b\}$.

TABLE I
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	11	Y	Fig 1a	8 12	13.8 45.0
2	15	N	chain	8 12	6.2 50.9
3	15	N	lambda (Λ) - sink on top	8 12	4.8 26.8
4	30	N	as in [3]	8 12	5.1 23.0
5	30	N	chain	8 12	11.3 107.4
6	50	N	random	8 12	10.2 27.3
7	100	N	random	8 12	32.4 68.1

Such networks face a number of issues: obstacles resulting to Non-line-of-sight (nLoS), isolated nodes, frequent and radical topology changes, weak wireless signals, sensor battery drainage, and device hardware heterogeneity. Under these circumstances, even the slightest communication opportunity should be exploited to avoid or eliminate interruptions of critical services, e.g., monitoring the miners' oxygen levels.

Table I 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 motes, heterogeneity in motes' behavior (fixed and/or mobile) and topology type. For example, in a mine with 11 randomly positioned motes, including both fixed and mobile, in case of an accident close to mote 4 (e.g., an isolated miner loses his senses) at 20 sec, the default RPL configuration fails to route its signal. Thus, it is important to begin with an "aggressive" graph setup policy to ensure that all motes are being connected to the graph, e.g., within almost 14 sec, and successfully report their data at the cost of control overhead. Actually, this cost at the given time is not actually an issue for the PDR, since data cannot be delivered without the routing paths provided by the topology graph. This is also useful in cases where the controller uses the same channel to communicate control messages.

In this paper, we argue for the idea of a centralized control facility providing the options of dynamic and individual motes' configuration. More precisely, we can start with a minimum I_{min} to setup the DODAG fast and then continue with high values of I_{min} for fixed motes and low for 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, i.e., in run time. Furthermore, live network monitoring enables early detections of abnormal or inefficient routing behavior, new configuration decisions

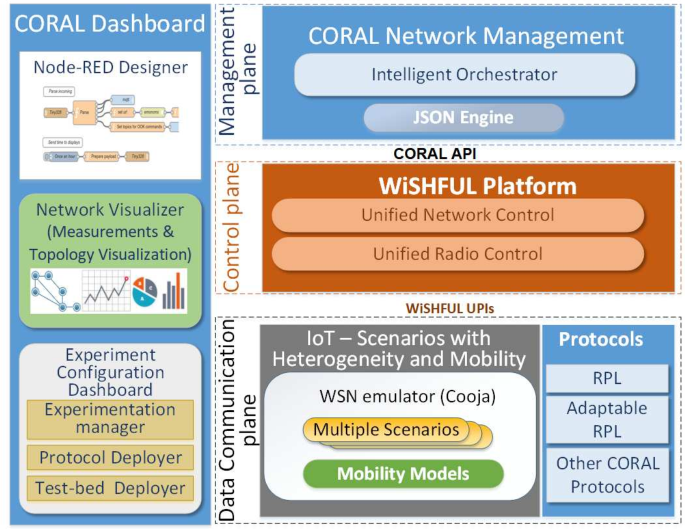


Fig. 2. The architecture of the CORAL infrastructure

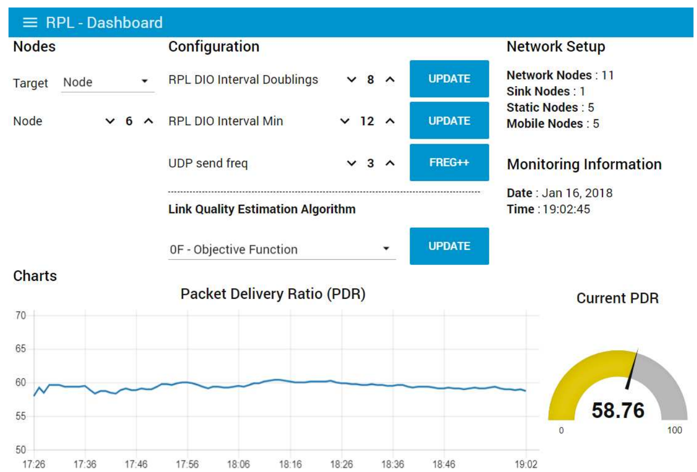


Fig. 3. On-the-fly RPL configuration with the CORAL dashboard

and their dynamic/individual enforcement. The details of the proposed control facility follow.

IV. THE PROPOSED IOT CONTROL FACILITY

The CORAL platform, introduced in the demo paper [4][1], is a general-purpose softwarized IoT control infrastructure implementing the discussed routing control features. Fig. 2 illustrates a high-level view of the CORAL architecture. Developed to provide SDN-inspired protocol control, it consists of three different planes which basically integrate the RPL's functionality and evolutionarily provide elasticity in routing, associated with our adaptable version of RPL. We discuss the different planes of the CORAL architecture right afterwards.

The *Data Communication plane* accommodates the multiple network settings' scenarios along with the motes' mobility

[1]The demo video and the equivalent source code can be found at: http://myminiurl.net/GKjBM and http://github.com/SWNRG/wishful-coral

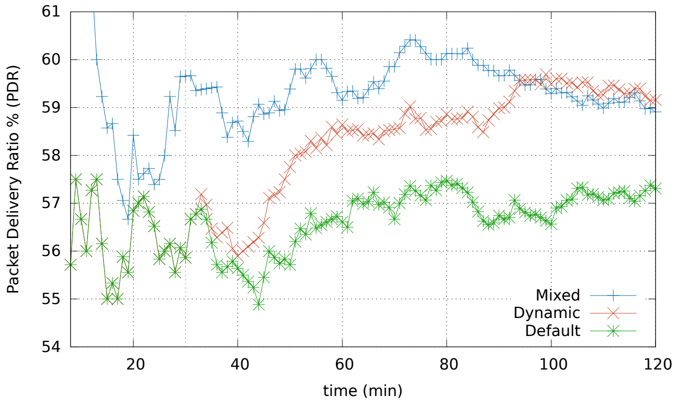


Fig. 4. Proof-of-concept experiment

models. To support dynamicity, this plane can be loaded and/or changed both in compile-time, according to bibliography examples [8], [10], but most importantly in run-time. In practice, we exploit Ansible scripting to dynamically parameterize the Cooja WSN emulator.

The *Control plane* offers abstracted and logically-centralized control of the network. Through *Unified Network* and *Radio Control*, it handles heterogeneity, while it uses separate interfaces to communicate northbound with the management plane, i.e., the CORAL API, and southbound with the data communication plane, i.e., the Unified Programming Interfaces (UPIs) provided by the WiSHFUL project [22]. The UPIs support both separate control-channel communication over serial interfaces and data-channel communication using the CoAP protocol. Here, we use the former method because we focus on the impact of the proposed paradigm rather than on the control protocol. The latter is complicated enough to deserve an independent study.

The *Management plane* provides high-level network management functionality by employing an *Intelligent Orchestrator* to decouple complexity from the RPL protocol and offload it to the network control software deployed at the fixed infrastructure. This way, it can monitor the network and detect changes in real time, and then enforce appropriate configurations in a softwarized and on-the-fly manner through JSON engine. This plane serves as a place-holder for intelligent algorithms, such as feature extraction link quality estimation [23]. The CORAL received the 1st runner-up award in the eWINE Grand Challenge, a competition organized by the same project (http://ewine-project.eu). Our plans include intelligent mechanisms in this plane to support mobility detection.

Our control facility interacts with the user through the *CORAL Dashboard*, a highly flexible graphical user interface (GUI), implemented in Node-RED (http://nodered.org) and depicted in Fig. 3. We can configure important RPL parameters from the GUI, including the I_{min}, the $I_{doubling}$ and the selection of the Objective Function. Node-RED flows are wired with JSON messages which pass updated RPL's configuration, compile- or run-time, to the management plane. A live visualization of the outcome illustrates the experiments' progress and its impact on the network.

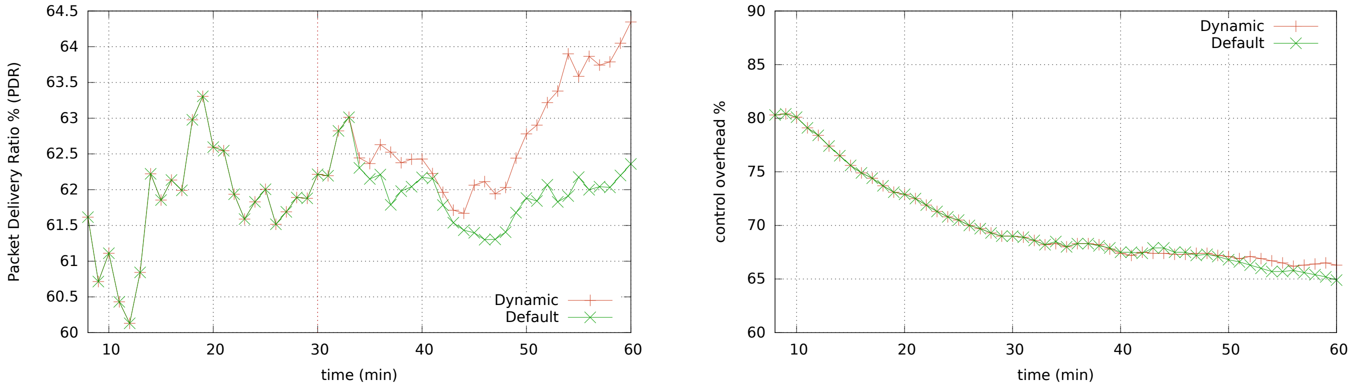
As a proof-of-concept experiment, Fig. 4 depicts the results of running three different scenarios corresponding to respective RPL's configuration settings, namely mixed, dynamic, and default configuration. In line with the use-case scenario presented in Section III and abstractly depicted in Fig. 1a, a mixed-configuration beginning with $I_{min} = 8$ for all motes and then specifying $I_{min} = 12$ for the fixed motes only, achieves constantly higher PDR levels compared to the default configuration, which globally defines $I_{min} = 12$. Furthermore, we observe that the dynamic configuration, which is identical with the default one from $0 \ldots 30\ min$, configures in run-time the fixed motes only, with $I_{min} = 8$, in order to provide connectivity chances for mobile motes, gradually improves PDR. Details of those RPL's configurations are given in Section V.

V. Performance Evaluation

In the following experimentation analysis, we demonstrate the adaptability advantages of the proposed integrated control facility and the adaptable RPL protocol. For example, the CORAL platform is able to change RPL parameters on-the-fly, for all or selective IoT nodes, depending on the network conditions. Our main goal is to overcome the performance limitations of RPL in mobile environments [8].

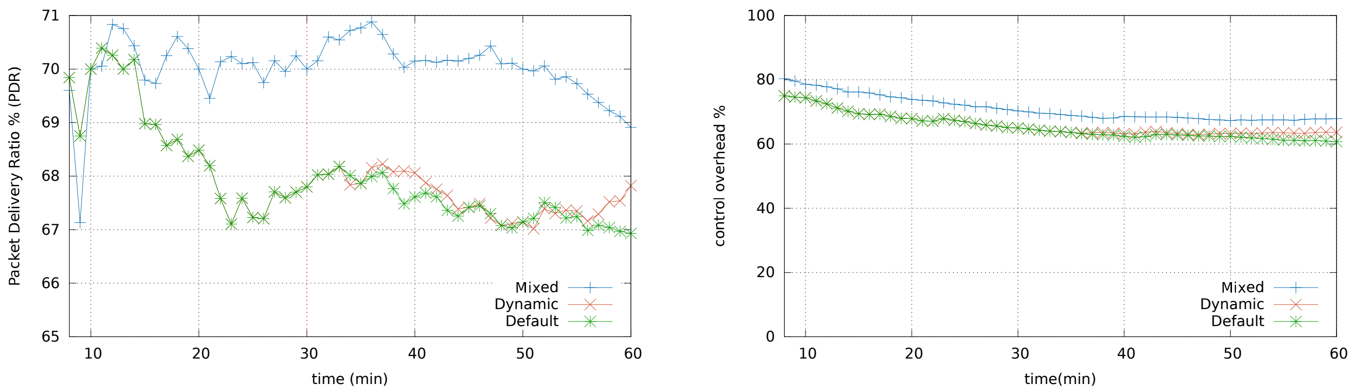
The CORAL can dynamically adapt the RPL protocol to operate efficiently in more complex network deployments. To highlight such capability, we elaborate on our strategy to tackle the performance issues in the use-case of Section III. In the mine scenario after an accident, a critical message has to be delivered even by increasing the network overhead and draining the motes' batteries. After the detection of the emergency situation, the CORAL can adapt dynamically the RPL's trickle timer configuration (i.e., the I_{min} and $I_{doubling}$) to increase the frequency of (re)sending DIO messages, even by sacrificing the overall overhead in the network.

The authors of [9] configure the trickle algorithm with a low maximum DIO interval to make RPL more suitable for mobile nodes. Here, we adopt a similar strategy only for the static nodes to conserve the energy of the mobile ones. Since the fixed nodes can be powered externally, they try more frequently to discover (new) mobile neighbors. The latter maintain their corresponding I_{min} parameters to higher values in order to avoid draining their batteries. We tested successfully the above hypothesis with a wide-range of experimentation setups using the CORAL facility, our adaptable RPL protocol, and the Cooja simulator. We selected experimentation scenarios highlighting the following aspects: (i) the dynamic adaptation of the RPL parameters; (ii) the capability of CORAL to enforce mixed RPL configurations (e.g., different parameters for mobile and fixed nodes); and (iii) the offloading of control overhead between mobile and fixed nodes.

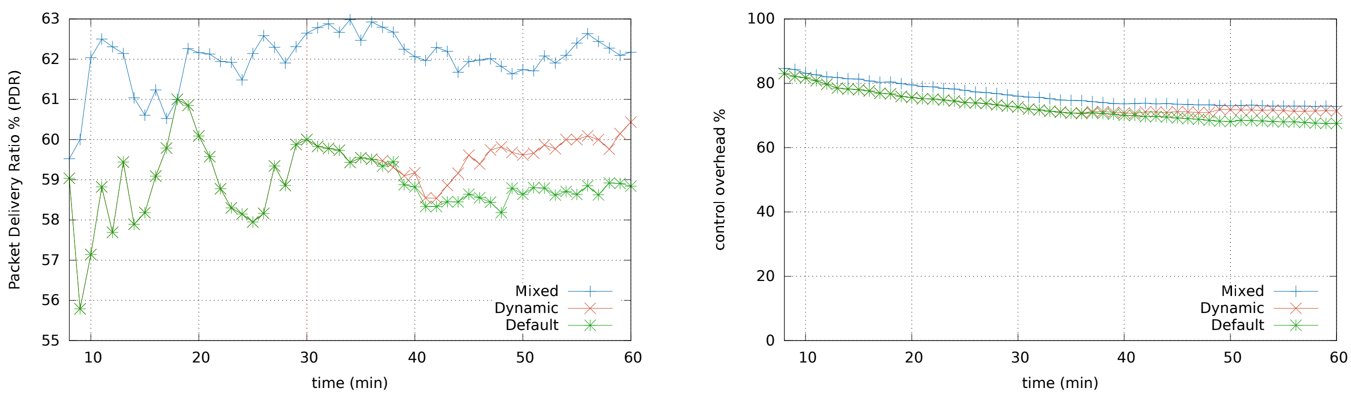


(a) PDR under medium-load traffic (b) Control overhead under medium-load traffic

Fig. 5. Dynamic configuration of RPL



(a) PDR under heavy-load traffic (b) Control overhead under heavy-load traffic



(c) PDR under light-load traffic (d) Control overhead under light-load traffic

Fig. 6. Mixed configuration of RPL

Regarding our methodology, all experiments have been conducted using the Cooja simulator (Contiki v3.0) and a real deployment of the CORAL platform. Experiments involve five mobile and six fixed nodes (including the sink). The duration of each experiment was 60 min. To highlight the involved trade-off between protocol's performance and overhead, we measure the PDR defined as the received ($rUDP$) over the number of sent UDP messages ($sUDP$, size 60 B), i.e., $PDR = rUDP/sUDP$, and the control overhead (OH) as the total control messages (CM) over the summation of CM and $sUDP$, i.e., $OH = CM/(CM + sUDP)$. All scenarios are using the same random 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% to eliminate randomness, and Transmission/Interference ranges were both set to 50 m [10]. We have the regular RPL as a comparison basis. Although we experimented with different combinations of RPL parameters, we focus here on the I_{min} configuration, since it is the most important RPL parameter for the targeted context. Each time the I_{min} value changes, the trickle timer is reset.

To confirm the general applicability of the proposed protocol adjustments, we incorporated into Cooja two mobility models: the first one is based on synthetic, and the second on real mobility traces. More precisely:

a) Synthetic mobility: We used the Cooja add-on [24] to emulate the nodes' mobility based on the random waypoint mobility model [25]. Each mobile node picks a random destination (i.e., x, y coordinates) and a random speed from the uniform distribution $[1.0, 4.0]$ m/sec, moves to the new destination, pauses for a random period uniformly distributed in the range of $[2.0, 10.0]$ sec, and then starts over.

b) Real mobility: The MONROE H2020 EU project [26] provides an 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 (e.g., buses, trains and tracks) in many European cities. We extracted real mobility traces from Stockholm buses, transformed the nodes' GPS coordinates to a 150 $m \times 150$ m canvas and removed the idle times.

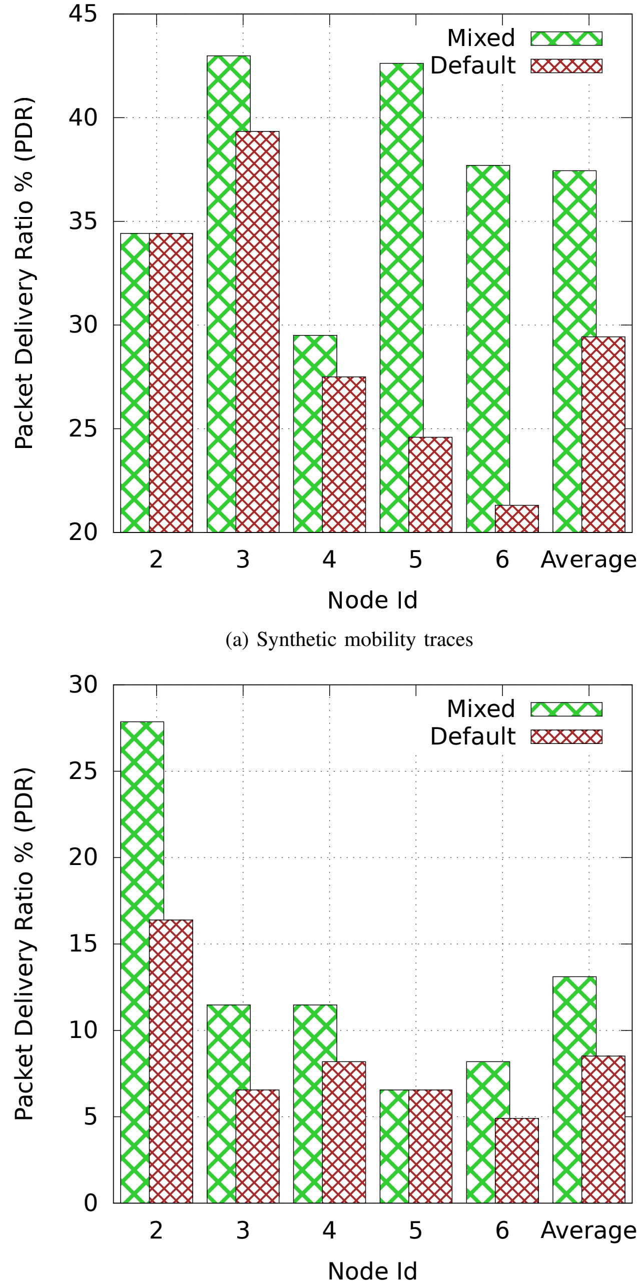
A. Experimental Results

In this subsection, we describe each scenario along with its results and justified outcomes.

a) Dynamic configuration of RPL: The protocol starts with the default RPL parameters and after 30 min, we dynamically change the I_{min} parameter of the sink and fixed nodes to the value 8. We use the synthetic mobility model and implement medium-load traffic, i.e., motes generate 45 $UDPs/min$. Fig. 5a depicts that setting where the $I_{min} = 8$ for the fixed nodes improves the PDR compared to the regular RPL protocol. Such performance increase appears when the I_{min} is changed on the 30 min. As discussed above, in case of an emergency scenario the minor increase in the protocol overhead shown in Fig. 5b, may be anticipated, since it is common that improving the PDR is associated with an increased control overhead [11].

We confirm here that the RPL parameters' adaptation plays an important role in the network's performance, especially in the case of mobile environments.

b) Mixed configuration of RPL: In this scenario, we carried out experiments with both the dynamic and the mixed configuration. We use the same parameters for the dynamic protocol as previously, while in the mixed configuration we adapt $I_{min} = 8$ only for the sink and the fixed nodes from the beginning of the experiment. We use the synthetic mobility model and implement both light- and heavy-load traffic, i.e., motes generate 20 and 60 $UDPs/min$, respectively. Fig. 6a and 6c demonstrate the performance advantages of the mixed configuration in terms of PDR, for both cases of traffic loads. In both scenarios, the performance of the dynamic configuration of RPL starts to converge with the performance of the mixed configuration after the 30 min. In Fig. 6b and 6d, we observe that PDR improvements come with an increase in control overhead, especially in case of mixed configuration and heavy-load traffic. However, such 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 improvements in the CORAL facility, i.e., implementing rapid intelligent detection of the network conditions, considered as a future work.



(b) Real MONROE mobility traces

Fig. 7. Routing performance under mobility

c) Offloading control overhead from the mobile nodes: In the last scenario, we measure the mobile nodes' performance separately to highlight the benefits of offloading the control overhead to the fixed nodes, as in [11]. We use the same configuration parameters with the previous scenario and report the PDR of the mobile nodes, i.e., the motes with $id = \{2, \ldots, 6\}$. Fig. 7a and 7b show the mobile nodes' PDR in case of the aforementioned synthetic and real mobility traces where mixed configuration improves PDR up to 35% in case of synthetic traces, and up to 11% in case of real traces. Actually, such an observation seems independent of the mobility model and was confirmed in a wide-range of scenarios, omitted here due to the lack of space.

To summarize, we confirmed that RPL can cover a wider number of complex IoT network deployments when it is augmented with our centralized control facility.

VI. Conclusions and Future Work

This paper presents a centralized control approach integrating the functionality of the RPL protocol and evolutionarily providing elasticity in routing, since it tackles heterogeneity and mobility in IoT deployments. We extend our CORAL infrastructure with management and control features in order to monitor routing related metrics in network, detect changes in real time, and then enforce appropriate configurations in run-time. Our results confirm that dynamic and mixed RPL configuration can bring improvements in PDR of the order of 35% 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.

The CORAL planes serve as place-holders for further experimentation and improvements. Intelligent detection of network conditions, e.g., mobility or abnormalities, can be accommodated in the management plane to enhance elasticity in the routing function, and are considered as a future work.

Acknowledgments

This work is partially supported by the open call schemes of the Horizon 2020 WiSHFUL (grant agreement no. 645274) and MONROE (grant agreement no. 644399) projects and the H2020 4th EU-BR Collaborative Call NECOS (grant agreement no. 777067) project funded by the European Commission and the Brazilian Ministry of Science, Technology, Innovation, and Communication (MCTIC) through RNP and CTIC.

References

[1] T. Winter, "RPL: IPv6 routing protocol for low-power and lossy networks," IETF RFC 6550, 2012. [Online]. Available: https://tools.ietf.org/html/rfc6550

[2] T. Tsvetkov and A. Klein, "RPL: IPv6 routing protocol for low power and lossy networks," *Network*, vol. 59, 2011.

[3] O. Gaddour and A. Koubâa, "RPL in a nutshell: A survey," *Comput. Networks*, vol. 56, no. 14, pp. 3163–3178, 2012.

[4] G. Violettas, T. Theodorou, S. Petridou, A. Tsioukas, and L. Mamatas, "Demo abstract: An experimentation facility enabling flexible network control for the Internet of Things," in *IEEE Conf. on Comput. Commun. Workshops (INFOCOM WKSHPS)*. IEEE, 2017, pp. 992–993.

[5] T. Theodorou and L. Mamatas, "Software defined topology control strategies for the internet of things," in *2017 IEEE Conf. on Network Function Virtualization and Software Defined Networks (NFVSDN)*, 2017.

[6] ——, "CORAL-SDN: A software-defined networking solution for the Internet of Things," in *Network Function Virtualization and Software Defined Networks (NFV-SDN), 2017 IEEE Conf. on*. IEEE, 2017, pp. 1–2.

[7] B. Mohamed and F. Mohamed, "QoS routing RPL for low power and lossy networks," *Int. J. of Distributed Sensor Networks*, vol. 11, no. 11, 2015.

[8] K. C. Lee, R. Sudhaakar, J. Ning, L. Dai, S. Addepalli, J. Vasseur, and M. Gerla, "A comprehensive evaluation of rpl under mobility," *Int. J. of vehicular technol.*, 2012.

[9] I. El Korbi, M. B. Brahim, C. Adjih, and L. A. Saidane, "Mobility enhanced RPL for wireless sensor networks," in *3rd Int. Conf. on the Network of the Future (NOF)*. IEEE, 2012, pp. 1–8.

[10] C. Cobarzan, J. Montavont, and T. Noel, "Analysis and performance evaluation of RPL under mobility," in *IEEE Symp. on Comput. and Commun. (ISCC)*. IEEE, 2014, pp. 1–6.

[11] C. Cobârzan, J. Montavont, and T. Noel, "Integrating mobility in RPL," in *European Conf. on wireless sensor networks*. Springer, 2015, pp. 135–150.

[12] L. B. Saad and B. Tourancheau, "Sinks mobility strategy in IPv6-based WSNs for network lifetime improvement," in *4th IFIP Intl. Conf. on New Technol., Mobility and Security (NTMS)*. IEEE, 2011, pp. 1–5.

[13] O. Gaddour, A. Koubâa, R. Rangarajan, O. Cheikhrouhou, E. Tovar, and M. Abid, "Co-RPL: RPL routing for mobile low power wireless sensor networks using Corona mechanism," in *9th IEEE Int. Symp. on Industrial Embedded Systems (SIES)*. IEEE, 2014, pp. 200–209.

[14] 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 *Int. Wireless Commun. and Mobile Computing Conf. (IWCMC)*. IEEE, 2016, pp. 678–683.

[15] 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, 2017.

[16] 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 *IEEE Conf. on Comput. Commun. (INFOCOM)*. IEEE, 2015, pp. 513–521.

[17] N. Abdolmaleki, M. Ahmadi, H. T. Malazi, and S. Milardo, "Fuzzy topology discovery protocol for SDN-based wireless sensor networks," *Simulation Modelling Practice and Theory*, vol. 79, pp. 54–68, 2017.

[18] S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat, "Soft-WSN: Software-defined WSN management system for IoT applications," *IEEE Systems J.*, 2016.

[19] C. Jacquenet and M. Boucadair, "A software-defined approach to IoT networking," *ZTE Commun.*, vol. 1, p. 012, 2016.

[20] B. T. de Oliveira, R. C. A. Alves, and C. B. Margi, "Software-defined wireless sensor networks and internet of things standardization synergism," in *IEEE Conf. on Standards for Commun. and Networking (CSCN)*. IEEE, 2015, pp. 60–65.

[21] P. Thubert, T. Watteyne, R. Struik, and M. Richardson, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4," *Working Draft, IETF Secretariat, Internet-Draft draft-ietf-6tisch-architecture-14*, 2018.

[22] P. Ruckebusch, S. Giannoulis, E. De Poorter *et al.*, "A unified radio control architecture for prototyping adaptive wireless protocols," in *European Conf. on Networks and Commun. (EuCNC)*. IEEE, 2016, pp. 58–63.

[23] M. Kulin, C. Fortuna, E. De Poorter, D. Deschrijver, and I. Moerman, "Data-driven design of intelligent wireless networks: An overview and tutorial," *Sensors*, vol. 16, no. 6, p. 790, Jun. 2016.

[24] P. Deshpande, "Mobility of Nodes in Cooja - Contiki," Nov. 2014. [Online]. Available: http://anrg.usc.edu/contiki/index.php/Mobility_of_Nodes_in_Cooja

[25] J. Broch, D. A. Maltz, D. B. Johnson *et al.*, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. of the 4th annual ACM/IEEE Int. Conf. on Mobile Computing and Networking*. ACM, 1998, pp. 85–97.

[26] O. Alay, A. Lutu, D. Ros, R. Garcia, V. Mancuso *et al.*, "MONROE: Measuring mobile broadband networks in Europe," in *Proc. of the IRTF & ISOC Workshop on Research and Applications of Internet Measurements (RAIM)*, 2015.