An Experimentation Facility Enabling Flexible Network Control for the Internet of Things

George Violettas, Tryfon Theodorou, Sophia Petridou, Antonios Tsioukas, Lefteris Mamatas
Department of Applied Informatics
University of Macedonia, Thessaloniki, Greece
Email: {georgevio, tryfonthe, tsioukas}@gmail.com, {spetrido,emamatas}@uom.edu.gr

***Abstract*—We present a showcase of our SDN-inspired facility for the Internet of Things (IoT) - the CORAL (Cross-Layer Control of Data Flows) infrastructure. CORAL provides a hierarchy of network controllers that monitor network behaviour and adapts network protocol parameters on demand. It is built on top of rich network control abstractions and is hardware-independent. Our facility offloads protocol functionality to the controllers, enabling intelligent IoT protocol adaptations. To demonstrate our solution, we focused on the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL). Although RPL is the state-of-the-art routing protocol for constrained devices and lossy networks, it is inefficient in mobile environments. We show a particular example where, using CORAL, we improve the Packet Delivery Ratio (PDR) in a dynamic IoT topology.**

I. Introduction

The Internet of Things (IoT) is an important technological area characterized by an explosive growth, changing our world rapidly. IoT systems and devices define a huge area of innovation, allowing people to develop their own designs and products, even at home. Furthermore, the Software-Defined Networks (SDNs) and their successful implementations for infrastructure networks gradually expand to new environments, such as the Wireless Sensor Networks (WSNs), that bring aesthesia in the IoT. The integration of SDNs and IoT provides new perspectives and ground for applications and research.

Investigating the above technologies is challenging, since: (i) the usage and configuration of IoT devices requires hardware specific knowledge; (ii) the solutions integrating SDNs with IoT are very limited or under development, because, they do not consider fundamental characteristics of wireless networks with mobile devices, such as mobility and radio signal issues; and (iii) there is a shortage of realistic experimentation environments. To address the aforementioned challenges, we implemented the CORAL test-bed for the IoT, which decouples complexity from the network protocols and offloads it to network control software deployed at the surrounding fixed infrastructure.

CORAL enables experimentation with SDN-inspired capabilities aiming at improved applications' performance over resource-constrained devices. It experiments with novel network control features and protocols that: (i) realize optimized routing over mobile devices with signal issues and intermittent connectivity; and (ii) improve resource allocation and energy consumption of mobile devices. CORAL builds up on the WiSHFUL infrastructure (http://www.wishful-project.eu/), a novel experimentation facility providing the required software and hardware experimentation capabilities, including the appropriate radio- and network-control abstractions over heterogeneous wireless environments, called Unified Programming Interfaces (UPIs).

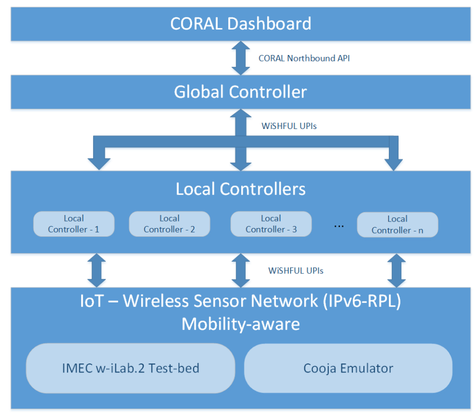

Fig. 1. The CORAL Architecture

II. Test-Bed Architecture

Our goal is to implement intelligent global and local control loops (i.e., global for the whole network and local near a small set of nodes) that monitor network behaviour and adjust, via the UPIs, a rich set of network protocol parameters. This way we can have a flexible network that adapts to dynamic conditions, while considering nodes' heterogeneity (e.g., through local configuration). In practice, our facility changes network protocol parameters dynamically and monitors network's metrics, as shown in Section III. Furthermore, we are in the process to enhance CORAL with intelligence by adopting the Link Quality Estimation (LQI) algorithm discussed in [1].

Fig. 1 demonstrates the CORAL test-bed architecture, which is divided into the following four modules:

- The *CORAL Dashboard* provides a highly flexible GUI, implemented with NODE-RED (i.e., https://nodered.org). It performs the overall monitoring and system management, while providing advanced functionalities and configuration options.
- The *Global Controller* module acts as the hub between the CORAL Dashboard and the rest of the system. It is

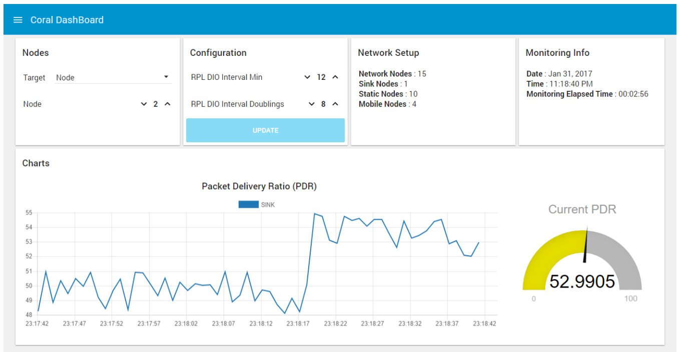

Fig. 2. The CORAL dashboard shows the impact of the dynamic RPL configuration in real-time

responsible for the setup, configuration, data collection and investigation of the experiment in real time. In order to communicate with the lower modules it uses the WiSHFUL software platform (https://github.com/wishful-project).

- The *Local Controller* module consists of a set of sub-modules, coordinated by the *Global Controller*, being responsible for monitoring and management of each mote separately. Controllers are communicating with the motes either with the COAP protocol or via serial ports.
- The *IoT WSN test-bed* module is the basis for conducting the experiment and is designed to support either real motes from the IMEC w-iLab.2 test-bed (http://wilab2.ilabt.iminds.be) or emulated motes using Cooja. It operates using the IEEE 802.15.4 MAC protocol and the IPv6 network layer implementing the RPL routing protocol.

The CORAL architecture coincides with the SDN paradigm, i.e., decoupling the data from the control plane. For the data plane, we use a WSN with Contiki-OS motes using IPv6 network stack, while for the control plane, we have a highly adaptable prototype framework that implements a controller hierarchy. Such facility performs a dual action; firstly, it updates, in real time, critical configuration parameters either in the whole network or in each mote separately, and, secondly, it collects and processes the experimental results.

III. Demo Description

To demonstrate the research potential of the CORAL infrastructure, we experimented with our novel SDN-inspired network control features to fine-tune the RPL protocol. RPL organizes nodes as a Destination-Oriented Directed Acyclic Graph (DODAG) rooted at a single destination called root or sink. The DODAG's maintance is placed at the very core of the RPL's functionality and, hence, a dedicated algorithm, the trickle timer, synchronizes the propagation of topology discovery messages (called DIOs) upon which, the graph's convergence is based. The critical aspect in this process is to achieve a short period of the graph's setup time and, thus, reinforcing network's metrics, e.g., the PDR, while restricting the routing traffic overhead to save node's energy consumption.

Although RPL is the state-of-the-art routing protocol for the network under consideration, it is inefficient in mobile environments [2]–[4]. In practice, when a mobile sensor node moves out of its parent's range, it is getting disconnected by the graph affecting the routing process and, consequently, the network's performance. Recent proposals tackle the RPL under mobility [2]–[4] through handling the interval time for the different RPL topology control messages, which is defined by the trickle algorithm's parameters (i.e., I_{min}, $I_{doubling}$). In contrast to these works, CORAL can change these parameters both dynamically (i.e., during the experiment) and individually (i.e., nodes are grouped according to their mobility behaviour and assigned different parameters' values).

Our experiment considers 15 RM090 motes (i.e., the root, ten static and four mobile nodes) with IEEE 802.15.4, $2.4GHz$ wireless connectivity, which are located in the IMEC w.iLab.2 test-bed in Ghent, Belgium. For the experiment we also use Cooja emulated motes with a firmware written in C, that implements the IPv6 network stack using the RPL protocol and an application layer protocol sending UDP messages to the root. The experiment is controlled via the CORAL Dashboard GUI, as depicted in Fig. 2, which allows the user to send real time configuration updates to the nodes. In practice, the user selects a node or all nodes and then defines a configuration parameter to be applied to the network protocols. In the showcase of Fig. 2, we change the RPL DIO interval min (I_{min}), which is expected to affect the PDR. Our configuration parameter is collected from the Global Controller, implemented in Java, and dispatched to the WiSHFUL infrastructure using UPIs. Commands in UPIs can also be delivered to the nodes via the Local Controller who is responsible for specific node(s). Once the selected node adapts its behaviour in real time according to the parameter received, it sends a notification to the controller. The control data are transferred using an alternate communication channel in order to keep the experiment unbiased from control information. The graph in Fig. 2 depicts that we succeed to improve the network's PDR by dynamically configuring node 2.

Acknowledgment

This work is partially supported by the open call schemes of the Horizon 2020 WiSHFUL (grant agreement number 645274) and MONROE (grant agreement number 644399) projects.

References

[1] T. Liu and A. E. Cerpa, "Foresee (4c): Wireless link prediction using link features," in *10th Int. Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2011, pp. 294–305.

[2] K. C. Lee, R. Sudhaakar, L. Dai, S. Addepalli, and M. Gerla, "Rpl under mobility," in *Consumer Communications and Networking Conference (CCNC)*. IEEE, 2012, pp. 300–304.

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

[4] C. Cobarzan, J. Montavont, and T. Noel, "Analysis and performance evaluation of rpl under mobility," in *IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2014, pp. 1–6.