Demo: Virtualized Wireless Sensor Networks for Distributed Applications over the Cloud Continuum

Adriana Arteaga-Arce, Alexandre Veremme, Carol Habib, Nathalie Mitton

Demo: Virtualized Wireless Sensor Networks for Distributed Applications over the Cloud Continuum

Adriana Arteaga-Arce, Alexandre Veremme, Carol Habib, Nathalie Mitton
Inria, France - Email: {firstname.lastname}@inria.fr

***Abstract*—In this demo, we present the implementation of a virtualized Wireless Sensor Network following a specific software stack called Virtual Object Stack (VOStack). This stack, proposed by the European Horizon NEPHELE programme, promotes openness and interoperability in the cloud-to-edge-to-IoT continuum. The demonstration shows a wireless sensor monitoring application distributed along the continuum, validating the end-to-end communication among FiPy-based sensor nodes and a web application through the virtual object of a Wireless Sensor Network Gateway.**

***Index Terms*—Cloud Continuum, Virtual Object, Web of Things, Wireless Sensor Network, NEPHELE project.**

I. Introduction

Wireless Sensor Networks (WSNs) have been extensively integrated to enhance context awareness across various Internet of Things (IoT) applications, including industry, healthcare, environmental monitoring, and disaster response. A WSN is formed by sensor nodes running on low-cost and energy-constrained devices with sensing, processing, and wireless communication capabilities to send data to a gateway. The wireless sensor network gateway (WSNG) is a more robust device than a sensor node, so it could act as the network's hub for processing data collected by the sensor nodes. Using a virtual object (VO) of IoT devices allows the deployment of the virtual instance on the Edge/Cloud infrastructure, extending the capabilities of resource-constrained devices by exploiting resources on the computing continuum. Additionally, the abstraction of the physical components through a virtual instance provides flexibility for supporting functionalities such as distributed data management, autonomic networking, and time-triggered IoT functions in dynamic WSN.

In this demo, we showcase the implementation of a distributed application to support a virtualized Wireless Sensor Network using a specialized IoT and edge computing software stack called VOStack [1]. This implementation is part of one of the use cases in the NEPHELE programme to test and validate the VOStack to enhance the situational awareness in the post-disaster recovery scenario [2].

II. Implementation

Figure 1 represents the architecture of our scenario, with components and layers along the continuum. The sensor nodes and the WSNG are in the device layer and communicate using MQTT protocol. The sensor nodes are deployed on FiPy boards [3], and support multiple communication technologies including WiFi, Bluetooth, LTE, Lora, and Sigfox. The boards

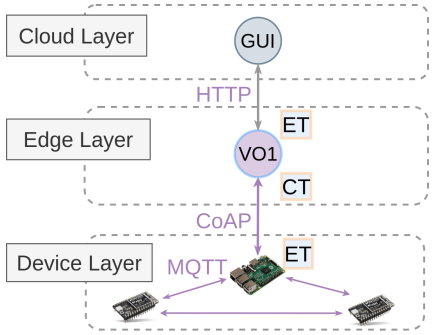

Fig. 1: Application graph for the implemented scenario

are equipped with multiple sensors for measuring acceleration, light intensity, pressure, temperature, and humidity. The WSNG operates on a well-known Raspberry Pi 3 [4].

The WSNG has its VO deployed in the edge layer; the physical device and its virtual instance communicate using CoAP protocol. The VO uses two interfaces called the Exposed Thing (ET) and the Consumed Thing (CT) for communicating with the physical device and other components in the architecture; more details about these elements are presented in Section II-B. Finally, a simple Graphical User Interface (GUI) displays the information retrieved from the VO using HTTP protocol.

A. Wireless Sensor Network

We have developed a MicroPython library dedicated to the sensor life cycle to reduce resource consumption whilst respecting external constraints, effectively balancing power and performance. According to the sensor's workflow presented in Figure 2, the sensor's wake-up frequency matches by definition the data transmission frequency. The sensor wakes up, establishes WiFi connection and starts the MQTT server. After, the sensor gathers data, and then transmits it via MQTT to the WSNG. Following transmission, the sensor reverts to a power-saving sleep mode, repeating the cycle at a predetermined frequency.

Owing to the simplicity and power of MQTT, there is a bidirectional communication using multiple topics: *i)* a *config topic* which maintains a single retained message to list sensor nodes permitted to initiate and transmit to the WSNG. Using this topic, at each start/wake-up, any sensor node can be disabled, and its entire configuration can be dynamically adjusted; and *ii)* an *output topic* for each sensor node to publish measurements. These measurements are stored in a MQTT-retained JSON-typed message.

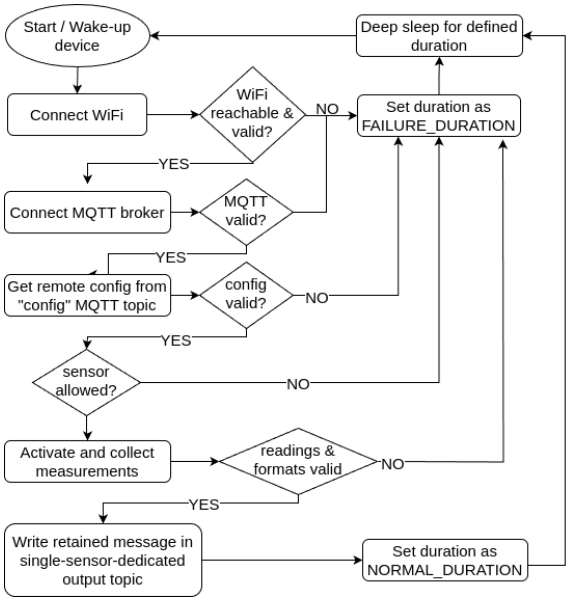

Fig. 2: Workflow of the implemented sensor node's life-cycle

B. Virtualized WSN gateway

To facilitate the WSN integration into NEPHELE platform, we propose to virtualize the WSNG and use this virtual instance to provide all the information and functionalities related to the WSN [2]. We define the *Thing Descriptor* (TD) and the VO files to enable communication between the gateway and its VO; based on the common concept of a *Servient* defined in the Web of Things and also implemented in the VOStack. A *Servient* is a software block capable of hosting and exposing Things and it offers two different functionalities: *i)* the Exposed Thing (ET) that represents a locally hosted Thing that can be accessed over the network by remote Consumers, and *ii)* a Consumed Thing (CT) that represents a remote Thing used by a local application. As a result, the VO consumes the Exposed Thing of the WSNG using CoAP protocol.

The VO runs generic functions to print its own status and the status of devices it is consuming. We also use two periodic functions: *i)* to read the sensor node data exposed as a property by the WSNG and store them in an Influx Database, *ii)* to calculate the average temperature and humidity over five samples. In addition, the VO emits an event each time it detects a battery level lower than a defined limit (eg.: 30%).

III. Demo description

We have tested the end-to-end communication from the physical sensor node to the web application passing by the virtualized WSNG using the VOStack. To run the demo, we deployed a sensor node on a FiPy board and the WSNG on a Raspberry Pi 3 board. The MQTT broker is hosted in the cloud whereas the CoAP server is hosted on the gateway (i.e. the Raspberry), and the CoAP client is at the VO side.

A light web application was designed to consume the VO's Expose Thing and to display data collected by the sensor nodes. This web application is developed with the Python-based Flask framework and uses usual web stack (i.e., HTML, CSS and Javascript). This web application consumes the sensor node data exposed by the WSNG, and displays plots dynamically and periodically updating with new readings as is shown in Figure 3. The VO and the web application run on distinct computers. So far, the Raspberry Pi and both computers are in the same local area network. By that, the end-to-end communication from the physical device to the web application through the VO is tested and validated [1].

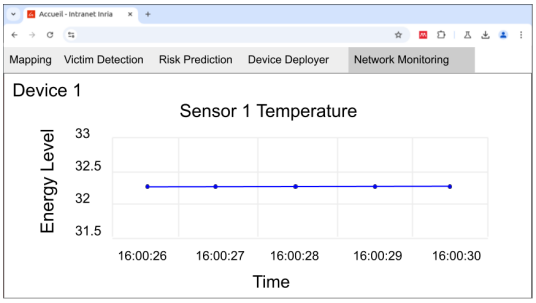

Fig. 3: Graphical User Interface

IV. Conclusions and Future work

In this demo, we have presented an implementation of a virtualized Wireless Sensor Network using the VOStack, the specialized IoT and edge computing software stack proposed in NEPHELE programme. We have described the integration of multiple communication protocols to provide a flexible end-to-end communication from IoT devices to a web application passing by a virtual object. In the future work, we intent to integrate a routing protocol to enable multi-hop communication and extend the WSN coverage range. We intend to provide additional periodic functions to calculates important networking metrics such as latency, throughput, and energy consumption.

Acknowledgment

This work has been supported by the European Union's Horizon Europe research and innovation program under grant agreement No. 101070487 (NEPHELE).

References

[1] A. Zafeiropoulos, E. Fotopoulou, C. Vassilakis, I. Tzanettis, C. Lombardo, A. Carrega, and R. Bruschi, "Intent-driven distributed applications management over compute and network resources in the computing continuum," in *2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2023, pp. 429–436.

[2] A. Arteaga, N. Filinis, L. Fu, C. Habib, L. Militano, D. Spatharakis, A. Zafeiropoulos, T. Bohnert, N. Mitton, and S. Papavassiliou, "Virtual Objects for Robots and Sensor Nodes in Distributed Applications over the Cloud Continuum," in *Distributed Applications over the Cloud Continuum. International Workshop on Distributed Intelligent Systems (DistInSys)*, Jun 2024, Paris, France.

[3] "Fipy board," https://docs.pycom.io/datasheets/development/fipy/, accessed: 2024-08-01.

[4] "Raspberry pi," https://www.raspberrypi.com/documentation/, accessed: 2024-08-01.

[1] A demonstration video reporting our implementation and validation results can be found at the following link: https://youtu.be/2FwyxgTdDvw