DEMO: A Cloud-based Virtual Network Operator for Managing Multimodal LPWANs and Devices

Bart Moons*, J. Haxhibeqiri*, M. Van Eeghem*, J. Rossey*, A. Karagaac*, J. Famaey†, and J. Hoebeke*
* IDLab, Ghent University – imec, Ghent, Belgium. † IDLab, University of Antwerp – imec, Antwerp, Belgium
Email: bamoons.moons@ugent.be

Abstract—The Internet of Things (IoT) has given rise to different sub-GHz Low Power Area Network (LPWAN) technologies, each having specific characteristics. Today, devices can support multiple LPWAN technologies, picking the best technology depending on application requirements and availability. Realizing such multimodal communication introduces complexity at both the device and network side. In this demo, we show how we can control and manage multimodal LPWA networks by a Virtual Network Operator (VNO) using unified control APIs based on the OMA Lightweight M2M protocol. On top, we demonstrate the use of a single stack across different LPWANs by implementing an adaptation layer between the IP and MAC layer and the feasibility to achieve real multimodal communication.

I. INTRODUCTION

Recent years have shown the rise of different LPWAN technologies such as LoRa, DASH-7 and Sigfox, which are used in the IoT for different settings and applications. In order to make use of multiple technologies, the end nodes can be equipped with support for multiple radio technologies. Such multimodal capabilities introduce complexity in the management of the involved LPWANs with heterogeneous APIs and designing applications on top. Our demo shows a VNO that homogenizes network management by means of distributed Docker containers that expose unified OMA LwM2M [1] API's to control and manage the network equipment. Using Static Context Header Compression (SCHC), a single IPv6-based stack is used for efficient end-to-end communication independent of the underlying LPWA technology. In section II we will describe the system design while in III we show step by step how we use the unified API's to configure the gateways and establish end-to-end IPv6 communication.

II. DEMO SYSTEM OVERVIEW

The system overview of the demonstrator is given in Figure 1 and is divided in three parts. The leftmost part shows the multimodal device, having support for both DASH-7 and LoRaWAN. At the center, the used LPWA network equipment is shown which consists of a DASH-7 gateway, a LoRa gateway and a LoRaWAN Network Server (NS). The DASH-7 gateway consists of a modem for packet reception and a Raspberry Pi for packet forwarding. At the right side, the Virtual Network Operator (VNO) architecture is shown, divided in a control/management plane and a data plane.

The OMA LwM2M protocol [1] is used by the control plane to manage and configure all LPWAN equipment. To achieve this, the equipment is turned into LwM2M compliant devices

by means of a Docker container that converts the domestic APIs into LwM2M APIs. Inside the container, we have the technology specific adapter part, whereas the Virtual Device Manager (VDM) and the LwM2M Client are generic across LPWAN equipment. The Adapter consumes the custom API's to interact with the equipment. The VDM makes a bridge between the LwM2M client and the custom adapter, while the LwM2M client exposes all configurable parameters of the LPWAN equipment as LwM2M resources. For that purpose, novel LwM2M data models have been foreseen. At the VNO side, the control plane consists of LwM2M server that can interact with each of the LwM2M clients in all the LPWAN equipments.

For realizing the data plane at the VNO, we foresee one Docker container per LPWAN deployment that is able to exchange data with that network. As the LoRa NS has an on-board MQTT broker, the container will consist of an MQTT client. Data packets received by the LoRa gateway are published to the LoRa MQTT broker on the LoRa Network Server and received by the LoRa MQTT client inside the Docker container. The DASH-7 gateways are equipped with an MQTT client that is running on the Raspberry Pi, so there can be a single, central broker in the container to support bi-directional communication over DASH-7. Next to this technology specific part, the container also has a generic part that offers a uniform way to pass the data for further processing, independent of the LPWAN technology being used. Here, we have opted to use MQTT again, where data is handed over to a Click server via a broker [2].

With these components, it is possible to bidirectionally exchange data. Next, in order to only have a single stack across the different LPWAN technologies being used, we have chosen to use a CoAP-UDP-IPV6 stack in combination with a custom SCHC adaptation layer [3] between layer 2 and 3. This layer can efficiently compress the packets, dealing with the limited packets sizes of LPWANs. At the VNO side, the click server acts as the central data management unit, handling the (de)compression, the (de)fragmentation and the connection to the IPv6 network. At the end device side, an embedded counterpart of this tack is foreseen. These devices are B-L072Z-LRWAN development kits powered by an STM32L072CZ MCU and a SX1276 transceiver. They run the Open Source Stack implementation of the DASH-7 alliance protocol (OSS-7) [4], which supports run-time reconfiguration of the physical layer as well as a full OSI-stack implementation. Our setup,
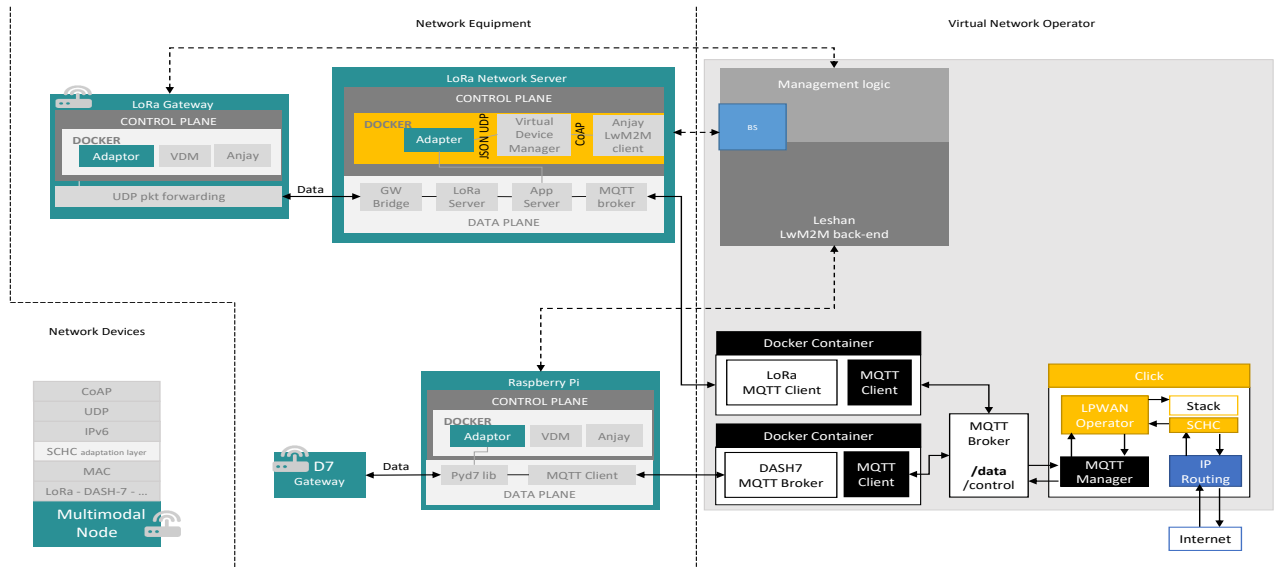
Fig. 1. Demo system overview.

however, only requires the use of the PHY- and MAC-layer. On top, a CoAP - UDP - IPv6 stack is used, with the custom SCHC adaptation layer.

III. DEMONSTRATION

Our demonstration shows how different LPWAN equipment can be controlled and configured through a cloud-based Virtual Network Operator (VNO). In this demo we use two different sub 1 GHz technologies: LoRa and DASH-7.

The first step demonstrates the roll-out of new LPWA network equipment that needs to be managed by the VNO. The LwM2M compliant LoRa equipment is deployed and automatically discovered by the VNO. Using the LwM2M data models, the configuration is performed by the VNO such that data can be exchanged between the new LoRa deployment and the VNO.

Once the data plane parameters have been configured in the LoRa gateway and NS, devices can be registered to the LoRa NS. The registration can be done by passing the AppEUI, DevEUI and APPKey of the end node to the LoRa NS when over-the-air-activation is used.

From this point on, the device can start exchanging data packets with the LoRa NS. A packet sent by the device is compressed by the SCHC adaptation layer. After compression, the device will configure the radio to send packets with Chirp Spread Spectrum (CSS) modulation, so it can reach the LoRa gateway. Once a packet has been picked up by the LoRa gateway it is forwarded to the LoRa NS and will eventually reach the VNO over MQTT. There, the received packet will be decompressed and forwarded towards the correct client in the IPv6 network.

Next, a device from the IPv6 network will try to send a large packet to the device, which only has an active LoRa connection

using SF12, and thus only able to receive 51 bytes. The VNO will therefore wait for a better connection to pop up. For this, the DASH-7 communication is configured by setting up the correct data plane parameters (MQTT broker IP and MQTT broker topics) on the DASH-7 gateway using the APIs and the LwM2M server at the VNO.

Once the network has been set up, the device can make use of this higher bandwidth network. The device switches from CSS to 2-GFSK modulation and is being configured to send DASH-7 packets. Once such a packet is picked up by the DASH-7 gateway, it will be sent to an MQTT broker inside a Docker container towards the VNO. The network operator now knows that the device is connected to this network and will send the large, pending packet, which could not be sent over LoRa.

REFERENCES

[1] Open Mobile Alliance, "Lightweight Machine to Machine Technical Specification," February 2017.
[2] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Trans. Comput. Syst.*, vol. 18, no. 3, pp. 263–297, Aug. 2000.
[3] A. Minaburo, Acklio, L. Toutain, and C. Gomez, "Lpwan static context header compression (schc) and fragmentation for ipv6 and udp," Internet draft, IETF, 2017.
[4] (Last visited 24/4/2018) The OSS-7 website. [Online]. Available: http://mosaic-lopow.github.io/dash7-ap-open-source-stack/