# A Cloud-based Virtual Network Operator for Managing Multimodal LPWA Networks and Devices

Jeroen Hoebeke*, Jetmir Haxhibeqiri*, Bart Moons*, Matthias Van Eeghem*, Jen Rossey*,
Abdulkadir Karagaac*, and Jeroen Famaey†
* IDLab, Ghent University – imec, Ghent, Belgium
† IDLab, University of Antwerp – imec, Antwerp, Belgium
Email: jeroen.hoebeke@ugent.be

*Abstract*—**The Internet of Things (IoT) domain is characterized by many applications that require low bandwidth communication over a long range at a low cost and at low power, which has given rise to novel Low Power Wide Area Network (LPWAN) technologies that operate in the sub-GHz domain. Today, these technologies are being adopted in more complex settings and configurations than originally intended. Devices are being equipped with multiple LPWAN radio technologies to satisfy more diverse requirements, connecting to different networks at different times and locations. Further, organizations can have devices with different LPWAN technologies in the field or existing infrastructure might be shared across different organizations. As a consequence, there arises an increasing complexity in managing such multimodal LPWANs and in designing IoT applications on top. We present and validate the concept of a cloud-based virtual LPWA network operator that unifies such multimodal communications and homogenizes the way such heterogeneous private LPWAN deployments are managed by means of distributed OMA Lightweight M2M (LwM2M) containers. In addition, our architecture provides support for the most recent efforts in adopting IPv6 for LPWAN communication.**

## I. Introduction

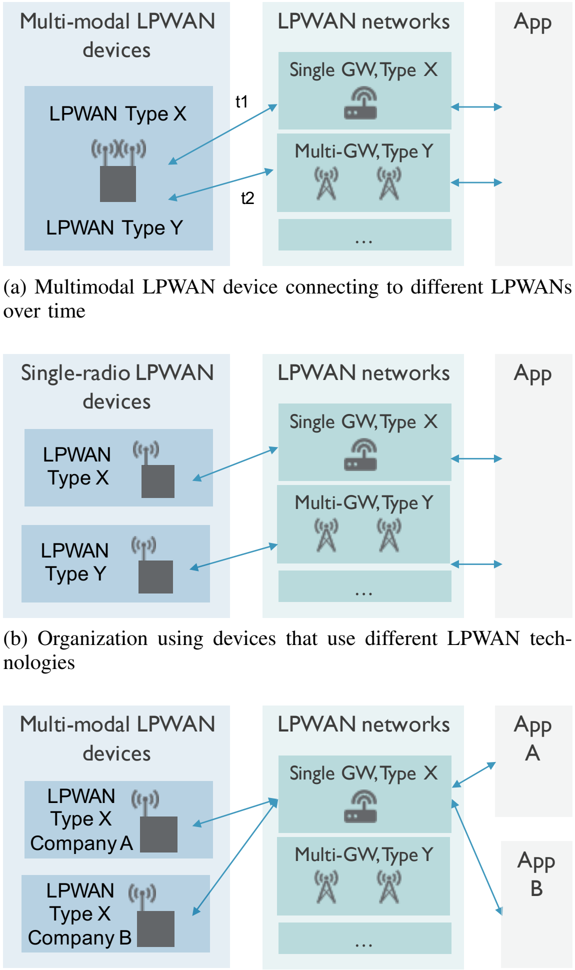
Many Internet of Things (IoT) applications are characterized by low-bandwidth communications, at a low cost and at low power. To address these needs, several novel wireless standards using sub-1GHz frequencies have recently been proposed. They are able to operate over long distances, bringing down the network infrastructure cost, at the expense of a reduced bit rate. Currently, several sub-1GHz technologies are being promoted simultaneously, all of which use the same sub-1GHz wireless spectrum. The most popular initiatives in this domain are LoRa, SigFox, IEEE 802.15.4g and the upcoming IEEE 802.11ah standard, and are often referred to as Low-Power Wide-Area Networks (LPWANs or LPWA networks) [1]. Each technology has its own strengths and weaknesses in terms of coverage, energy consumption and throughput. For simple IoT applications, one can suffice with picking the most suitable LPWAN technology, deploying single-radio IoT devices of that technology and building a vertical application where the IoT application is knowledgeable about the underlying LPWAN specifics.

However, a single technology is not flexible enough to satisfy the requirements of more demanding or diverse IoT applications. As a consequence, we see scenarios where these LPWAN technologies are being adopted in more complex settings and configurations than originally intended. Devices are being equipped with more than one LPWAN radio technology to satisfy more diverse requirements, such as long range outdoor communication combined with higher data rate indoor communication. Such multimodal devices connect to different networks at different times and locations, as is illustrated in Figure 1a. Further, IoT use cases may involve the collection of heterogeneous data, requiring the deployment of single-radio devices that use different LPWAN technologies, as shown in Figure 1b. Last but not least, the fact that LWPAN equipment can cover large areas, makes it interesting to share the same infrastructure across different organizations, as is illustrated in Figure 1c.

The above trends and observations lead to an increasing complexity in managing such multimodal LPWANs and in designing IoT applications on top. In order to deal with this increasing complexity, this paper introduces and validates the concept of a cloud-based virtual LPWA network operator. The operator homogenizes the way such heterogeneous LPWAN deployments are managed (control plane) and ensures that end-to-end communication with multimodal LPWAN devices or interactions with heterogeneous LPWAN device is unified (data plane), independent of the details of the underlying LPWAN technologies being used.

To our knowledge, this is the first paper to present such a concept. Further, we also make following additional contributions: (i) the use of the OMA Lightweight Machine-to-Machine (LwM2M) protocol to build the virtual network operator control plane, (ii) a generic and easy-to-deploy method to convert LPWA network equipment into LwM2M compliant devices, (iii) novel LwM2M objects for LPWAN management, (iv) the use of Static Context Header Compression (SCHC) for multimodal communication to unify the data plane and (v) an initial validation of both the control and the data plane.

The remainder of the paper is organized as follows. Section II gives an overview of related work on multimodal



(a) Multimodal LPWAN device connecting to different LPWANs over time

(b) Organization using devices that use different LPWAN technologies

(c) Organizations of which the LPWAN devices share the same LPWAN infrastructure

Fig. 1. Use cases enabled by multimodal LPWANs and devices.

LPWA devices, networks and their management. A high-level overview of the Virtual Network Operator concept is presented in Section III, followed by a detailed presentation of the realization of the control plane and data plane in Sections IV and V respectively. Section VI validates the architecture, following by conclusions in Section VII.

## II. Related Work

LPWAN radios have only been around for a few years and have already been complemented with shorter range technologies such as Wi-Fi and BLE within a single device [2]. Nowadays, there is also trend towards multimodal LPWAN devices, combining different LPWAN types, in order to get the best of different worlds. As an example, the electronics company Murata announced dual mode LoRa/SigFox modules [3], that can even support modulation types used by other LPWAN technologies, paving the way to the provisioning of multi-technology LPWANs that can be perceived as a single logical virtual network. To perform such adaptations at the infrastructure level of multiple, heterogeneous LPWANs and to enact necessary (re)configurations, there is a need for technology-independent solutions that allow configuration of resource usage and settings on this LPWAN infrastructure. Hereby, we currently consider LPWANs of different technologies that jointly offer services to connected LPWAN devices, although coexistence of these technologies on the same hardware might also be considered for the infrastructure side, as has been demonstrated in [4] for shorter range technologies. Today, research on the management and virtualization of multimodal LPWANs is non-existent. Ongoing research efforts involving multiple LPWAN technologies mainly focus on coexistence aspects, such as [5] and [6]. They highlight the negative impact of uncoordinated LPWANs on each other. To mitigate this, joint management and coordination of such multimodal LPWANs is needed, which is only possible when having the necessary control elements, for which our work lays the foundation. Looking at existing research works on managing and using heterogeneous networks, we end up in the Wi-Fi/LTE world, where efforts have taken place to virtualize these networks, including slicing of the network by sharing and isolating infrastructural resources and simplifying the configuration and management [7] [8]. Decoupling of the control plane and data plane is here a key concept, but is not considered for LPWANs. At the higher layers, unified communication for such multimodal Wi-Fi/LTE devices is achieved by leveraging on the Internet Protocol and higher layer protocols that are able to deal with the availability of multiple paths, such as Multipath TCP [9]. Such approaches cannot easily be transferred to LPWAN devices due to their typical constraints in processing power and memory. As such, both the control plane and data plane require further study in order to come to multimodal LPWANs that can be efficiently managed and appear as a single logical network to the IoT applications on top.

## III. Cloud-Based Virtual Network Operator Concept

As said, we see a trend towards both multimodal LPWAN devices and IoT applications involving data coming from a device using different LPWAN technologies. On top of that, LPWA network infrastructure might be virtualized and reused across organizations. In order to deal with this increasing complexity, we propose the concept of a cloud-based virtual LPWA network operator. Such an operator homogenizes the way such heterogeneous LPWAN deployments are managed and ensures that end-to-end communication is unified, independent of the details of the underlying LPWAN technology being used. The high-level architecture of such a virtual network operator (VNO) for multimodal LPWANs is shown in Figure 2 and can be broken down in 4 main building blocks.

The *control plane* of the VNO is responsible for discovering, properly configuring and monitoring all LPWA network equipment that it manages. This involves the addition of new LPWANs to be managed, including gateways or other components such as a Network Server for LoRaWAN networks, and ensuring that data can be exchanged between the devices

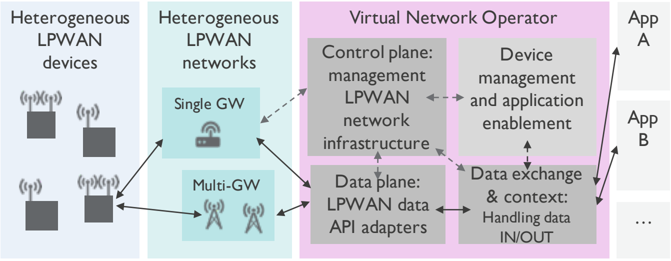
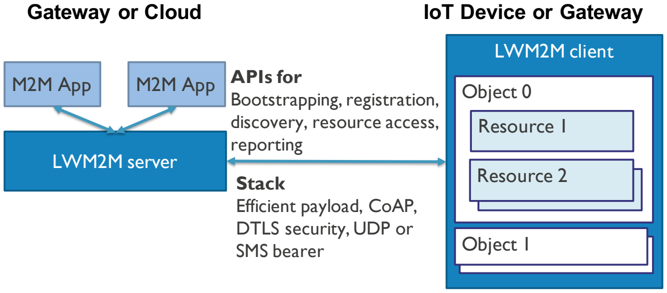

Heterogeneous LPWAN devices
Heterogeneous LPWAN networks
Single GW
Multi-GW
Virtual Network Operator
Control plane: management LPWAN network infrastructure
Device management and application enablement
Data plane: LPWAN data API adapters
Data exchange & context: Handling data IN/OUT
App A
App B
...

Fig. 2. High-level breakdown of the components of the Virtual Network Operator. Solid arrows illustrate a sample data flow, whereas dashed arrows represent a sample control plane flow dealing with either LPWA network equipment management or end device management.



E.g. CoAP PUT with payload 'on' to URI /3311/0/5850 → Turning 1st instance (Instance ID=0) of light control object (Object ID=3311) on (Resource ID=5850)

Fig. 3. LwM2M principles: APIs and data model

and the VNO over these networks (i.e. configuring the data plane). The control plane is also responsible for defining which LPWAN devices are allowed to make use of which networks. Last but not least, it also has monitoring responsibilities, following up on the state of the components and collecting relevant statistics.

The *data plane* is responsible for the actual data exchange. both in uplink and downlink, over the different LPWANs being managed. How the data is exchanged between the VNO and the LPWA network equipment depends on the LPWAN technology under consideration as well as on implementation specifics at the equipment side. As such, it requires proper configuration by the control plane.

The data arriving at the data plane is then handed over to the *data exchange and context* block of the VNO, which further processes the incoming data. Relevant contextual information is derived (e.g., the network and gateway(s) over which the data arrived, the signal strength, etc.), in order to be used for taking decisions on downlink communication. Finally, the data might be transformed into a format that is fully LPWAN technology independent and suitable for delivery to either the end application (not part of the VNO) or to the optional *device management and application enablement* component of the VNO in case the VNO also takes care of device management and application layer data processing for the devices. For downlink data that must be sent to a device, the *data exchange and context* block will take the forwarding decision, i.e. which available LPWAN technology to use and when, upon which the data will be delivered to the data plane for further transmission.

In the following two sections, we will zoom in on the realization of both the control plane and data plane of the proposed Virtual Network Operator concept.

## IV. Control Plane

As explained, the control plane is responsible for properly managing the LPWANs under control of the VNO. Typically, LPWA network equipment provides its own APIs for configuration and management, which would require the VNO to support all these APIs, resulting in a significant coding and integration effort. Therefore, we propose the use of the LwM2M device management standard as a unified mechanism to manage all LPWA network equipment. Such a design decision requires a generic and easy-to-deploy solution for turning existing LPWA equipment into LwM2M equipment. For LPWANs that are closed, such as SigFox, control will merely consist of instantiating and configuring the data plane such that data for the managed devices can be exchanged using the APIs offered by this closed network. In the following subsections, we first explain the basic concepts of LwM2M, followed by a generic solution to build LwM2M compliant network equipment by means of Docker containers as well as novel reusable LwM2M data models to support the management of heterogeneous LPWANs.

### A. Intro to OMA LwM2M

The Lightweight Machine-to-Machine (LwM2M) specification is a secure, efficient and deployable client-server protocol, with several functionalities for managing machine-to-machine (M2M) and IoT devices on a variety of networks, from the Open Mobile Alliance (OMA) [10]. Recently, the version 1.0 of LwM2M protocol has been ratified by the Device Management Working Group at OMA in February 2017 [10].

In addition to fundamental management functionalities (e.g., bootstrapping, client registration, firmware updates, remote management and fault management), the LwM2M protocol also defines efficient interactions for transferring service and application data. It also defines appropriate semantics, i.e. common interfaces and data models, in order to boost interoperability in the IoT ecosystem and create more loosely coupled systems and better connective devices. An overview of the key LwM2M functionalities and the data model are presented in Figure 3. As it makes use of light and compact application protocols, management mechanisms and an efficient resource data model, the LwM2M protocol has already attracted much attention from both the industrial world and the research community for managing IoT devices, which involve both resource constrained end devices as well as gateway devices.

In order to adopt LwM2M in our VNO concept and use it to realize the control plane, two key challenges must be overcome. First, we need to be able to turn any type of existing LPWAN equipment into LwM2M compliant equipment, thereby mapping implementation specific APIs to standardized

LwM2M APIs. On top, this must be achieved in a generic and easy-to-deploy way. Second, as the usage of LwM2M for managing LPWAN equipment is novel, no suitable LwM2M data models exist. As such, novel, sufficiently generic data models must be proposed to perform the management of heterogeneous LPWAN components.

### B. Docker concept for LPWAN equipment

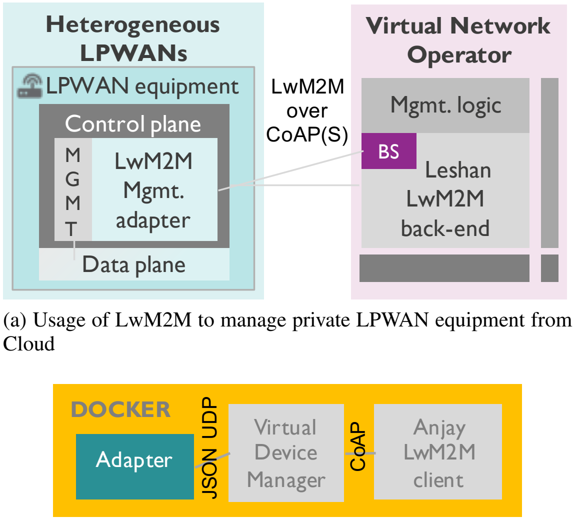
To overcome the first challenge, we introduce the idea of an easily deployable LwM2M Management Adapter that is able to convert legacy APIs into LwM2M compliant APIs, as shown in Figure 4a. The adapter is realized as a Docker container, as shown in Figure 4b, that can be easily deployed on top of existing LPWA network equipment, which typically runs a Linux OS with sufficient memory (+256Mb RAM) to run a single Docker instance. Inside the Docker container there are 3 components. The design of these components has been done in such a way to minimize the work that has to be done to implement the mapping between the existing APIs and the LwM2M APIs. These three components are the following:

1) *Anjay LwM2M client:* A standard-compliant LwM2M client that implements the LwM2M APIs for bootstrapping, registration, etc. as well as all required data models, including both existing LwM2M data models and custom models for LPWAN management. For the client, a modified version of Anjay, which is a C implementation of a LwM2M client, is used, which enables the creation of Object instances and resources by the Virtual Device Manager.
2) *Virtual Device Manager (VDM):* Via a well-defined API, the VDM receives configuration info from the Adapter, asking to create specific LwM2M Object instances and resources in the Anjay client. Once configured, the VDM also receives requests from the Adapter to update the state of LwM2M resources, updates which are passed to Anjay. Finally, the VDM also observes changes to LwM2M resources in Anjay and passes the changes back to the Adapter. The interface between Anjay and the VDM, is the standard LwM2M interface, with the VDM acting as a locally running trusted LwM2M server.
3) *Adapter:* The Adapter block is the only block that contains code that is specific to the device on top of which the Docker container is being deployed. It communicates with the device using its legacy APIs, implements the mapping to LwM2M objects and interacts with the VDM to create object instances, update object resources and be informed about their changes.

These three blocks are put inside a Docker image and run as a singular container that can be deployed on the LPWAN equipment once the Adapter has been properly implemented.

### C. LwM2M Data Models

Using the presented Docker concept and Virtual Device Manager approach, we are able to convert LPWAN specific APIs into LwM2M compliant APIs, provided suitable data models are available. LwM2M has defined several Objects and Resources to perform device management, but lacks suitable data models for the management of LPWAN network infrastructure. To this end, we have defined several new Objects and Resources using the private range of Object and Resource IDs that has been made available by OMA. These Objects and Resources have been defined in such a way that they are sufficiently generic for use across different LPWAN technologies and serve as a starting point for the VNO design. Table I gives an overview of the most important newly defined Objects and Resources, together with their ID, access rights (read, write or both) and short description.



(a) Usage of LwM2M to manage private LPWAN equipment from Cloud

(b) Realization by means of Docker and Virtual Device Manager

Fig. 4. Turning LPWAN equipment into LwM2M compliant devices.

The Profile Object contains general information about the device or components. For instance, when applied to LoRaWAN, it will reveal that it is a LoRa device and whether we are dealing with a LoRa gateway or Network Server. Next, we need to know in which multimodal networks the component is used, dealing with use cases where infrastructure is reused across different virtual LPWANs. Finally, information is provided about the number of gateways and devices handled by this gateway, which is equal to the number of instances of the Gateway Object and Device Object respectively. The first object provides info related to the identification of the gateway, the devices that are allowed to make use of it as well as statistics. The second object provides relevant device related info that must be known by the LPWAN network infrastructure as well as statistics at the device level. Finally, we have defined a Data Plane Object that is used to discover the data plane interfaces offered by the component and to properly configure those interfaces. For instance, LoRaWAN gateways often use a UDP connection to forward data to their Network Server, implying proper configuration of the Network Server's IP address and port at the gateway side. The Network Server itself hosts a Message Queuing Telemetry Transport (MQTT) publish-subscribe broker that makes the LoRaWAN data available to interested subscribers, i.e. MQTT clients. The

TABLE I
LwM2M OBJECTS AND RESOURCES FOR MANAGEMENT OF LPWAN INFRASTRUCTURE

| Name | ID | R/W | Description |
|---|---|---|---|
| **Profile Object** | **26241** | **-** | **General info on network component** |
| LPWAN tech | 26241 | R | Identifier describing technology type |
| Device type | 26242 | R | Type such as gateway, Network Server |
| LPWAN VID | 26243 | R/W | Unique ID of multimodal network(s) the component belongs to |
| # gateways | 26244 | R | Number of gateways managed by this device (equals to 1 for gateway device) |
| # devices | 26245 | R | Number of devices handled |
| **Data Plane Object** | **26242** | **-** | **Info/configuration of data plane interfaces to communicate with other components (multiple instances possible)** |
| Available methods | 26250 | R | ID for available interface methods such as UDP, MQTT, etc. |
| Interface type | 26251 | R | Incoming or outgoing interface |
| Select method | 26252 | R/W | Selected interface method to use |
| Configuration | 26253 | R/W | Configuration for selected iface method |
| **Gateway Object** | **26243** | **-** | **Info for gateway (multiple instance possible)** |
| Gateway ID | 26260 | R | Unique gateway ID |
| Accepted devices | 26261 | R/W | Devices that may use the gateway |
| On time | 5852 | R/W | Time since device has been turned on. Reset by writing value of 0. |
| # packets RX | 26263 | R/W | # of packets received. 0 to reset. |
| # packets TX | 26265 | R/W | # of packets transmitted. 0 to reset. |
| **Device Object** | **26244** | **-** | **Info about end nodes allowed to use network** |
| Device EUI | 26270 | R | Unique device identifier |
| Dev. address | 26271 | R/W | Address in multimodal LPWAN network |
| Nwk key | 26272 | R/W | Network session key used |
| Nwk config | 26273 | R/W | LPWAN network configuration of device |
| # packets RX | 26263 | R/W | # of packets received from device |
| # packets TX | 26265 | R/W | # of packets transmitted to device |

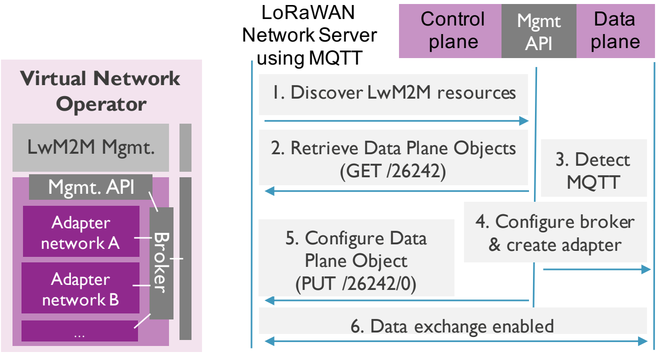
MQTT connection between the Network Server and such a client also requires proper configuration.

The defined objects are sufficiently generic to be used across different technologies. In addition, the LwM2M Object and Resource Model easily enables extensions, allowing the management capabilities to grow according to the needs identified.

## V. Data Plane

### A. *Adapters for a flexible data plane*

As explained in the previous section, the control plane can be used to configure the data plane, ensuring the LPWAN equipment can exchange data with the VNO. Apart from that, we still need the data plane counterpart at the VNO side. In order to be sufficiently flexible, the VNO should be able to interface using different communication or data exchange methods. To this end, we introduce an adapter-based concept where, upon the detection of a specific interface by the control plane, the right adapter can be instantiated in order to enable bidirectional data exchange with the newly discovered LWPAN infrastructure. Towards the right, all adapters will have a unified interface towards a broker inside the VNO for further exchange and handling of the data. This approach is illustrated in Figure 5a. Taking again our LoRaWAN example with the Network Server hosting an MQTT broker as an interface to exchange data coming from and going to the LoRaWAN network, the LwM2M-based control plane will discover the availability of this API. Using the management API to the data plane, an adapter that hosts an MQTT client will be instantiated and configured to connect to that broker in order to able to exchange traffic with the LoRaWAN network. As a last step, the control plane will provide the Network Server with any required information to accept this MQTT client. Once done, data exchange can take place. The process is shown in Figure 5b.


Virtual Network Operator
LwM2M Mgmt.
Mgmt. API
Adapter network A
Adapter network B
...
Broker
LoRaWAN Network Server using MQTT
Control plane
Mgmt API
Data plane
1. Discover LwM2M resources
2. Retrieve Data Plane Objects (GET /26242)
3. Detect MQTT
4. Configure broker & create adapter
5. Configure Data Plane Object (PUT /26242/0)
6. Data exchange enabled

(a) Adapter concept for the data plane

(b) Example data plane configuration for LoRaWAN Network Server using MQTT

Fig. 5. Data plane at the Virtual Network Operator side

### B. *Unified communication using SCHC*

Using the adapter concept explained, it is possible to exchange data with different types of LPWANs that are managed by the VNO. However, what is still missing, is a unified way to communicate over different technologies, avoiding a different payload encoding for every technology. The IETF LPWAN working group strives to realize this vision, adopting the Constrained Application Protocol (CoAP) over UDP and IPv6 as a common stack across different layer 2 LPWAN frame formats. To achieve this goal, a novel compression mechanism, named Static Context Header Compression or SCHC, is being designed. SCHC exploits the fact that most protocol header fields remain the same across packet transmissions between a device and its back-end. Therefore, a common context is stored in the LPWAN device and the network, which consists of a list of rules. Each rule has a unique identifier and provides a packet header template and compression actions such as omitting or abbreviating header fields. The rule that best matches an

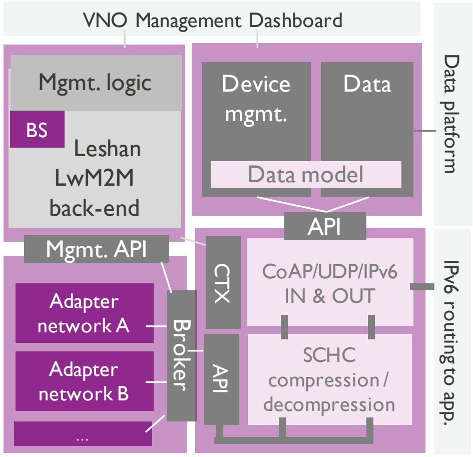


Fig. 6. Architecture of the Virtual Network Operator side including LwM2M-based management of LPWAN infrastructure, an extensible data plane and unified communication using SCHC.

outgoing packet is selected and the corresponding compression is applied. Using SCHC, packets can be compressed down to a few or even a single byte (i.e. only the rule ID). The same vision is applied to our VNO design, motivating multimodal LPWAN devices to use the standard-based CoAP stack and SCHC (de)compression on top of the LPWAN technologies used. At the VNO side, SCHC (de)compression can be applied based on the SCHC context stored for that device at the VNO (in the context block), resulting again in IPv6 packets. Depending on the approach chosen, the VNO can either forward the IPv6 packets to the final application (e.g. the Cloud platform of an organization) or can serve as the end point of the traffic on behalf of that organization. In the latter case, the VNO can also partake in device management (either using a custom-designed management protocol or, in the future, also using LwM2M, provided further optimizations are made to the LwM2M specification to enable more compact data exchanges to accommodate the small LPWAN packet sizes) and data handling. Figure 6 shows the final part of the VNO design. Important to note is that the Context block is also foreseen to perform decisions on how to send downlink data, but this is outside the scope of this paper.

## VI. Validation

In order to validate our proposed VNO concept and show its feasibility, we have realized an initial implementation of both the control and data plane of the proposed VNO architecture, which will be discussed in the following two subsections.

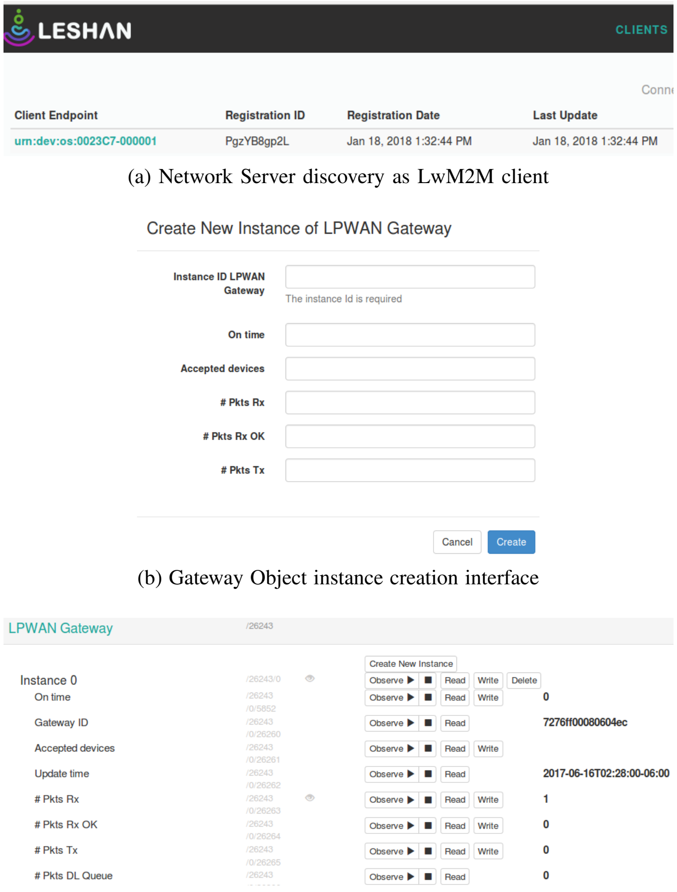
### A. Control plane

Based on the presented Docker concept and defined LwM2M data models for management of LPWAN network infrastructure, we can turn existing LPWAN equipment into LwM2M compliant devices that can be managed in a uniform way, independent of the underlying technology. Using the adapter concept, bidirectional data can be exchanged with various LPWAN equipment. As such, we have presented a flexible, extensible and standard-based Virtual Network Operator architecture that supports multimodal LPWAN use cases. In order to validate our architecture, we have implemented the proposed control plane approach for a LoRaWAN Network Server that runs the open source software LoraServer.io [11]. LoRaServer.io uses JSON REST APIs to obtain information related to the Network Server such as the number of gateways and end devices registered, traffic statistics per gateway and end devices, etc. The Adapter that is part of the Docker container that runs on the Network Server consumes these APIs and maps them the LwM2M objects shown in Table I for LPWAN management. As all these objects are custom LwM2M objects, we also implemented these objects in the Anjay client inside the Docker container and in the Leshan server that runs at the VNO. At this point, we implemented object 26243 for managing the gateways in the LoRaWAN Network Server. The result is a LoRaWAN Network Server that has been turned into a LwM2M compliant device with APIs for LPWAN management.

Following the LwM2M registration specifications, the Anjay client at the Network Server will automatically register with the VNO, as shown in Figure 7a. To set the gateways that are managed by the Network Server, the VNO can create a new instance of a Gateway object, as shown in Figure 7b, by using the user interface offered by Leshan. The request will be received by the Anjay client in the Docker container and the VDM will be informed. Next, the adapter will be alerted and will register the new gateway at the Network Server with the MAC address provided by the Anjay client. To do so, the API method */api/gateway/[MAC_ADD]* in the LoraServer.io is called. The other way around, the custom adapter in the Docker container interacts with the APIs of the LoRaWAN Network Server and informs the Anjay client via the VDM for any updates. For instance, all gateways registered with the Network Server can be discovered by polling the */api/gateways* API method. Once all registered gateways have been discovered, the traffic statistics can be observed using the LwM2M API as shown in Figure 7c. Behind the scenes, the */api/gateways/[MAC_ADD]/stats* API method is polled to get the traffic statistics for the specific gateways. The outputs from the polling API methods are parsed and passed in JSON format towards the VDM by the custom adapter and then further to the Anjay client and the VNO. Alternative data formats such as XML might be considered as well.

### B. SCHC data plane

To validate the data plane and unified communication using SCHC, we implemented the header compression mechanism for the CoAP/UDP/IPv6 stack, as described by the IETF in [12], [13], on top of a LoRaWAN stack, running on the LPWAN device, and in Click Router [14], running at the VNO side. In the best case, the header can be compressed down to a few bytes. Consisting of the rule ID and some inevitable bytes of the CoAP header. The proposed solution by the IETF assumes the compressor to use 1 context containing the fields of the full protocol stack. However, this would require many



LESHAN

CLIENTS

Conne

| Client Endpoint | Registration ID | Registration Date | Last Update |
|---|---|---|---|
| urn:dev:os:0023C7-000001 | PgzYB8gp2L | Jan 18, 2018 1:32:44 PM | Jan 18, 2018 1:32:44 PM |

(a) Network Server discovery as LwM2M client

Create New Instance of LPWAN Gateway

Instance ID LPWAN Gateway
The instance Id is required

On time

Accepted devices

# Pkts Rx

# Pkts Rx OK

# Pkts Tx

Cancel Create

(b) Gateway Object instance creation interface

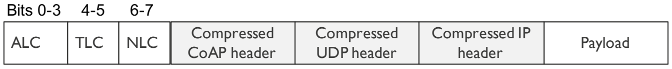
LPWAN Gateway /26243

Create New Instance

| | | | | |
|---|---|---|---|---|
| Instance 0 | /26243/0 | Observe ▶ ■ | Read Write Delete | |
| On time | /26243/0/5852 | Observe ▶ ■ | Read Write | 0 |
| Gateway ID | /26243/0/26260 | Observe ▶ ■ | Read | 7276ff00080604ec |
| Accepted devices | /26243/0/26261 | Observe ▶ ■ | Read Write | |
| Update time | /26243/0/26262 | Observe ▶ ■ | Read | 2017-06-16T02:28:00-06:00 |
| # Pkts Rx | /26243/0/26263 | Observe ▶ ■ | Read Write | 1 |
| # Pkts Rx OK | /26243/0/26264 | Observe ▶ ■ | Read Write | 0 |
| # Pkts Tx | /26243/0/26265 | Observe ▶ ■ | Read Write | 0 |
| # Pkts DL Queue | /26243 | Observe ▶ ■ | Read | 0 |

(c) Gateway Object instance observation interface

Fig. 7. Turning a LoRaWAN Network Server into a LwM2M compliant device using a Docker container and custom LwM2M objects.



| Bits 0-3 | 4-5 | 6-7 | | | | |
|---|---|---|---|---|---|---|
| ALC | TLC | NLC | Compressed CoAP header | Compressed UDP header | Compressed IP header | Payload |

Fig. 8. SCHC: compressed packet structure

rule tables for e.g. varying CoAP requests and a static UDP/IP context. Therefore, we implemented an extension of the compression mechanism, layered SCHC, proposed by Abdelfadeel et al. [15]. An example packet is shown in Figure 8. The layered approach enables up to 15 different CoAP contexts, 3 UDP contexts and 3 IPv6 contexts (rule 0 is reserved for an uncompressed header), but can be adjusted, depending on the use case. Rule ID's will therefore range from 0 to 255 and require the VNO to map a set of rules to a device's L2 address, in order to keep the rule ID as small as possible.

Our implementation, applied to a LoRaWAN deployment that is managed by our VNO, results in the end-to-end data flow that is shown in Figure 9. The LPWAN device fabricates a CoAP/UDP/IPv6 packet that is fed to the SCHC compressor (step 1). The resulting compressed SCHC packet is sent over the air as the payload of a LoRaWAN packet (step 2) and is then picked up by a LoRaWAN gateway and forwarded to its Network Server over UDP (step 3). Using MQTT, the

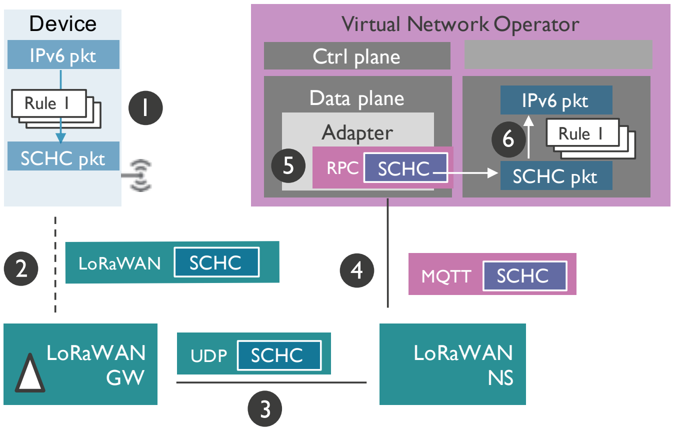


Fig. 9. End-to-end data flow via data plane and using SCHC

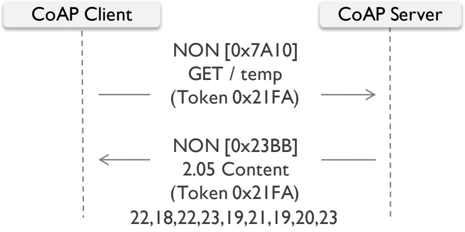


Fig. 10. A GET request with a separate response

Network Server delivers the SCHC packet to its corresponding adapter at the VNO (step 4). Using a Remote Procedure Call, the adapter at the data plane passes the packet to the SCHC decompressor at the VNO (step 5), where the corresponding rule ID for that device is applied to end up with the original IPv6 packet (step 6). The same will hold for downlink communication.

We have put forward SCHC as a solution to have unified and standardized communication across different LPWAN technologies that are being managed by our VNO. As these technologies have limitations in terms of bandwidth and maximum packet size, it is important to verify whether SCHC compression can limit the additional overhead to a bare minimum. To do so, we have considered an example CoAP request, as shown in Figure 10, and evaluated the resulting response packet size for the following 4 scenarios:

1) No compression, i.e. a complete CoAP/UDP/IPv6 packet
2) A varying IPv6 destination, in the same IPv6 pool, a varying UDP destination port and a non-compressed CoAP header
3) Full IPv6 and UDP compression, except for some variation on the CoAP part: `GET /humi` with 2 bytes token length and 2 bytes message id
4) A fully compressed CoAP, UDP and IPv6 header

The packet contents for the different situations is shown in Table II. Each packet contains the same payload and header. Only the header compression ratio differs, resulting in different packet sizes. If the rules are set up in the most efficient

TABLE II
SCHC EVALUATION FOR DIFFERENT SITUATIONS

| | SCHC | CoAP | UDP | IPv6 | Payload | Total |
|---|---|---|---|---|---|---|
| **Situation 1** | 1 | 12 | 8 | 40 | 9 | 70 |
| **Situation 2** | 1 | 12 | 2 | 8 | 9 | 32 |
| **Situation 3** | 1 | 6 | 0 | 0 | 9 | 16 |
| **Situation 4** | 1 | 2 | 0 | 0 | 9 | 12 |

way possible, the CoAP header can be compressed down to 2 bytes, without the help of any intermediaries, since the token and message id are vital for CoAP communication, these can only be compressed to a minimum of 1 byte. The total compression achieved is 83%. If only the headers are compared, a compression ratio of 95% is achieved. The last situation demonstrates that using SCHC, a CoAP/UDP/IPv6 packet can even be sent over Sigfox, with maximum packet lengths down to 12 bytes and 8 bytes for uplink and downlink respectively, meaning that it is a viable choice for achieving unified communication in our multi-modal LPWANs.

## VII. CONCLUSION

As a single LPWAN technology is not able to cover the requirements of every IoT use case, there is a trend to move towards multimodal deployments, where devices generate data of interest across different LPWAN deployments. Apart from that, the long-range character of these networks, motivates the reuse of infrastructure by multiple organizations. To deal with the complexity of managing and using such networks, this article presented the concept of a cloud-based Virtual LPWA Network Operator that takes care of properly configuring and managing LPWAN equipment as well as enabling unified data exchanges across heterogeneous LPWANs, relieving the users from this burden. How this control and data plane are being realized has been described in detail and has been validated by means of an implementation.

Using distributed Docker containers that map implementation specific management APIs to LwM2M compliant APIs and data models, a standard-based, easy-to-deploy and extensible control plane has been realized. By combining adapters for LPWAN data exchanges with the most recent state-of-the-art work for realizing efficient IPv6 communication over LPWANs, the feasibility of unified end-to-end communication across different LPWAN technologies has been shown.

As such, we believe to have shown a viable architecture for the future generation of multimodal LPWA networks and devices. Of course, our architecture only provides the foundations in realizing the control and data plane. Several open challenges remain both at the control plane and the data plane. At the control plane, we target more advanced management, thereby taking into account the life cycle of multimodal devices, as well as research on improving the scalability and coexistence of such co-located networks, as discussed in [16]. At the data plane, we see the need of an intelligent usage of the multiple LPWAN technologies for both uplink and downlink communication depending on aspects such as traffic requirements and network availability. Working on these topics, will definitely pave the way for more demanding IoT applications in a multimodal LPWAN context.

## ACKNOWLEDGMENT

Part of this research was funded by the Flemish FWO SBO S004017N IDEAL-IoT (Intelligent DEnse And Longe range IoT networks) project, and by the ICON project MuSCLe-IoT. MuSCLe-IoT is a project realized in collaboration with imec, with project support from VLAIO (Flanders Innovation and Entrepreneurship). Project partners are imec, Flash Private Mobile Networks, Engie M2M, Sensolus and Aertssen.

## REFERENCES

[1] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 855–873, Secondquarter 2017.

[2] M. Haghighi, Z. Qin, D. Carboni, U. Adeel, F. Shi, and J. A. McCann, "Game theoretic and auction-based algorithms towards opportunistic communications in lpwa lora networks," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Dec 2016, pp. 735–740.

[3] (2017) Murata partners with stmicro to add sigfox. [Online]. Available: https://www.murata.com/about/newsroom/news/product/frontend/2017/0718

[4] J. M. Batalla, G. Mastorakis, C. X. Mavromoustakis, and J. Zurek, "On cohabitating networking technologies with common wireless access for home automation system purposes," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 76–83, October 2016.

[5] B. Reynders, W. Meert, and S. Pollin, "Range and coexistence analysis of long range unlicensed communication," in *2016 23rd International Conference on Telecommunications (ICT)*, May 2016, pp. 1–6.

[6] M. Masoudi, A. Azari, E. A. Yavuz, and C. Cavdar, "Grant-free radio access iot networks: Scalability analysis in coexistence scenarios," *CoRR*, vol. abs/1711.00581, 2017. [Online]. Available: http://arxiv.org/abs/1711.00581

[7] F. Granelli, A. A. Gebremariam, M. Usman, F. Cugini, V. Stamati, M. Alitska, and P. Chatzimisios, "Software defined and virtualized wireless access in future wireless networks: scenarios and standards," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 26–34, June 2015.

[8] M. Yang, Y. Li, D. Jin, L. Zeng, X. Wu, and A. V. Vasilakos, "Software-defined and virtualized future mobile and wireless networks: A survey," *Mobile Networks and Applications*, vol. 20, no. 1, pp. 4–18, Feb 2015. [Online]. Available: https://doi.org/10.1007/s11036-014-0533-8

[9] A. Nikravesh, Y. Guo, F. Qian, Z. M. Mao, and S. Sen, "An in-depth understanding of multipath tcp on mobile devices: Measurement and system design," in *Proceedings of the 22Nd Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '16. New York, NY, USA: ACM, 2016, pp. 189–201. [Online]. Available: http://doi.acm.org/10.1145/2973750.2973769

[10] Open Mobile Alliance, "Lightweight Machine to Machine Technical Specification," February 2017.

[11] https://docs.loraserver.io/overview/, Last visited January 2018.

[12] A. Minaburo, Acklio, L. Toutain, and C. Gomez, "Lpwan static context header compression (schc) and fragmentation for ipv6 and udp," Internet draft, IETF, 2017.

[13] A. Minaburo, Acklio, and L. Toutain, "Lpwan static context header compression (schc) for coap," Internet draft, IETF, 2017.

[14] 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. [Online]. Available: http://doi.acm.org/10.1145/354871.354874

[15] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, "Lschc: Layered static context header compression for lpwans," in *Proceedings of the 12th Workshop on Challenged Networks*, ser. CHANTS '17. New York, NY, USA: ACM, 2017, pp. 13–18. [Online]. Available: http://doi.acm.org/10.1145/3124087.3124092

[16] E. De Poorter, J. Hoebeke, M. Strobbe, I. Moerman, S. Latr, M. Weyn, B. Lannoo, and J. Famaey, "Sub-ghz lpwan network coexistence, management and virtualization: An overview and open research challenges," *Wirel. Pers. Commun.*, vol. 95, no. 1, pp. 187–213, Jul. 2017. [Online]. Available: https://doi.org/10.1007/s11277-017-4419-5