Virtualizing LoRaWAN Nodes: a CoAP-based Approach

Antonio Cilfone, Luca Davoli, and Gianluigi Ferrari
Internet of Things (IoT) Lab
Department of Engineering and Architecture, University of Parma
Parco Area delle Scienze, 181/A, 43124 Parma, Italy
Email: {antonio.cilfone, luca.davoli, gianluigi.ferrari}@unipr.it

Abstract—**In the near future, Internet of Things (IoT) will play a relevant role in people's lives and one of the main challenges will be the integration of heterogeneous networks. Among widely adopted network technologies, the development of the so-called Low-Power Wide-Area Networks (LPWANs), in particular Long Range WAN (LoRaWAN) is attracting a significant interest from both academic and industrial worlds. The integration of LoRaWAN with other communication technologies represents a fundamental requirement for a successful rapid and large-scale diffusion of IoT paradigms, such as Smart Farming, Smart Factory, and Smart City. The aim of this paper is two-fold: we propose (i) a mechanism for automatic discovery of the sensors a LoRa device is equipped with; and (ii) a novel networking architecture, based on cloud computing and node virtualization, to enable the interaction of LoRaWAN end-nodes with other IP-based IoT devices. Our solution does not impact LoRaWAN networking and enables a seamless interaction between LoRaWAN end-nodes and other Constrained Application Protocol (CoAP)-based nodes.**

Index Terms—**Internet of Things, LPWAN, LoRaWAN, CoAP, Smart City, Smart Farming, Cloud Computing, Virtualization.**

I. Introduction

The Internet of Things (IoT) represents one of the key technological revolutions of last decades, especially thanks to the technological progresses allowing to develop devices meeting the constraints imposed by the IoT. In the near future, it is expected that IoT will play a significant role in people's lives, leading to the design and development of new paradigms and architectures. There will be new challenges, thus leading to the deployment of new services. The IoT can be described as a heterogeneous ecosystem where a massive number of constrained devices (denoted as Smart Objects, SOs) will be deployed and connected in order to efficiently cooperate for multiple purposes, such as data collection, actuation, and interaction with people, in a Human-to-Machine (H2M)-oriented way.

In this context, IoT networks are usually deployed through the integration of several heterogeneous communication and application protocols. Looking at this, one of the main challenges is the seamless integration and interaction of heterogeneous networks, e.g., in terms of transmission range capabilities, when both short-range and long-range communication protocols have to be used [1].

Among widely adopted networking technologies, the so-called Low-Power Wide-Area Networks (LPWANs) are attracting a significant interest from both academic and industrial worlds. LPWANs have the advantage to meet almost all the (usually very constrained) IoT requirements, such as: (i) easy and inexpensive deployment, (ii) wide coverage, (iii) simple architecture, (iv) use of unlicensed bands, and (v) low-power consumption, at the cost of few limitations (i.e., limited data-rate; restrictions on uplink and downlink capabilities; and nominal network throughput). Moreover, LPWANs fill the gap between short-range multi-hop networks and cellular-oriented ones; as an example, one of the most studied LPWANs is Long Range WAN (LoRaWAN), operating in the Industrial, Scientific and Medical (ISM) bands.

Thus, the integration of LoRaWAN with other technologies (and, in general, when SOs use different communications paradigms [2]) represents a fundamental requirement for a successful rapid and large-scale diffusion of Smart Farming, Smart Factory, and Smart City paradigms. Indeed, a good solution would be to leave the standard architecture of LPWANs unmodified, while to adopt self-organizing network's mechanisms enabling heterogeneous networks to interact with each other, as anticipated before. In this context, devices' virtualization paradigms will likely play a key role in enabling the desired trade-off between network flexibility and performance.

In this paper, we propose a novel architecture enabling the interaction between LoRa-based nodes and IP-based IoT nodes. This interaction is based on the concept of cloud computing and virtualization of end-nodes [3]. More in detail, we do not modify the behavior of LoRa devices nor the LoRaWAN architecture, but we provide a mechanism for the discovery of the sensors mounted on a LoRa device and the related virtualization of LoRa-based nodes by emulating them as CoAP servers. In this way, our solution enables a virtual seamless interaction between end-nodes and external—usually IP-based—devices. Hence, the computation is performed on the server side, leaving both the general network architecture and the end-node protocol stack unchanged.

The remainder of this paper is organized as follows. In Section II, we provide a short introduction on LoRaWAN. Section III briefly overviews LoRa-based applications and LoRaWAN integration with other technologies. In Section IV, we present the proposed architecture. Finally, in Section V, we draw our conclusions.

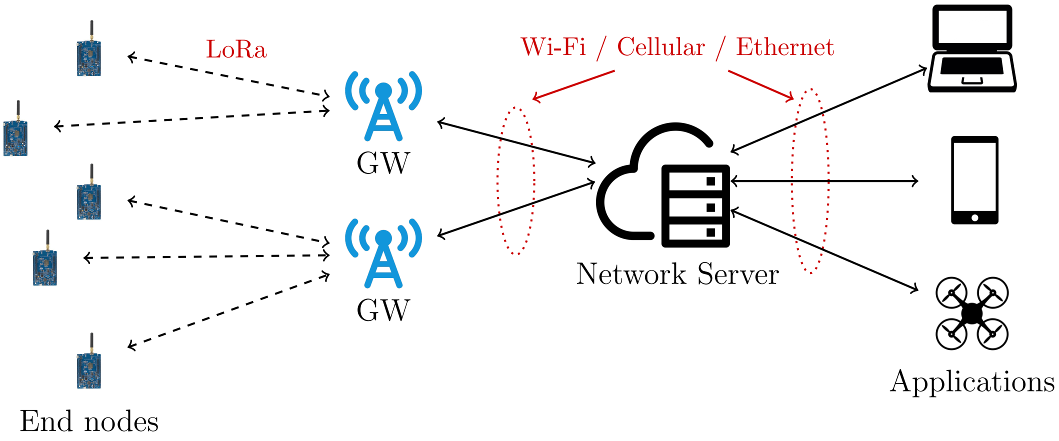

Fig. 1. LoRaWAN architecture.

II. Overview on LoRa

As highlighted in Section I, LPWANs are designed to offer affordable connectivity to a large number of constrained devices geographically distributed over large areas. In the following, we provide a brief introduction on LoRa and LoRaWAN, highlighting the fact that LoRa and LoRaWAN refer to distinct concepts: LoRa is a proprietary modulation, based on Chirp Spread Spectrum (CSS) and patented by Semtech Corporation. Therefore, LoRa represents the physical (PHY) layer, while LoRaWAN specifies the radio access method and the network architecture, is open, and is defined by the LoRa Alliance.

In detail, the LoRa modulation uses different Spreading Factors (SFs) which define the duration of the signal—the higher the SF, the longer the symbol time, as well as the longer the distance supported by a link. In detail, the SF may vary between 7 and 12, with SF = 12 resulting in the highest sensitivity and range, but achieving the lowest data rate and increased energy consumption [4]—intuitively, the longer the symbol duration, the more active the radio transceiver. A decrease of one unit in SF doubles the transmission rate and halves the transmission duration, as well as the energy consumption. Since chirps at different SFs are orthogonal, the Gateways (GWs) can thus receive multiple transmissions in the same frequency band with different SFs.

Regarding the network topology, as shown in Fig. 1, LoRaWAN entails a "star-of-stars" network topology composed of end-nodes and GWs. The latter are in turn connected (through IP-based networks) to a Network Server (NS), which is finally connected to high-layer applications. Moreover, in LoRaWAN, the end-nodes can be of three types: Class `A`, `B`, and `C`, where the Class defines the behavior about downlink packets. In Class `A`, which must be supported by all LoRaWAN devices, a device can receive downlink packets only after sending a packet, thus resulting in the lowest energy consumption mode. Class `B` devices are suited to applications requiring a more intense downlink traffic, since they open extra receive slots at scheduled intervals by receiving a time-synchronized beacon from the GW. Finally, Class `C` devices are always listening to the channel: their energy consumption level is thus the highest one, but they can receive a downlink packet at any time, leading to the lowest downlink latency. GWs receive packets from all the nodes in their reception range and forward packets to the NS, which is then responsible for the management of the LoRaWAN network. A NS can handle multiple GWs—usually, it receives the same packet (originally sent by an end-node) from more than one GW.

The radio channel access in LoRaWAN is based on ALOHA access method: (i) an end-node wakes up and sends a packet on a selected radio channel; (ii) one or more GWs, within the transmission range of the node, receive the packet and (iii) forward it to the NS, that eventually processes the received packets. Since in Europe LoRaWAN operates in the unlicensed bands (868 MHz bands), both end-devices and GWs must comply with the ETSI limitation (depending on the frequency, it varies from 0.1% to 10%) of the duty cycle—unless they perform Listen-Before-Talk (LBT) or frequency-hopping techniques. Due to this constraint, each time a frame is transmitted, the *Time on Air* is calculated and, subsequently, the time in which the transmitter can not use the channel, denoted as *time off* (T_{OFF}), is evaluated.

In order to participate to the network operations, a LoRa-compliant node must be registered and activated through the NS. LoRaWAN defines two activation methods: (i) Over-The-Air-Activation (OTAA), which is the most secure as the end-node sends a *join-request* frame to the NS, which, in turn, (potentially) sends a *join-accept* frame; and (ii) Activation-By-Personalization (ABP), in which there are no join procedures, since the end-device has all the required configurations for the activation. In our work, we will refer only to the OTAA procedure, since it is the one recommended by the LoRa Alliance—due to its high security level.

A. Cayenne LPP

The limitations on the payload size of a LoRa packet—in Europe, the smallest payload is 59 bytes at SF12—is relevant when designing LPWAN-oriented applications. One of the

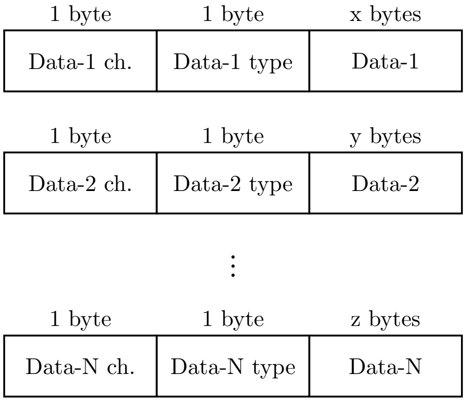

Fig. 2. Uplink payload structure of Cayenne LPP.

most widely adopted payload formats is the Cayenne Low Power Payload (LPP) [5]. Cayenne LPP is compliant with the payload size's limitations, can be reduced to 11 bytes, and allows the end-device to send multiple sensor data at once. In addition, it allows to split the same data across consecutive frames, prefixing each sensor data with the following 2 bytes:

- *data channel*, which identifies each sensor in the device across frames—namely, it is an identification number for the sensors, useful in the case of multiple sensors of the same nature;
- *data type*, which identifies the sensor type in the frame (e.g., "temperature"). In Table I, the identifiers and data resolution values related to the sensors adopted in this work are summarized.

As shown is Fig. 2, a frame is structured as a sequence of *data channel*, *data type*, and "raw" data, without additional separators. Moreover, *data types* are compliant with the IPSO Alliance Smart Objects Guidelines, which identify each *data type* with an "Object ID" [6] usually longer than 1 byte. Thus, a conversion is required in order to compress the Object ID into 1 byte: the ID of the LPP *data type* is obtained converting in HEXadecimal (HEX) value the result of the following operation:

$$LPP_{\text{DATA\_TYPE}} = IPSO_{\text{OBJECT\_ID}} - 3200.$$

In our work, we exploit the frames received in LPP format in order to perform an automatic discovery of the sensors each node is equipped with—creating a CoAP resource for each sensor.

III. Related Works

From the considerations in Section I and Section II, it emerges that LoRa and, in general, LPWANs, are interesting solutions for typical IoT scenarios, such as Smart Farming [7], [8], Smart Factory [9], [10], and Smart City [11]. Regarding Smart City applications, the use of LoRaWAN has been motivated and justified considering both the technological side [12], [13], [14], through extensive performance analysis, and the economic side [15], proving that LoRa is a key enabler for the wide adoption of Smart City-like paradigms.

TABLE I
LPP sensor codes

Sensor	**HEX value**	**Length [bytes]**	**Data Resolution per bit**
Analog input	`0x02`	2	0.01 Signed
Temperature	`0x67`	2	0.1 °C Signed MSB
Humidity	`0x68`	1	0.5% Unsigned
Accelerometer	`0x71`	6	0.001 G Signed MSB per axis
Barometer	`0x73`	2	0.1 hPa Unsigned MSB
GPS	`0x88`	9	Lat: 0.0001° Signed MSB Lng: 0.0001° Signed MSB Alt: 0.01 m Signed MSB

Since most of the literature on Smart Industry is related only to performance and coverage evaluation, we focus on Smart Agriculture and Smart City. Regarding the first scenario, in [7] a LoRaWAN-based irrigation system has been deployed in a real context in China, with the irrigation mechanism remotely manageable using a mobile application, in turn used to trigger the irrigation through downlink messages to the end-node. Another interesting solution is proposed in [8], where authors highlight the importance of IoT and LoRaWAN in a Smart Farming scenario, using the radio access protocol to create a service-oriented solution in which data collected from the field are used to trigger actuators and then stored in databases for further processing.

In [16], LoRa (the PHY layer) has been used to monitor temperature and velocity of a river in Dublin for a 8-month period: authors conclude that a better approach would rely on the use of LoRaWAN to achieve a higher security level. However, their results confirm the reliability of such a solution. LoRaWAN is considered also in [17], in order to collect information about waste containers in the region of Salamanca, Spain: through the containers' levels monitoring, the authors optimize the routes for waste collection, thus reducing both fuel consumption and environmental pollution.

Nevertheless, to the best of authors' knowledge, LoRa/LoRaWAN-based solutions in the literature are considered as standalone systems, with very limited integration with other protocols, if compared to other "smart" services. In fact, considering, as an example, the case of a Smart City, the future services offered by a municipality should be integrated taking into account different types of network, such as vehicular networks (Vehicle-to-Everything, V2X), in which LoRaWAN will play a key role [18]. The same holds for a Smart Farming scenario, in which, considering the farm as a service, different types of networks must be taken into account, e.g., connected Unmanned Aerial Vehicles (UAVs) and blockchain for food traceability [19].

IV. Virtualization

The virtualization architecture proposed in this work is based on the classical LoRaWAN networking, in which end-nodes send packets to LoRa GWs which, eventually, forward

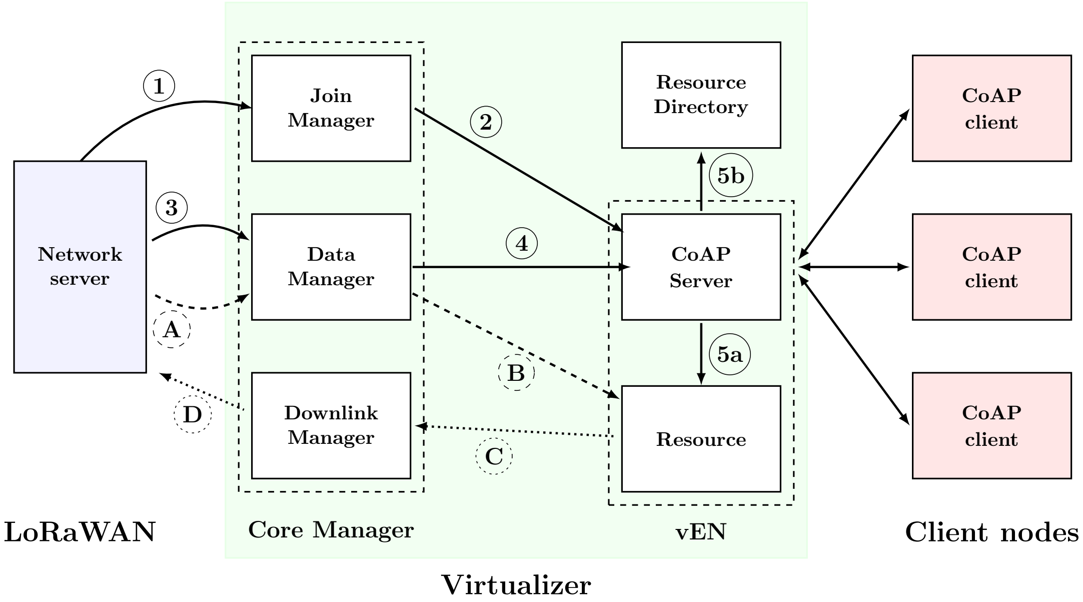

Fig. 3. Architecture of the Virtualizer.

them to the NS. Beyond the NS, the proposed infrastructure, called *Virtualizer*, manages the "discovery" of resources available on the end-nodes—e.g., on-board sensors. The main goal of the proposed virtualization architecture is to replicate (i.e., to create a "digital twin" of) the end-nodes on top of the IP layer. We denote a LoRaWAN node's virtual replica as *virtual End Node* (vEN). In this way, a generic IP-compliant node, using CoAP as application protocol, can interact with the deployed sensors. From a wider perspective, any application protocol (e.g., HTTP) could be chosen such that a vEN can interact with a large number of IoT devices.

Moreover, another key point of our proposed virtualization architecture is that it allows direct communication (a sort of "end-to-end" interaction) and routing among LoRaWAN nodes. Such interaction is not foreseen in the reference LoRaWAN architecture since, normally, a LoRaWAN node can not directly communicate or interact with another LoRaWAN device. This data exchange behavior is thus enabled through the *Data Manager* (DM) component, which is implemented as a MQTT client subscribing to all uplink topics and listening to all incoming packets. Upon reception of new data from a device, the DM looks for the vEN corresponding to the specific LoRa end-node and leases its data, storing it until new ones arrive. The main advantage of this approach is that if an internal node receives two requests within an interval shorter than the off period T_{OFF} of LoRa nodes, it can still use the last available data to provide a response to the external node, thus providing a *caching* mechanism to the entire virtualization infrastructure.

A. LoRaWAN Network Virtualizer

The main building blocks of our virtualization solution are shown in Fig. 3, in which: (i) solid arrows describe the "discovery" phase; (ii) dashed arrows represent the "normal" behavior of LoRa-based nodes (i.e., data sent in uplink, collected by the vENs, and thus sent as CoAP response packet's payload to external CoAP requestor clients); and (iii) dotted arrows denote data flows for possible downlink communications.

In general, the NS can exchange information about end-nodes with external non-LoRa networks using different protocols and paradigms, such as, for example, REST API, MQTT, WebSocket. In our work, we use MQTT protocol, since it is a common choice in several IoT scenarios. As stated in Section II, in our implementation we consider OTAA (instead of ABP) as join procedure: it follows that once a node joins the LoRaWAN network, the join event is published on the `events/` topic (step (1) in Fig. 3). A specific software module, named as *Join Manager*, runs a MQTT client subscribing to the events' topic and retrieving the end-node's device address. Thus, the Join Manager starts a CoAP server (step (2) in Fig. 3) with a certain IP address, associated with the "discovered" device address. In this way, our virtualization approach has a good scalability, since a "reserved" CoAP server is started for each LoRa end-node participating to the LoRaWAN network.

Resource discovery is another key aspect of our implementation. In fact, when a LoRa end-node is deployed, the back-end architecture does not generally know in advance which data will be received from this node. In the same way, the LoRaWAN NS does not know the amount and nature of sensors each LoRa end-device is equipped with. Hence, our virtualization solution allows an automatic discovery of these sensors by parsing the LoRaWAN packet's payload (under the assumption that an end-node is using the Cayenne LPP format, described in Section II-A). For each sensor on the

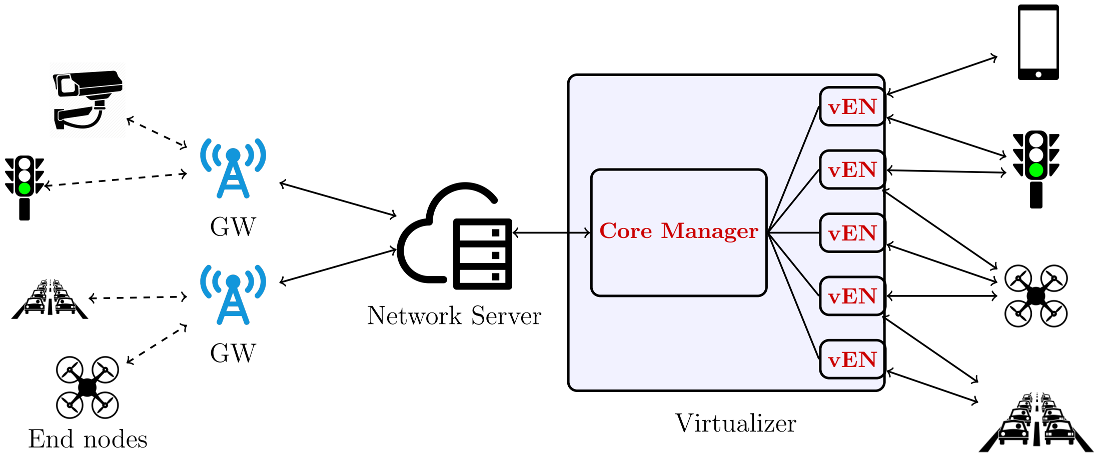

Fig. 4. Extended LoRaWAN with the developed Virtualizer.

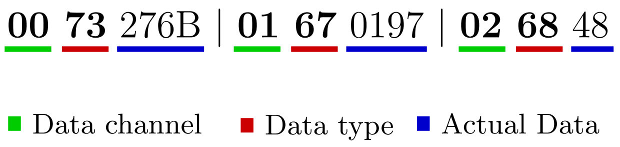

Fig. 5. Example of received uplink packet.

end-node, the corresponding vEN creates a CoAP resource as *virtual replica* of the specific sensor. In detail, the DM gathers data from the NS (step (3) in Fig. 3) and then parses the received payload: an illustrative example is given in Fig. 5, where the main fields of the received payload are highlighted. Once the analysis of *data type* codes is completed, the DM sends a CoAP POST request to a dummy resource (denoted as /handler) of the specific vEN (step (4) in Fig. 3), with a payload containing the name of the resources to be added. As an example, if *data type* code corresponds to 0x73, a *barometer* resource is started. Finally, the triggered vEN starts a CoAP resource (step (5a) in Fig. 3) of the proper type. The syntax used in our implementation is simple: the CoAP resource name corresponds to sensor name, as indicated in the first column of Table I. Once the resources are added, they are updated by the DM, which sends a CoAP PUT request to the proper resource, with the updated value as payload.

Considering the uplink payload in Fig. 5, for each "real" sensor the parser obtains both *data channel* and *data type* (in green and red color in Fig. 5, respectively) according to values in Table I. Then, the parser decodes the values according to bit resolution rules in Table I. Finally, for each sensor, the DM sends a CoAP PUT request (step (B) in Fig. 3) to proper resources, setting the decoded value as the payload. As an example, in Fig. 5, the discovered sensors are: (i) barometer (0x73) with 1009.1 hPa as decoded value, (ii) temperature (0x67) with 40.7 °C as value, and (iii) humidity (0x68) with 36% as decoded value. Thus, the following CoAP PUT requests are sent to the IPv4/IPv6 address of the virtualized end-node.

For sensors discovery:

- POST <ip>/handler   barometer
- POST <ip>/handler   temperature
- POST <ip>/handler   humidity

For sensors update:

- PUT <ip>/barometer   1009.1
- PUT <ip>/temperature   40.7
- PUT <ip>/humidity   36

Once the nodes and their resources are discovered and added to the *Resource Directory* (RD) [20], the rest of the message exchange corresponds to the classical interaction among CoAP nodes [21]: an external client can look for resources on the RD and sends any CoAP request to the vENs. According to the proposed architecture, the behavior of the LoRaWAN network is transparent to external (CoAP) clients: for these clients, virtualized nodes appear as classical IP-based nodes. When an end-node starts to transmit data, a specific module, the *Payload Parser Block* (PPB), analyzes the uplink packet (step (3) in Fig. 3), retrieves the *data type* and, consequently, the sensors available on the device.

B. *Illustrative Implementation*

In order to make the Virtualizer suitable to a generic computing platform, it has been developed in Java language, so that it can run on any machine supporting a Java Virtual Machine (JVM): illustrative platforms are a Raspberry Pi 3 Model B (RPi3) [22] and a laptop with an Intel i7-7700HQ CPU. LoRa end-nodes are STM STEVAL-STRKT01 LoRa devices [23], equipped with a Cortex M0+ CPU and with the following sensors: temperature, humidity, accelerometer, barometer, GPS, and battery voltage's analog reading. The LoRa GW runs on top of a RPi3, while, for the NS, we use an open-source implementation, denoted as *lorawan-server* [24], running on another RPi3.

As a confirmation of the modularity and flexibility of the proposed solution and in order to show its independence from a specific network operator, our approach has been successfully tested using the LoRaWAN NS deployed on a public network called "The Things Network" (TTN) [25].

In case an external CoAP client wants to send a downlink message to an internal LoRa node (e.g., to perform an action based on data collected from some other nodes), it simply sends a CoAP PUT request to the virtual CoAP resource; this request is sent to the DM (step (C) in Fig. 3) that, eventually, sends the request to the NS (step (D) in Fig. 3) to reach the LoRa end-node.

V. Conclusions

In this paper, we have presented a virtualization architecture for LoRaWAN nodes in which each end-device is replicated as a CoAP server, denoted as vEN, on a central Virtualizer that interacts with the LoRaWAN NS. Its resources are *discovered* by parsing the payload, which is in Cayenne LPP format. In this way, a common IP-based device can communicate with a LoRaWAN-based node. Our approach allows both communication and routing among LoRaWAN nodes, thus overcoming the impossibility of inter-node communication proper of the LoRa specifications. The LoRaWAN network's behavior is completely transparent for external nodes. Moreover, this approach can be further extended to other types of networks, thus allowing rapid integration, flexibility and scalability. Finally, our architecture has the advantage of being compliant with standard protocols, since it relies on CoAP and MQTT protocols, which can be considered typical IoT-oriented protocols.

Acknowledgment

The work of A. Cilfone, L. Davoli and G. Ferrari is partially funded by: the European Commission H2020 Framework Program, under Grant No. 783221, AFarCloud project - "Aggregate Farming in the Cloud," and the University of Parma, under "Iniziative di Sostegno alla Ricerca di Ateneo" program, "Multi-interface IoT sYstems for Multi-layer Information Processing (MIoTYMIP)" project. The work reflects only the authors' views; the European Commission is not liable for any use that may be made of the information contained herein.

References

[1] L. Davoli, L. Belli, A. Cilfone, and G. Ferrari, "From Micro to Macro IoT: Challenges and Solutions in the Integration of IEEE 802.15.4/802.11 and Sub-GHz Technologies," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 784–793, April 2018.

[2] M. Nitti, V. Pilloni, G. Colistra, and L. Atzori, "The Virtual Object as a Major Element of the Internet of Things: A Survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1228–1240, Secondquarter 2016.

[3] R. Morabito, "Virtualization on Internet of Things Edge Devices With Container Technologies: A Performance Evaluation," *IEEE Access*, vol. 5, pp. 8835–8850, May 2017.

[4] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the Limits of LoRaWAN," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, September 2017.

[5] myDevices. How Cayenne LPP works. Accessed on July 23, 2019. [Online]. Available: https://mydevices.com/cayenne/docs/intro/

[6] J. Jimenez, M. Koster, and H. Tschofenig, "IPSO Smart Objects," in *IoT Semantic Interoperability Workshop 2016*, San Jose, California, March 2016, pp. 1–7. [Online]. Available: https://www.iab.org/wp-content/IAB-uploads/2016/03/ipso-paper.pdf

[7] W. Zhao, S. Lin, J. Han, R. Xu, and L. Hou, "Design and Implementation of Smart Irrigation System Based on LoRa," in *2017 IEEE Globecom Workshops (GC Wkshps)*, Singapore, Singapore, December 2017, pp. 1–6.

[8] C. Cambra, S. Sendra, J. Lloret, and L. Garcia, "An IoT Service-Oriented System for Agriculture Monitoring," in *2017 IEEE International Conference on Communications (ICC)*, Paris, France, May 2017, pp. 1–6.

[9] J. Haxhibeqiri, A. Karaagac, F. Van den Abeele, W. Joseph, I. Moerman, and J. Hoebeke, "LoRa Indoor Coverage and Performance in an Industrial Environment: Case Study," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Limassol, Cyprus, September 2017, pp. 1–8.

[10] P. Neumann, J. Montavont, and T. Noël, "IIndoor Deployment of Low-Power Wide Area Networks (LPWAN): a LoRaWAN case study," in *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, New York, NY, October 2016, pp. 1–8.

[11] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-Range Communications in Unlicensed Bands: The Rising Stars in the IoT and Smart City Scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, October 2016.

[12] J. Jaime, I. Sousa, M. P. Queluz, and A. Rodrigues, "Planning a Smart City Sensor Network Based on LoRaWAN Technology," in *2018 21st International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Chiang Rai, Thailand, November 2018, pp. 35–40.

[13] D. Magrin, M. Centenaro, and L. Vangelista, "Performance Evaluation of LoRa Networks in a Smart City Scenario," in *2017 IEEE International Conference on Communications (ICC)*, Paris, France, May 2017, pp. 1–7.

[14] M. Loriot, A. Aljer, and I. Shahrour, "Analysis of the use of LoRaWan technology in a Large-Scale Smart City Demonstrator," in *2017 Sensors Networks Smart and Emerging Technologies (SENSET)*, Beirut, Lebanon, September 2017, pp. 1–4.

[15] F. Yu, Z. Zhu, and Z. Fan, "Study on the feasibility of LoRaWAN for smart city applications," in *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Rome, Italy, October 2017, pp. 334–340.

[16] W. Guibene, J. Nowack, N. Chalikias, K. Fitzgibbon, M. Kelly, and D. Prendergast, "Evaluation of LPWAN Technologies for Smart Cities: River Monitoring Use-case," in *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, San Francisco, CA, March 2017, pp. 1–5.

[17] Á. Lozano, J. Caridad, J. De Paz, G. Villarrubia González, and J. Bajo, "Smart Waste Collection System with Low Consumption LoRaWAN Nodes and Route Optimization," *Sensors*, vol. 18, no. 5, 2018.

[18] H. A. Khattak, H. Farman, B. Jan, and I. U. Din, "Toward Integrating Vehicular Clouds with IoT for Smart City Services," *IEEE Network*, vol. 33, no. 2, pp. 65–71, March 2019.

[19] A. Cilfone, L. Davoli, L. Belli, and G. Ferrari, "Wireless Mesh Networking: An IoT-Oriented Perspective Survey on Relevant Technologies," *Future Internet*, vol. 11, no. 4, 2019.

[20] S. Cirani, L. Davoli, G. Ferrari, R. Léone, P. Medagliani, M. Picone, and L. Veltri, "A scalable and self-configuring architecture for service discovery in the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 508–521, Oct 2014.

[21] L. Davoli, L. Belli, A. Cilfone, and G. Ferrari, "Integration of Wi-Fi mobile nodes in a Web of Things Testbed," *ICT Express*, vol. 2, no. 3, pp. 95–99, 2016, special Issue on ICT Convergence in the Internet of Things (IoT).

[22] Raspberry Pi Foundation. Raspberry Pi 3 Model B. Accessed on August 2, 2019. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[23] STMicroelectronics. STEVAL-STRKT01 LoRa® IoT Tracker. Accessed on August 2, 2019. [Online]. Available: https://www.st.com/en/evaluation-tools/steval-strkt01.html

[24] Petr Gotthard. LoRaWAN Network Server. Accessed on July 29, 2019. [Online]. Available: https://github.com/gotthardp/lorawan-server

[25] The Things Industries. The Things Network (TTN). Accessed on August 2, 2019. [Online]. Available: https://www.thethingsnetwork.org/