Real-Time Demand-Response using NB-IoT

Ashish Kumar Sultania*, *Student Member, IEEE,* Farouk Mahfoudhi*, and Jeroen Famaey, *Senior Member, IEEE*

***Abstract*—The Internet of Things (IoT) already connects billions of devices and keeps growing exponentially. These devices are designed to be integrated with industrial machines, home appliances and infrastructures. One such use-case is to build a smart grid management system that relies on demand-response techniques to control the appliances automatically so that the power can be distributed optimally. The mission-critical smart grid communications require secure, reliable, two-way communicable, and latency bounded connections between the management system and the electrical appliances. To realize these, cellular technology is arguably the most feasible solution. 3GPP has already released the Narrowband-Internet of Things (NB-IoT) standards as the low-power dense-area coverage IoT cellular solution. In this paper, we present an NB-IoT system to monitor and control the connected electrical appliances in a smart grid network. The platform is also capable of configuring the network dynamically. We assess the latency performance for our solution using the commercially available Orange network in Belgium. It is observed that NB-IoT enabled devices can be controlled and monitored with a maximum latency less than 8 seconds in the deep-indoor environment and within 2 seconds for the outdoor environment.**

***Index Terms*—Narrowband-Internet of Things, Smart grid, Demand response.**

I. Introduction

THE concept of the IoT is to connect everything to the Internet. It already connects billions of devices and keeps growing exponentially [1]. Typically, IoT use-cases target small-scale low-power devices such as sensors, actuators, detectors and switches that are deployed on the appliances, embedded in the infrastructure or on the body. We have considered the use-case of the smart grid, which is to deliver electricity from suppliers to consumers using two-way digital communications controlling appliances at the consumer site. The problem of power shortage happens during peak demand hours, and it can cause power outages. Therefore, the principle of demand response techniques can be used to manage the risks of power outages, reduce cost and increase reliability. The goal is to prioritize the electrical appliances usage based on customer's configurations and electricity supply to make optimal use of the energy sources and minimize the negative impact on the environment. Therefore, the power networks and information networks shall be integrated to create a Smart Grid network having bidirectional data, control and energy flows. By connecting the appliances with the system, they can be monitored or controlled remotely. There are many wireless technologies that can connect these appliances. The key requirements of a demand-response application includes bounded latency, wide range, high data rate, high reliability, security, scalability, and availability [2]. The architecture of the demand-response connectivity needs all the electrical appliances to connect with a long-distance Demand Response server (DRS). The DRS is responsible to provide control and visibility of the appliances. They exchange a few bytes of data usually less than 100 bytes including meter readings or control messages with the maximum latency requirement between 2 (for primary response) and 30 seconds (for secondary response) [3] [4]. However, some residential equipment can tolerate larger delay, while industrial equipment has tight delay constraints. Therefore, long communication range protocols and technologies can be used.

Existing standards for long-range communication, such as NB-IoT, Sigfox, and LoRa are known as Low-power Wide-area Networks (LPWANs). LPWAN targets applications that require the transfer of small amounts of data with long battery life and large network coverage. Sigfox has several commercial networks up and running as an ultra-narrowband-IoT technology. There exists a constraint of 140 packets/day and the maximum packet size of only 12 bytes in Sigfox [5]. It supports a maximum throughput of only 100 bps and the downlink communication (from server to the end device) can only occur following an uplink transmission. Operators also use the LoRa Wide Area Network Protocol (LoRaWAN) which is generally intended to operate in the unlicensed 433 MHz and 868 MHz bands with a minimum channel bandwidth of 125 kHz [6]. It supports data rates up to 50 kbps depending on spreading factor and channel bandwidth. However, it requires a gateway to be installed. In contrast, NB-IoT takes the lead with existing deployed infrastructure of LTE. Moreover, the downlink communication in Sigfox and LoRaWAN Class A and B end devices can only possible after an uplink transmission which is a bottleneck for low downlink latency requirements. LoRaWAN Class C end devices support simultaneous bi-directional communication but at the expense of excessive power consumption [7]. NB-IoT works seamlessly with other LTE services and is optimized for lower power consumption, deeper coverage, reliability, and higher device density. It has faster response times than LoRa and can guarantee a better quality of service. The data rate is limited to 200 kbps for the downlink and 20 kbps for the uplink [7]. NB-IoT claiming lower latency makes it suitable for latency-sensitive use-case of demand-response. NB-IoT relies on the cellular network and so it shares an excellent level of security and availability. It supports up to more than 50 000 devices per cell enables scalability of the system. Therefore, we can say that NB-IoT can be considered as a good candidate for demand-response

*Co-primary authors

Ashish Kumar Sultania, and Jeroen Famaey are with University of Antwerp and imec, Belgium, Farouk Mahfoudhi is with Nokia Networks, Belgium (e-mail: {ashishkumar.sultania, jeroen.famaey} @uantwerpen.be), mahfoudhi.farouk@gmail.com

applications.

In this paper, we present a complete end-to-end NB-IoT based architecture to connect, monitor and control electrical appliances as a demand-response application. The proposed architecture does not depend on any other technology to connect the plug with the server or cloud. As a case study, we performed tests to evaluate the latencies for reporting Smart Plug data to the DRS and control messages from Demand Response Controller (DRC) to Smart Plug considering various environmental conditions (indoor, outdoor) for the Smart Plug. We have also presented communication latencies for payloads of different sizes using NB-IoT. We introduced the future directions for building smart IoT solutions to control or monitor the appliances based on the proposed architecture over NB-IoT.

The outline of this paper is as follows. In Section II, we provide an overview of the related literature. In Section III, we present the proposed system architecture. The evaluation of the system performance is discussed in Section IV and conclusion is presented in Section V.

II. Related Work

There are some existing simulator and testbeds to analyze the communication protocols of smart grids. Song et al. [8] proposed a wireless SG lab (named SmartGridLab) to design and analyze new protocols for both power system and communication network aspects in a lab environment. Other researchers also built simulators to study smart grid systems such as GridLAB-D [9] and GridSim [10]. A ZigBee network-based testbed with WiFi is also presented for smart grid applications [11]. But these tools have limited capability for analyzing real scenarios. Also, a residential demand-response system supports data transmission between the appliances and DRS is demonstrated by Strobbe et al. [13]. They connect the appliances creating a local network using ZigBee and Power line communication (PLC) and these appliances communicate to the DRS via a gateway that is connected via Ethernet network. However, PLC often shows intermittent failures for large number of operating appliances and the high power consumption of ZigBee network opens a path to try few other alternatives such as Z-Wave. A Z-Wave monitoring system has been developed by Gong et al. [15] and a controller by Wei et al. [16]. But to connect the system to the Internet most of the proposed architectures use power-hungry technologies such as WiFi. Also, ZigBee uses the same frequency band as WiFi, this can result in radio interference.

Long range communication technologies are also used to connect devices directly to the Internet. Joris et al. [17] presented a system based on Sigfox where autonomous sensor nodes have the capability to transmit data collected by a range of sensors directly to the cloud. Some researchers also used multiple technologies to provide solutions for a smart management system. For example, Xu et al. [18] have designed an intelligent street lamp control system based on NB-IoT and LoRa. The key data and instructions are transmitted in real-time using NB-IoT. LoRa is used to control the street lights' power. Based on a LoRaWAN Smart Plug, Soe et al. [19] proposed an energy management system for smart homes. In [20], a secure system based on LoRa technology was proposed for smart metering. Recently, a LoRa, PLC and optical fiber based demand response system was also proposed by Ku et al. [21] where LoRa or PLC are used for communication between appliances and a central entity. This central entity then communicates with DRS via optical fiber communication system. A coverage analysis [22] for NB-IoT and LoRa has been provided for Smart Grid applications in real scenarios and it is concluded that NB-IoT out-performs for deep indoor coverage. Li et al. [2] demonstrated that NB-IoT perfectly satisfies most of the quantitative and qualitative requirements, such as reliability, security, and scalability. They evaluated the reliability of NB-IoT for smart grid communication environments via Monte Carlo simulations which demonstrated that it works well in all typical communication scenarios (i.e. rural, urban, and hilly terrain) in the smart grid. Most of the existing work on NB-IoT focuses on the analysis based on analytical or simulation calculations rather considering real networks [23], [24]. As per our knowledge, no architecture has been proposed particularly for demand-response applications and evaluating the system performance on a live NB-IoT network. However, Xiong et al. [25] proposed a real-time active NB-IoT based network design to collect power failure data of smart meters. Some other real-time monitoring systems have been built using NB-IoT. Anand et al. [26] presented a mechanism for remote monitoring of water levels in storage tanks using NB-IoT. For hospitals, Zhang et al. [27] have designed an infusion monitoring system to monitor the real-time drop rate and remaining drug volume during the intravenous infusion using NB-IoT.

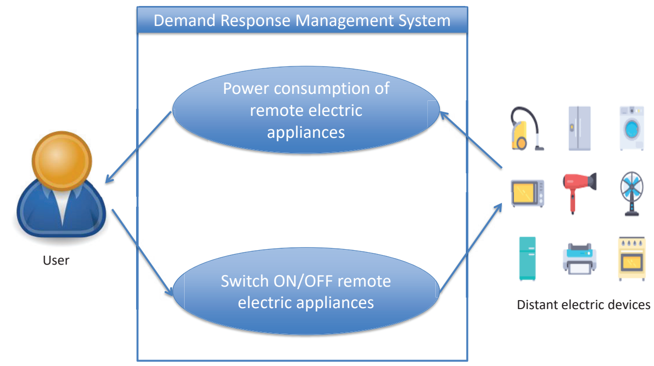

Fig. 1: Use cases Diagram of the Demand Response System

In order to optimize energy consumption and minimize the negative impact on the environment, we have designed a demand-response energy management system based on NB-IoT. This increases energy efficiency using the intelligent control system and by monitoring the electrical appliances. We have connected the DRC and the appliances both using the NB-IoT network. This helps in removing the dependency on other technologies to connect to the Internet or server.

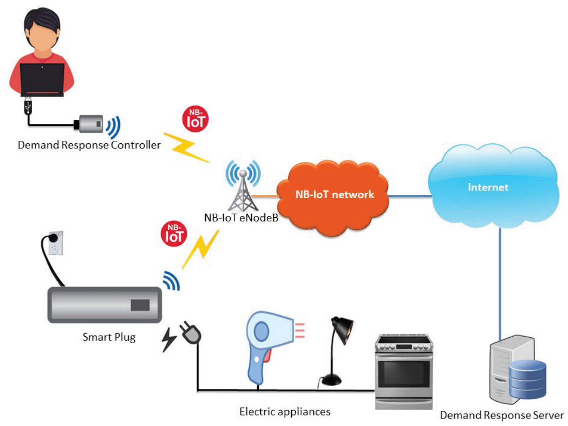

Fig. 2: Network Architecture

III. Proposed System Architecture For Demand-Response

In this section, we propose our system architecture to monitor or control electrical appliances in real-time using NB-IoT. The electrical appliances' are connected to the DRS using a Smart Plug. The use case diagram is shown in Figure 1 where users or an automated system can monitor the power consumption of distant electrical appliances and can switch them ON or OFF in real-time.

In a demand-response system, appliances might be installed in deep indoor areas and do not transmit large packets and therefore low-powered NB-IoT can be used to connect these electrical appliances to the grid management network. The link budget of NB-IoT enables it to reach the electrical appliances installed in low signal locations such as basements, tunnels, and remote rural areas. We designed a distributed network architecture, as shown in Figure 2. It consists of three layers namely device layer, network layer and application layer. The device layer consists of sensors and has the responsibility to sense the environment, collect data, and control appliances. This layer also has a communication module which connects the sensors with the network layer. The network layer acts as a bridge between the device layer and application layer. We propose NB-IoT as a method for the transmission. Lastly, the application layer has the responsibility to provide the services to the demand-response applications. The three-layered architecture for IoT based smart grid systems has been widely used in the literature as already described in [28] and [29]. In this architecture, the distributed power suppliers and consumers are connected and any new supplier or consumers can easily be added into the system. There are four main agents in the system:

- **Smart Plug**: This is the monitoring unit of the system installed at the customer site. In our use case, it is the device that measures the power consumption of the plugged electrical appliances and periodically sends the real-time measurements to the DRS using the long-range NB-IoT network. This unit consists of a sensor module used to measure power consumption and an NB-IoT module to send the measured values to DRS using the NB-IoT network. Each Smart Plug installed in the system can be simply identified by the International Mobile Equipment Identity (IMEI) which is uniquely assigned by the NB-IoT network operator. Usually, the network operators issue SIM cards for the devices. It also performs on/off operations of connected appliances based on the instructions received from the DRC. These instructions can also be modified and represented as levels such as for a fan it could be described as low, normal, high, and off.
- **Demand Response Server**: It is used to maintain the database of the system. It can be a self-deployed server connected to the Internet or a cloud service. The Smart Plugs when booted up, first register themselves to the DRS. It can be done by negotiating security protocol messages using the unique IMEI number. The location of the plugs can be pre-assigned by the moderators or otherwise can be sent by the Smart Plugs. The DRS maintains the logs of power measurements, plug states (ON/OFF) and recent IP addresses, which are sent periodically by the Smart Plugs via NB-IoT. It also maintains all the requests from the DRC.
- **Demand Response Controller**: The DRC application can fetch the required data from the DRS to decide the actions to perform for different Smart Plugs. Generally, the DRC periodically collects Smart Plug information from the DRS, which is called *load profile*. Its application considers customer load profile and available energy to manage the system overall power distribution. It sends control instructions to the Smart Plugs to power up or down some of the connected appliances. This application could also be controlled manually by the user or administrator. The administrator can monitor any specific Smart Plug based on its unique identifiers such as location, IP address or IMEI number and send instructions to the Smart Plug. Its application can be installed in a separate machine such as a Laptop, Raspberry Pi, Mobile device or the external server running a mobile or web application, which is connected to the Internet. This would provide flexibility to the users to access the system. However, to send control instructions to the monitored-appliances, the access of the Smart Plug is required from an external network. Many technologies could be potentially used for this external network, such as Wi-Fi, GPRS, LTE, or wired LAN. However, they all have important disadvantages. Wi-Fi and wired LAN need their users to install and maintain the network which is an additional overhead. Whereas GPRS and LTE provide similar security as NB-IoT but they struggle to provide sufficient indoor coverage and have significantly higher subscription costs. [30] [31]. Moreover, accessing from an external network creates a challenge for many network operators to enable the firewall and maintaining the NAT table for a long time to provide the access. This can be avoided by assigning each Smart Plug with IPv6 addresses but that is expensive and being adopted slower. Therefore, to avoid this issue,

we propose a controller to be another NB-IoT module. Additionally, this provides the application developer with flexibility, security and a system that does not rely on the Internet to access the appliances.
- **User Interface**: It is the web interface or mobile application for the users or administrators to monitor the appliances' energy consumption remotely, which shows power consumption curve over time or actions for the appliances.

A. System Prototype

We built the prototype by using a reliable device widely used in home automation and able to measure the power consumption of electric appliances plugged into it. The Fibaro Wall Plug is chosen to measure the power consumption of electric appliances which is already a commercially successful product in the market. The possibility of connecting more than one wall plug in a closed area where, each plug has a unique identification number makes the Fibaro Wall Plug a better choice. It uses short range Z-Wave technology to exchange information between the socket and dongle. The long distance between each appliance and the DRC can be covered by low-cost LPWAN technology. We use the NB-IoT infrastructure offered by Orange in Belgium for the prototype. The Smart Plug contains the following key components:

- **Fibaro Wall Plug**: It measures the power consumption of plugged electric appliances and reports it via Z-Wave. It also switches ON or OFF the power of the appliances, for which it polls for the instruction of its controller.
- **Z-Wave Dongle**: It is used as a controller of the wall plugs. It reads all the measurements from the Fibaro Wall Plug and can send them Switch-ON or -OFF commands using Z-Wave.
- **NB-IoT module**: It sends the power measurements to the DRS and receives switch ON/OFF commands from other DRCs via NB-IoT. We used the SARA-N210 [32] based module as NB-IoT device (LTE Cat NB1).
- **Raspberry Pi**: It hosts and executes the application that manages the NB-IoT and Z-Wave communication between different modules.

All these components of the Smart Plug are shown in Figure 3. The shown prototype is used only as a lab environmental setup and can be miniaturized for commercial use. Additionally, a distant DRC is considered which is an NB-IoT module connected through an FTDI cable to a Linux based device running its application. This application can provide a graphical user interface or command prompt to interact manually with a user. The controller is used to receive all the load profiles of the connected Smart Plugs from the DRS and send instructions for switching ON or OFF commands for appliances using NB-IoT.

B. Message Exchange

Mainly the messages in the system are exchanged for controlling and monitoring. These message flows are described in the following subsections.

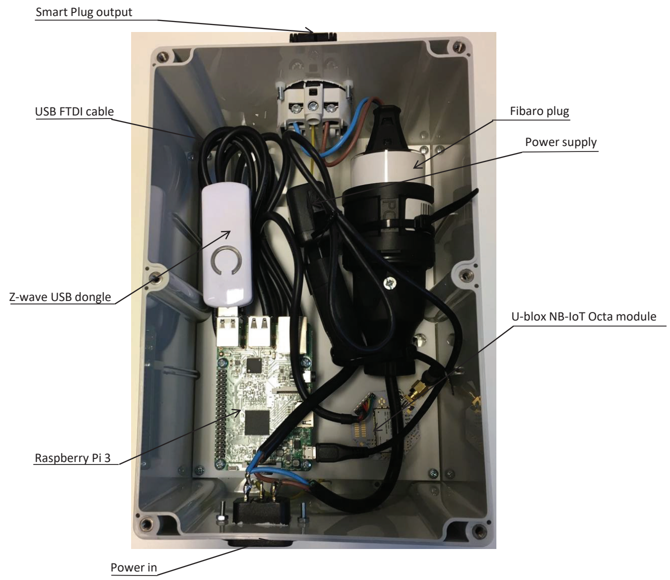

Fig. 3: Smart Plug Prototype

1) Reporting Power Consumption: The different actors involved in reporting power consumption of plugged appliances by the Smart Plug include the user, Smart Plug, plugged appliances and external server. The sequence diagram is shown in Figure 4. When the Smart Plug boots up, it initializes its internal components including the power measuring module and the NB-IoT module. It (re)establishes a connection to the NB-IoT network. Once the initialization is finished, it sends a bootstrap message to the DRS using NB-IoT. These messages include the parameters such as IMEI number of the NB-IoT SIM card to use it as a unique identifier of the Smart Plug, IP address obtained by it after establishing NB-IoT connection, current state of the smart plug, and the current time-stamp which is synchronised for both the Smart Plug and the DRS with a single Network Time Protocol (NTP) server. The DRS is configured to listen on a specific User Datagram Protocol (UDP) port. Once it receives a bootstrap message from any Smart Plug, it parses the content and stores the data corresponding to the Smart Plugs Identities database.

The Fibaro Wall Plug starts the measurement at every default time interval. This interval can be changed using the DRC by updating the configuration. If the measurements are nearly constant in a particular defined range, the Smart Plug reports them periodically at each defined time interval. It is also configured to send the report at each power value change. The measurement message towards the DRS includes IMEI, the measured power consumption value, current state, and time-stamp. The DRS, once it receives a measurement message from a Smart Plug, parses, and stores the data in the *Measurements* database.

2) Monitoring Power Consumption: A user can monitor the electric appliances plugged in a specific Smart Plug using a web interface accessible via the Inter- or intranet using any browser. The sequence diagram for monitoring power consumption is described in Figure 5. Mainly, it follows the following steps:

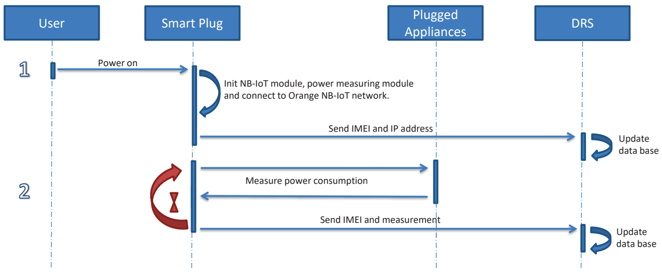

Fig. 4: Sequence Diagram to Report Measurement

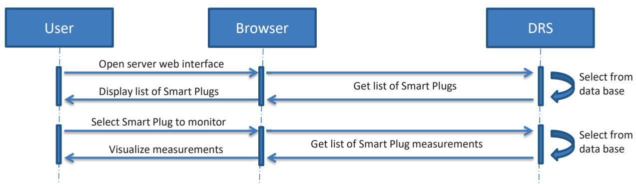

Fig. 5: Sequence Diagram to Monitor Smarts Plug

- The list of connected Smart Plug IMEIs is fetched from DRS database and displayed to the browser.
- When a user selects any specific Smart Plug to monitor, the corresponding measurements are fetched from the DRS database and power consumption over the time is displayed.

3) Sending Instructions to a Smart Plug: The instruction is sent to set the power socket status or to change the configuration of the Smart Plug. Figure 6 defines the exchanged communication between the different agents to manually switch ON or OFF a Smart Plug. When a user executes the controller application, it re-initializes the connected NB-IoT module and establishes a connection to the NB-IoT network. Once finished, it receives the list of connected Smart Plugs and it follows the following steps.

- The DRC requests DRS for the total number of connected Smart Plugs already registered accessible for the logged-in user.
- The DRC sends individual requests to the DRS for getting the IMEI, the IP address and the state of the Smart Plugs.
- The list of IMEIs and states of Smart Plugs can be displayed to the user.

Now a user can choose a specific appliance connected to the Smart Plug to be switched ON/OFF. This instruction is sent directly to the Smart Plug via NB-IoT. When the Smart Plug receives the control message, it sets the plugged electric appliance's status. The manual intervention of the user is optional, the DRC can also automatically send instructions based on the users' load profile.

C. Message Format

The communication between the Smart Plug, DRC and DRS is established using UDP sockets. DRS can be configured to listen on a particular UDP port. We define a specific command format for each message exchanged over the NB-IoT channel. They are presented in Table I. However, these messages can

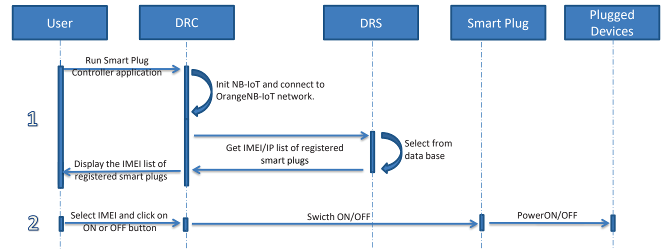

Fig. 6: Sequence Diagram to Set Smarts Plugs Status

be replaced with standard application layers format defined by OpenSG [33].

D. Software Architecture

In this section, we describe the software stacks of each entity of the system. As mentioned before, these consist of Smart Plug, DRC and DRS.

1) Smart Plug: The key software components of the Smart Plug program that can be hosted on a Raspberry Pi include Raspbian, standard serial library, serial modem controller openzwave, SARA-N210 module driver, and Fibaro Z-Wave controller. Additionally, we have developed python scripts that initialize all the modules, receives measurements from the Fibaro Wall-Plug, forwards them to DRS and handles received instructions over NB-IoT. These scripts run as daemons.

2) Demand Response Controller: We developed a graphical user interface (GUI) based application for a DRC. It is a Qt application running on a system connected to the NB-IoT module via an FTDI cable. The software entities of the controller are Qt API for creating GUI, serial modem controller, and SARA-N210 NB-IoT driver. The controller application is written that initializes the NB-IoT module, receives the Smart Plugs' information from DRS and sends instructions over NB-IoT to the Smart Plug selected by the user.

3) Demand Response Server: The software architecture of DRS that is hosted on a dedicated machine or cloud machine publicly accessible via the Internet is described in Figure 7. The configuration and libraries required to configure DRS are socket, mysql connector, mysql server, and database. The database stores Smart Plugs' information in two tables namely *Plugs* and *Measurement*. The first one stores IMEI, IP address and state of a Smart Plug with the timestamp which are received when the Smart Plug connects to the NB-IoT network and the later stores IMEI, power, state of a Smart Plug when they are reported as a measurement at a particular timestamp. IMEI number is used as the Smart Plug's identifier. The DRS application is written that listens on UDP sockets and parses the received messages from Smart Plugs and Smart Plugs controllers. It runs as a daemon and so becomes active on booting up the DRS.

The interaction of these components at booting-up a smart-plug follows the following actions:

- The Fibaro Plug Z-Wave controller class is initialized and defines a *powerMeasurementCallback* function that

TABLE I: NB-IoT Messages Format

Source	Destination	Message Type	Message Format
Smart Plug	DRS	Smart Plug identity	Plug#<IMEI>#<IP>#<State>#<Timestamp>
		Measurement	Measurement#<IMEI>#<Power>#<State>#<Timestamp>
DRC	Smart Plug	Switch ON an appliance	Switch#ON#<ApplianceID>
		Switch OFF an appliance	Switch#OFF#<ApplianceID>
DRC	DRS	Get registered number of Smart Plugs	GetPlugsNumber#
		Get details of registered Smart Plugs	GetPlug#<identifier>
DRS	DRC	Send registered Smart Plugs number	PlugsNumber#<Count>
		Send details of registered Smart Plug	Plug#<IMEI>#<IP>#<State>

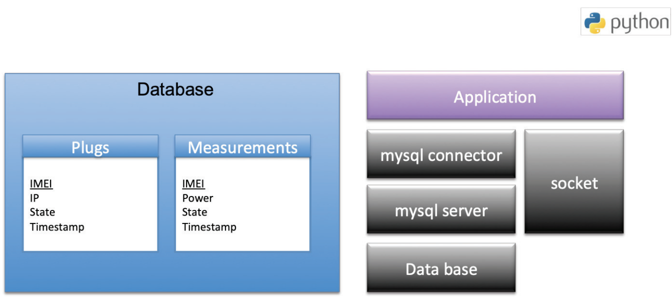

Fig. 7: DRS Software Architecture and Database Schema

is called when receives a power consumption measurement. This class uses the openzwave library to manage communication with the Fibaro Wall-Plug over Z-Wave using the Z-Wave USB stick.

- The state of Fibaro Plug is received by calling the *getPlugState* API from the Fibaro Plug Z-Wave controller class.
- The NB-IoT driver class of SARA-N210 is initialized and defines an *nbiotRxCallback* function that is called on receiving any messages from the NB-IoT network. This class uses the Serial modem controller class to communicate with the NB-IoT module via UART. It uses the serial library to control the serial ports.
- The NB-IoT module is configured and connected to the NB-IoT network by calling the *initialize* API from the SARA-N210 NB-IoT driver. This API returns the connection status, obtained IP address, and IMEI received from the NB-IoT module.
- IMEI, IP address, plug state, and current timestamp are sent to the DRS via NB-IoT network using an API *send* from the SARA-N210 NB-IoT driver class.
- To get the power consumption value of the plugged appliance the API *getPower* is called periodically from the Fibaro Plug Z-Wave controller class. IMEI, power, plug state, and current timestamp are sent to DRS via NB-IoT by calling the *send* API from SARA-N210 NB-IoT driver class. When the power consumption changes, the *powerMeasurementCallback* is called and the application again sends the measurement to DRS.
- When a message is received from the NB-IoT network, *nbiotRxCallback* is called. The application calls the *switchOn* or *switchOff* APIs from the Fibaro Plug Z-Wave controller class if the received instruction is a switch ON or OFF command respectively.

Whereas, when running the GUI application, the following actions are performed by the DRC:

- Similar to the boot-up process, the SARA-N210 NB-IoT driver class is initialized and calls the *nbiotRxCallback* function at arrival of a payload from NB-IoT network. Also, the NB-IoT module is configured and calls the *initialize* API.
- A request for the Smart Plug information from the DRS is sent over NB-IoT using the messages as shown in Table I through the *send* API from SARA-N210 NB-IoT driver class.
- The callback function *nbiotRxCallback* is called on receiving a response from the server and the information is stored.
- The connected plugs are displayed to the user using the Qt widgets.
- When the user selects a Smart Plug to switch ON/OFF, a message is sent directly to the Smart Plug via NB-IoT using the *send* API from the SARA-N210 NB-IoT driver class.

At the DRS, its application listens on a UDP port using the socket library APIs. The messages are parsed on arrival from a Smart Plug or a DRC and are inserted/selected into/from the database using the mysql connector API.

E. System Challenges

With the benefits of NB-IoT such as fast deployment time, better network coverage, sufficient bandwidth, and low cost, various system challenges can rise to integrate NB-IoT in smart grid deployments. We list the most important ones below.

- Portability and cost: It is required to design a cost-effective small and portable smart plug design. It is important to integrate it into existing appliances. In this paper, we have presented our prototype model which can be further reduced in size.
- System scalability testing: As an end-user of the NB-IoT network, it is difficult to get the real-time configuration of the base station. It is important to evaluate the scalability of the network.
- Network addressing: It is needed to assign unique IP addresses to the smart plugs to connect them to the Internet. It is observed that currently many NB-IoT networks assign private IPv4 addresses. But due to shortage of the IPv4 address space, the network needs to be updated to support IPv6.

- Availability: NB-IoT had been designed with the IoT use cases in mind, supporting more than 50 000 devices per cell. However, the services of the network are shared with other network service domains which can decrease the network performance. Therefore, a requirement of new scheduling techniques for prioritizing demand-response data for an NB-IoT eNode-B could be considered.
- Security and Privacy: Many security concerns must be considered for demand-response applications such as access control, and data integrity. Being an LTE based network, NB-IoT is robust against security threats. However, it can be possible that the conventionally available security measures are not completely sufficient to handle a smart grid system. On the other hand, Sigma Designs also continuously works to improve the security of Z-Wave [35]. The last updated security framework was announced in 2017 and so it is needed to be investigated more in the future. In this work, we have not focused on the security aspects.

IV. Evaluation

The performance evaluation of the system is performed in terms of latency, data delivery percentage, and command delivery percentage. We calculate the latency for monitoring an appliance for its power measurements and to set its status using the Smart plug prototype setup. As our setup makes use of a public NB-IoT network in Belgium, the latency value received includes the grant acquisition, Random access, signal processing, queuing delay due to congestion, propagation delay, packet re-transmission delay and processing time. The experiments are performed in an outdoor environment and then the attenuator is added to represent building conditions. An attenuation of 0 dB represents the outdoor or close to window conditions, 30 dB represents the inside of a building shielded by one wall, and 60 dB is for deep indoor locations such as basements or inside an elevator. During the experiments, the average best Received Signal Strength Indicator (RSSI) received is observed to be -90 dBm. It should be noted that the experiments assumed that the real-time network performance changes during different times of the day. One of the important factors that affects the results could be a real-time network load. Since the Fibaro wall plug communicates with the local controller over Z-Wave, interference could affect end-to-end performance. However, since they are positioned a few centimeters apart in a closed box, the signal strength is strong, and no noticeable negative effects were observed during the performed tests.

Figure 8 shows the average latency for reporting measurements of 64 bytes to the DRS. The representative latency values in this figure considers the Smart Plug's identity message of 64 bytes and a Z-Wave controller is used to collect the measurement. It can be observed that under good conditions the message of 64 bytes from a Smart Plug can be sent in less than 1.5 seconds and in the worst environmental conditions in less than 8 seconds. Similar behaviour can be observed in case of setting the appliance's status as shown in Figure 9. For the latency calculation in Figure 9, it is assumed that the size of

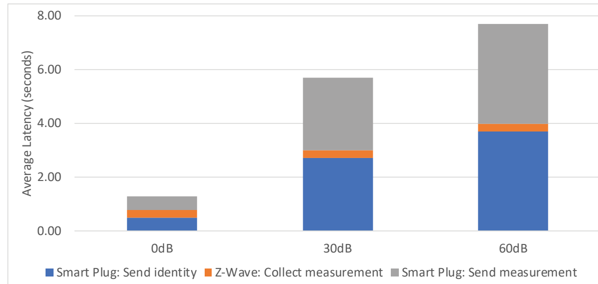

Fig. 8: Communication Latency of Reporting Measurement

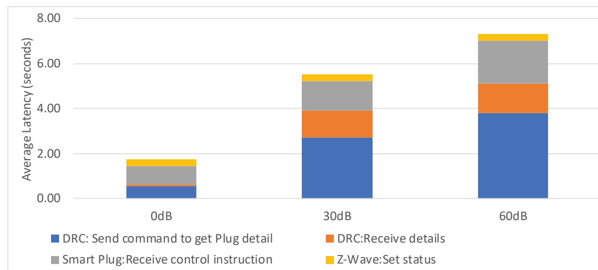

Fig. 9: Communication Latency of Setting Appliance's Status

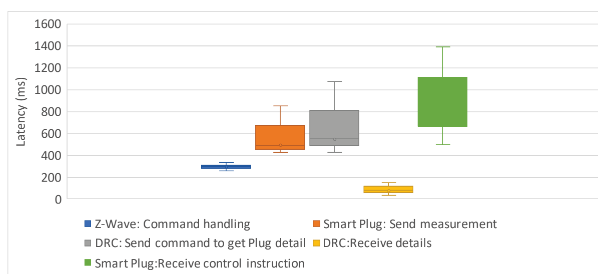

Fig. 10: Min-Max Latency Summary of Different Activities at 0 dB Attenuation

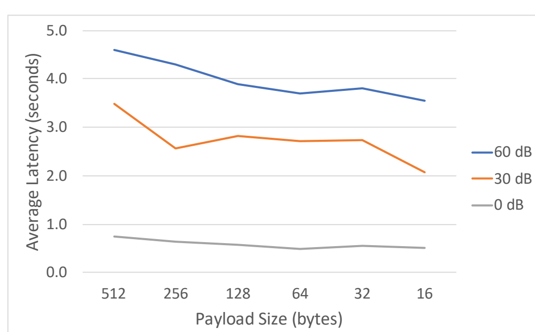

Fig. 11: NB-IoT Latency from Smart Plug or DRC to DRS

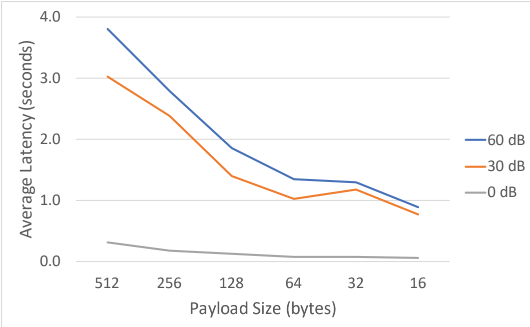

Fig. 12: NB-IoT Latency from DRS to DRC

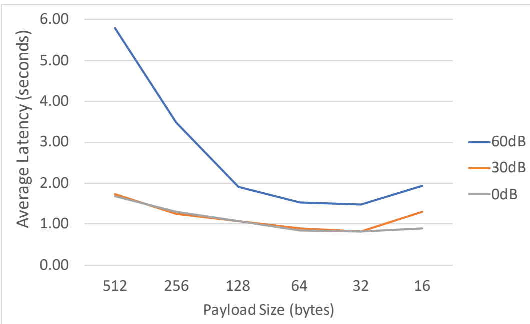

Fig. 13: NB-IoT Latency from DRC to Attenuated Smart Plug

the request sent and the received load profile is 32 bytes and for the control instruction 16 bytes. We also observed that the Z-Wave gateway needs only around 300 ms to execute a command to switch ON/OFF the appliances. Therefore, in addition to the average latency of NB-IoT, we also add this latency into our calculation. The latency variation is shown in Figure 10. The latency of the Z-Wave gateway to successfully handle the commands varies from 260 to 332 ms. The latency variation is maximum when the smart plug receives the control instruction which is from 495 ms to 1390 ms. Therefore, if we consider the maximum latency of all the activities, the communication latency in changing the appliance status increases to around 3 s instead of 1.8 s at 0 dB. Whereas, the latency in reporting the measurement increases to around 2 s from the average value of 1.3 s in the worst case scenario.

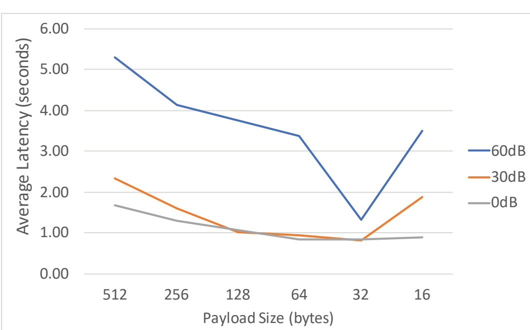

Fig. 14: NB-IoT Latency from Attenuated DRC to Smart Plug

We have also evaluated the latency for different payload sizes and for different signal strengths. Figure 11 shows the communication latency of the data transmission using NB-IoT network when the Smart Plug sends power measurement to DRS or when the plug data request is sent via the DRC to DRS. Attenuated at 60 db, the latency reaches up to 4.6 seconds for 512 bytes payload, and is more than five times lower for 0 dB attenuation. Whereas, when the data is sent from the DRS to the DRC (such as sending registered smart plug details), the latency is observed to be lower at 3.8 seconds for a deep indoor scenario (as seen in Figure 12). It decreases more than five times for 0 dB when the payload size is decreased from 512 to 16 bytes and takes maximum around 320 ms. The overall trend of the graphs is decreasing which means the latency decreases with payload size. However, the rate of decrease for the data transmission towards the DRC is more than when it originates from it for the DRS. The latency in sending an instruction from an NB-IoT enabled DRC to an attenuated Smart Plug is shown in Figure 13 and from an attenuated DRC to a Smart Plug in Figure 14. A significant increase in latency is visible for 60 dB attenuation compared to 0 or 30 dB attenuated scenarios. The lowest value of latency is found at 32 bytes data which is around 1.5 seconds when attenuated Smart Plug receives and 1.32 seconds when attenuated DRC sends. However, at 0 dB latency values in both the scenarios are almost the same, varying from 1.67 seconds (512 bytes) to 0.83 seconds (32 bytes). At 16 bytes data size in Figure 13 and 14, we observed a slight increase in the latency. There is a high possibility that it is because of the signalling overhead caused due to aggregation and reassembly of the small consecutive packets. The packet aggregation aims to improve the system efficiency by making a better use of the resources allocated to the device. The bundled packet needs wait to receive uplink grant so that it can be sent on the uplink data channel. From 64 bytes to 32 bytes, the latency does not vary much and remains around 0.85 seconds. Most importantly, we received all the packets over the NB-IoT network without any loss, which also shows high reliability of the network. The packet delivery rate is measured through the sequence number of the total packet sent and received.

The communication latency requirements of demand-response applications depend on their purposes. Many European countries require maximum 30 seconds latency for secondary response applications, which is observed to be easily satisfied by NB-IoT. However, Ireland and UK require a fast primary response of 2 seconds. This can be achieved using NB-IoT when the appliances can achieve a high link budget (i.e., 0 dB attenuation), which corresponds with outdoor or close-to-window indoor conditions under good coverage. Moreover, the real-time demand-response service mode requires a maximum delay of 10 seconds [34]. Given this requirement, the NB-IoT technology can deliver the data for one-way communication under all environmental conditions (cf., Figure 8 and Figure 9). Such one-way communication can be used to collect the meter reading periodically for billing purposes, or change the appliance status. Such communications typically comprises less than 100 bytes and can be delivered in less than 4 s with high reliability of more than

99% as observed in Figure 11.

Our results can be compared to the ones obtained by other state-of-the-art demand-response systems. It should be noted that it is difficult to compare them fairly due to the differences in architecture, topology, hardware, communication technology and device density. However, we compare our result with a similar architecture as defined by Viswanath et al. [36]. They have developed a cloud-enabled system with a gateway. The gateway enabled with a Z-Wave module is responsible to communicate with the devices. They have used RESTful HTTP technology to send the sensor data from the gateway to the cloud server. The setup is established such that the server and the controller are on one local area network (LAN) and the gateway on another LAN. The latency to send data from the controller to server takes 0.071 s, and from server to gateway 0.34 s. The gateway takes an average of 0.78 s to control a node and receive an acknowledgement. By comparing these results with our system at 0 dB we can infer that our device takes around 1.18 s to receive and set the status. However, the other architecture (at [36]) takes around 1.12 s to do the same. The end-to-end latency of their proposed system is slightly lower. However, they use a private WLAN combined with a wired backbone. In contrast, our solution uses a long-range and fully wireless access network (NB-IoT) that has many advantages over a wired gateway, such as easy device installation, freedom in choosing location, support of a large number of end devices, large coverage, and deep indoor penetration. Moreover, their solution targets communication only in one direction that is towards the devices. We have targeted two-way communication, which helps in receiving the device status on-demand.

V. Conclusion

In this paper, we designed and implemented a complete end-to-end NB-IoT solution for a demand-response system where the principle of demand response technique is used to monitor and control the appliances. We demonstrated the feasibility of the solution by evaluating the latency performance in different scenarios on the public Orange NB-IoT network in Belgium. NB-IoT shows having a great potential for smart grid communication, with its features such as deep-indoor coverage, low power consumption, reliability, and large appliance density. The achieved performance of the system observed in the experiments suffices the requirements of primary and secondary response time. By assessing the reliability, response time and deployment time, we believe that it can provide a complete solution for a demand-response management system. Our prototype uses the commercially available Z-Wave solution to control the appliances. Practically, the NB-IoT communication latency values between different entities of the system are observed to be less than 6 seconds in different environmental scenarios. Further improvements at the controller side are planned to include the features such as price-driven real-time demand-response and appliances failure reporting.

Acknowledgment

Part of this research was funded by the Flemish FWO SBO S004017N IDEAL-IoT (Intelligent DEnse And Long range IoT networks) project and by the ICON project MAGICIAN. MAGICIAN is a project realized in collaboration with imec, with project support from VLAIO (Flanders Innovation and Entrepreneurship). Project partners are imec, Orange, Televic, Citymesh and REstore.

References

[1] Internet of Things (IoT) Market Is Growing Exponentially by 2022, https://www.marketwatch.com/press-release/internet-of-things-iot-market-is-growing-exponentially-by-2022-key-players-google-ibm-apple-cisco-oracle-and-hewlett-packard-million-insights-2020-01-27, last accessed 22nd April 2020.

[2] Li, Y., Cheng, X., Cao, Y., Wang, D. and Yang, L., "Smart choice for the smart grid: Narrowband Internet of Things (NB-IoT)," IEEE Internet Things Journal, vol. 5, no. 3, pp. 1505-1515, June 2017.

[3] Firm frequency response (FFR), https://www.nationalgrideso.com/balancing-services/frequency-response-services/firm-frequency-response-ffr, last accessed 2nd October 2019.

[4] Pitì, A., Verticale, G., Rottondi, C., Capone, A., Lo Schiavo, L., "The role of smart meters in enabling real-time energy services for households: The Italian case," Energies 2017, vol. 10,no. 199, pp. 1-25, February 2017.

[5] Sigfox system description, https://tools.ietf.org/html/draft-zuniga-lpwan-sigfox-system-description-03, last accessed 22nd April 2020.

[6] A technical overview of LoRa and LoRaWAN, https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf, last accessed 22nd April 2020.

[7] Mekki, K., Bajic, E., Chaxel, F., and Meyer, F., "A comparative study of LPWAN technologies for large-scale IoT deployment," ICT Express, vol. 5, no. 1, pp. 1-7, March 2019.

[8] Song, W. Z., De, D., Tan, S., Das, S. K., and Tong, L., "A wireless smart grid testbed in lab," IEEE Wireless Communication, vol. 19, no. 3, pp. 58-64, June 2012.

[9] Chassin, D. P., Schneider, K., and Gerkensmeyer, C., "GridLAB-D: An open-source power systems modeling and simulation environment," IEEE/PES Transmission and Distribution Conference and Exposition, pp. 1-5, April 2008.

[10] Anderson, D., Zhao, C., Hauser, C., Venkatasubramanian, V., Bakken, D., and Bose, A., "Intelligent design" real-time simulation for smart grid control and communications design," IEEE Power and Energy Magazine, vol. 10, no. 1, pp. 49-57, February 2012.

[11] Yi, P., Iwayemi, A. and Zhou, C. "Developing ZigBee Deployment Guideline Under WiFi Interference for Smart Grid Applications," IEEE Transactions on Smart Grid , vol. 2, no. 1, pp. 110-120, March 2011.

[12] Yiqi, W., Lili, H., Chengquan, H., Yan, G., and Zhangwei, Z., "A Zigbee based Smart Home Monitoring System," Fifth International Conference on IEEE Intelligent Systems Design and Engineering Applications (ISDEA), pp. 114-117, December 2014.

[13] Strobbe, M., Vanthournout, K., Verschueren, T., Cardinaels, W., and Develder, C., "Large-scale Residential Demand Response ICT Architecture," IEEE PES Innovative Smart Grid Technologies, pp. 1-6, October 2014.

[14] Wang, T., Zheng, B., Liang, Z.-L., "The design and implementation of wireless intelligent light control system base on Zigbee light link," 10th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), pp. 122-125, December 2013.

[15] Gong, J., Tan, C., Liu, L., and Zhou, L., "A new monitoring system of portable microcomputer injection pumps based on Z-Wave," Eighth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), pp. 19-21, June 2016.

[16] Wei, C.C., Chen, Y.M., Chang, C.C., and Yu, C. H., "The Implementation of Smart Electronic Locking System Based on Z-Wave and Internet," IEEE International Conference on Systems, Man, and Cybernetics, pp. 2015-2017, October 2015.

[17] Joris, L., Dupont, F., Laurent, P., Bellier, P., Stoukatch, S., and Redoute, J.M.m, "An Autonomous Sigfox Wireless Sensor Node for Environmental Monitoring, " IEEE Sensors Letters, vol. 3, pp. 1-4, Jul 2019.

[18] Xu, Y., Liu, S., and Peng, Y., "Research and design of environmental monitoring and road lighting system based on the Internet of things," IEEE Chinese Automation Congress (CAC), pp. 1073- 1078, December 2018.

[19] Soe, W.T., Mpawenimana, I., Fazio, M.D., Belleudy, C., and Ya, A.Z., "Energy Management System and Interactive Functions of Smart Plug for Smart Home," World Academy of Science, Engineering and Technology, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering, vol. 11, no. 7, pp. 884-891, 2017.

[20] Cheng, Y., Saputra, H., Goh, L.M., and Wu, Y., "Secure smart metering based on LoRa technology," IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA), pp. 1-8, January 2018.
[21] Ku, T.T., Chen, C.S., Lin, C.H., and Hsu, C.T., "Transformer Management System for Energy Control of Customer Demand Response and PV Systems," IEEE Industry Applications Society Annual Meeting, pp. 51-59, February 2019.
[22] Persia, S., Carciofi, C.and Faccioli M., "NB-IoT and LoRA Connectivity Analysis for M2M/IoT Smart Grids Applications," AEIT International Annual Conference, pp. 1-6, September 2017.
[23] Nair, V., Litjens, R., and Zhang, H., "Optimisation of NB-IoT deployment for smart energy distribution networks," EURASIP Journal on Wireless Communications and Networking, vol. 1, pp. 1-15, December 2019.
[24] Persia, S., and Rea, L., "Next generation M2M Cellular Networks: LTEMTC and NB-IoT capacity analysis for Smart Grids applications," AEIT International Annual Conference, pp. 1-6, October 2016.
[25] Xiong, D., Chen, Y., Chen, X., Yang, M., and Liu, X. "Design of power faliure event reporting system based on NB-IoT Smart meter", International Conference on Power System Technology (POWERCON), pp 1770-1775, November 2018.
[26] Anand, S. and Regi, R., "Remote monitoring of water level in industrial storage tanks using NB-IoT," International Conference on Communication information and Computing Technology (ICCICT), pp. 1-4, February 2018.
[27] Zhang, H., Li, J., Wen, B., Xun, Y., and Liu, J. "Connecting intelligent things in smart hospitals using NB-IoT," IEEE Internet Things Journal, vol. 5, no. 3, pp. 1550-1560, January 2018.
[28] Wang,C., Li, X., Liu, Y., and Wang, H. "The research on development direction and points in IoT in China power grid," in International Conference on Information Science, Electronics and Electrical Engineering, pp. 245-248, April 2014.
[29] Song,Y. E., Liu, Y., Fang, S., and Zhang, S. "Research on applications of the Internet of Things in the smart grid," in 7th International Conference on Intelligent Human-Machine Systems and Cybernetics, pp. 178-181, August 2015.
[30] Vejlgaard, B., Lauridsen, M., Nguyen, H., Kovacs, I., Mogensen, P., and Sorensen, M. "Coverage and Capacity Analysis of Sigfox, LoRa, GPRS, and NB-IoT," in IEEE 85th Vehicular Technology Conference, pp. 1-5, June 2017.
[31] Gandarillas, C., Iglesias, V., Aparicio, M., Mino-Diaz, E., and Olmos P. "A New Approach for Improving Indoor LTE Coverage," Broadband Wireless Access Workshop, pp. 1330-1335, December 2011.
[32] SARA-N2 series, u-blox, https://www.u-blox.com/en/product/sara-n2-series, last accessed 17 July 2019.
[33] Open Smart Grid - OpenSG , http://osgug.ucaiug.org/default.aspx, last accessed 19 August 2019.
[34] Transmission & Distribution Interface 2.0, https://www.nationalgrid.com/sites/default/files/documents/SDRC9.1%20Unrestricted%20document.pdf, last accessed 2nd October 2019.
[35] Z-Wave Alliance Continues Dominance, https://www.sigmadesigns.com/news/z-wave-alliance-continues-dominance-in-smart-security-category-with-new-members-products-developments-at-isc-west-2017/, last accessed 23rd April 2020.
[36] Viswanath, S.K., Yuen, C., Tushar, W., Li, W.T., Wen, C.K., Hu, K., Chen, C., and Liu, X. "System design of the internet of things for residential smart grid," IEEE Wireless Communication, vol. 23, pp. 90-98, October 2016.

Ashish Kumar Sultania received the M.Sc. degree in Computer Science from the University of Tartu, Estonia, and Norwegian University of Science and Technology, Norway in 2017 and B.E. in Information Technology from University of Delhi, India in 2011. He is currently a Ph.D. researcher with the University of Antwerp and imec. His research focuses on optimizing energy consumption of IoT devices and their networks. Prior to starting his masters, he worked as a Senior Software Engineer at NXP Semiconductor, India (2011-2015).

Farouk Mahfoudhi is a Senior Embedded Software Engineer at Nokia Bell Belgium, graduated from the Higher School of Communications of Tunis in 2012 in Telecommunication Systems Architecture. He has been working in different multinational industrial companies and research centers in Tunisia and Belgium for over eight years. He worked on embedded software development, network programming, IoT and home entertainment devices. In 2017, he also joined the University of Antwerp and the imec Research Center as a Software Development Engineer.

Jeroen Famaey is an assistant professor associated with imec and the University of Antwerp, Belgium. He received his M.Sc. degree in Computer Science from Ghent University, Belgium in 2007 and a Ph.D. in Computer Science Engineering from the same university in 2012. He is co-author of over 120 articles published in international peer-reviewed journals and conference proceedings, and 10 submitted patent applications. His research focuses on performance modeling and optimization of wireless networks, with a specific interest in low-power, dense and heterogeneous networks.