Residential Wireless Interfaces Virtualization: a Feasibility Study

Antonio da Silva Fariña,
Ana Belén García Hernando
Dept. of Telematic and Electronic Engineering
Universidad Politécnica de Madrid
Madrid, Spain

antonio.dasilva@upm.es,anabelen.garcia@upm.es

Mary Luz Mouronte López
Universidad Francisco de Vitoria
Madrid, Spain

maryluz.mouronte@ufv.es

Abstract—**This paper investigates the possibility of virtualizing and distributing the functionality that runs on top of residential wireless communications. Specifically, we propose, describe and test a solution that transports USB communications to remote locations, for scenarios in which the in-home wireless interfaces are consumed at the server side through this type of general-purpose and widely used interfaces. We frame this study in a general architecture by which Software Defined Networking (SDN) and Network Functions Virtualization (NFV) bring economies of scale, flexibility and programmability to residential Internet of Things (IoT) environments. As a result of our tests, we prove the feasibility of the remote presence of the IoT systems through the Universal Serial Bus (USB) tunnels, and we obtain approximate bandwidth measurements that serve as a hint on the type of services that can be offloaded to the cloud. For those functionalities that would need more bandwidth, we propose to embed a lightweight virtualization environment in home and to execute in it part of the virtualized components, something that is in line with the recent fog computing approaches.**

*Keywords-**IoT virtualization; SDN; NFV; residential wireless communications; USB/IP.***

I. INTRODUCTION

Nowadays application-layer gateways are needed to provide connectivity to IoT devices in the home. Current gateways mix network connectivity, in-network processing, and user interface functions. We share the view of [1] by which separating these functions would improve the connectivity potential for IoT devices. To help in this separation, two new trends, namely SDN and NFV, can be considered.

In fact, both SDN and NFV [2] are recently beginning to be proposed for the home environment. With SDN the control plane (in which the logical procedures supporting the networking protocols and the most important decisions are carried out) is separated from the data plane (in which the forwarding of packets on the most suitable interface towards the intended destination is carried out). SDN is an excellent mechanism to do Traffic Engineering (TE) and exploit effectively the network resources in an IoT scenario. NFV leverages commodity storage, networking and processing equipment in order to execute, through the use of a virtualization layer (sometimes called hypervisor), sophisticated network functionality on top of a virtualized infrastructure. It may be used to combine the available resources in a network by dividing the available bandwidth into channels or slices, each of which is independent from the others. NFV allows multiple service providers to carry out multiple separate and isolated virtual networks by sharing physical resources.

The main use cases of SDN and NFV in the home deal with pure networking tasks, and more specifically with the virtualization of the Customer Premises Equipment (CPE). There are also some recent proposals to augment the scope of cloud computing, NFV and SDN and integrate some IoT (basically sensing and actuation) capabilities into their frameworks [3] [4].

In our previous work [5], we proposed to leverage the virtualization possibilities of NFV together with the programmability of SDN in order to offer a portfolio of IoT-related functions to the residential users. Specifically, we aimed at the existence of a set of remote and virtualized “Plug-and-Play” (PnP) functions in order to recognize and manage IoT hardware belonging to different manufacturers.

We propose to have a generic and programmable gateway at home, called Home Radio Head (HRH), which does not need to implement any IoT vendor-specific function above the wireless communications provided by USB dongles. USB interfaces would be virtualized and managed remotely thanks to the establishment of tunnels between the HRH and the ISP. At the remote end of these tunnels, a NFV infrastructure hosts the IoT applications and management functionality implemented as a set of Virtual Network Functions (VNFs).

The virtualization of USB interfaces allows reaching economies of scale by offering a reasonably inexpensive customer premises equipment supporting most home wireless communications. In addition, we propose the use of a SDN-programmable data path in order to achieve greater flexibility in the management of the USB tunnels. Furthermore, in order to reduce the necessary bandwidth between home and the ISP, some processing can be carried out inside the customer premises by downloading and running lightweight VM containers. The rest of the paper is organized as follows: Section 2 summarizes the background that forms the basis of our work and some related works. Section 3 describes our proposed architecture and its building blocks. Section 4 explains the experimental setup we have implemented and discusses the results. Finally, in section 5, we provide some conclusions and our foreseen future work.

ISBN: 978-1-61208-572-2

II. BACKGROUND

In this section we start by describing two technological trends that we leverage for our work (namely, virtualization and remote execution of radio functionality, and lightweight virtualization environments) and continue by summarizing recent work very related to our proposal (virtualization of local wireless smart home functionality).

A. *Virtualization and Remote Execution of Radio Functionality*

The Remote Radio Head (RRH) approach used in cellular wireless access networks aims to move wireless baseband processing to the cloud. This has a high cost in terms of bandwidth and it is solved using dedicated high speed lines connecting the RRH with the Base Band Unit (BBU) at the edge of the core network.

We find that the virtualization of higher level functionality is appealing also for residential environments. However, the specificities of local area protocols, inherently different from cellular wireless, make it necessary to assess to what extent and under what circumstances this externalization of functions is feasible.

B. *Lightweight Virtualization Environments*

The orchestration and maintenance of the software running on the gateways in large-scale deployments is a challenging task. There are studies, such as [8], that evaluate the performance of the container-based approach compared to a hypervisor-based virtualization when running on gateway devices. The comparison between traditional heavy VMs vs Docker containers architecture is shown in Figure 1. Docker containers are a type of lightweight VMs that can be easily deployed in inexpensive common single board computers like Raspberry Pi.

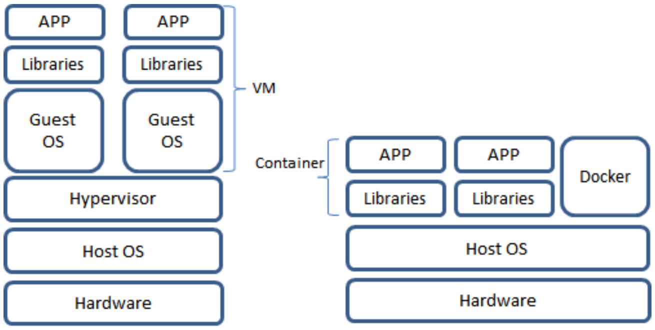

Figure 1. Traditional VM vs Docker architecture.

This Platform-as-a-Service (PaaS) environment is very useful to deploy custom home sensors preprocessing functionality at runtime. In this way, both the necessary bandwidth to the cloud and the round trip delay for those functions are reduced.

The possibility of distributing the resource-intensive functions in intermediate points between the end devices (e.g., sensors and actuators) and the cloud is aligned with the philosophy behind fog computing approaches. This distribution has to be done transparently for the users, and a certain amount of intelligence is needed to manage a fog scenario, preferably in an open and interoperable manner. IoT and 5G are two of the drivers that currently push the activity in fog computing.

C. *Virtualization of Local Wireless Smart Home Functionality*

Building functions that cope with all the diversity that the products in the home present is not feasible, at least currently and at a reasonable cost, for the residential user. However, for an Internet Service Provider (ISP) this would be much easier, especially if these functions are offered as services to its customers and economies of scale can be applied. The ISP should virtualize the actual physical infrastructure of its customers to deliver a set of general and reusable services [5]. In fact, sensing as a service (S2aaS) architectural proposals are specifically concerned with the organizational relationships between the different components and omit details about short range components communications as well as other technical aspects [7].

This is a very active research field. Among the recent works in this area we highlight the following:

In [1], the authors propose an architecture that leverages the increasingly ubiquitous presence of Bluetooth Low Energy radios to connect IoT peripherals to the Internet. The authors propose the use of mobile devices (i.e., Laptops, Smartphones and tablets) as a gateway. The same approach is followed in [8] where the use of smartphones running as a gateway bridges with Bluetooth-enabled devices in a home environment is evaluated.

In [9], a new user-centric management architecture is proposed, to increase the active engagement of residential users in the management tasks of their own networks, improving the usability of the network and facilitating the provision of new services. The proposed architecture combines the SDN and NFV approaches. Additionally, the user-centricity is achieved by implementing interaction and management layers. These layers together constitute a residential network management application. The interaction layer, which can be deployed over different devices, hosts the application that allows the user to configure the network and receive notifications. The interaction layer interacts with the management layer by means of a REST API.

In [2], the authors survey the state of the art on the application of SDN and NFV to IoT. They provide a comprehensive description of the possible implementation aspects for both technologies

In [3], the authors highlight some IoT challenges that the network & IT infrastructure will face. The NFV and SDN benefits are presented from a network operator point of view. The authors present a new multi-layered IoT architecture involving SDN and NFV, and they show how the proposed architecture is able to cope with the identified IoT challenges.

In [4], the authors discuss the usage of NFV technologies and construct a virtual advanced metering infrastructure (AMI) network to transmit energy-related information in a dependable and cost-effective way. The reliability, availability and cost of the new architecture is analyzed and compared to current AMIs.

ISBN: 978-1-61208-572-2

In [10], the authors review the distributed approach to NFV, discuss phased NFV deployment and present critical factors to take such functionalities into account at the customer edge.

III. Proposed Architecture

In our previous work [5], we described our proposal for an architecture that leverages NFV and SDN to offer residential users a portfolio of IoT services. The virtualization of IoT vendor-specific functionality together with the presence of a cost-effective and generic customer premises equipment called HRH would bring economies of scale, easier updates and faster support for new IoT products, among others. This in-home HRH:

- Would not need to implement IoT vendor-specific functions, since these functions would be virtualized and run on a standard NFV infrastructure.
- Would exhibit the in-home raw layer 1 physical flows to the ISP by using tunnels.
- Would be SDN-manageable to establish and maintain the aforementioned tunnels in a standardized manner.

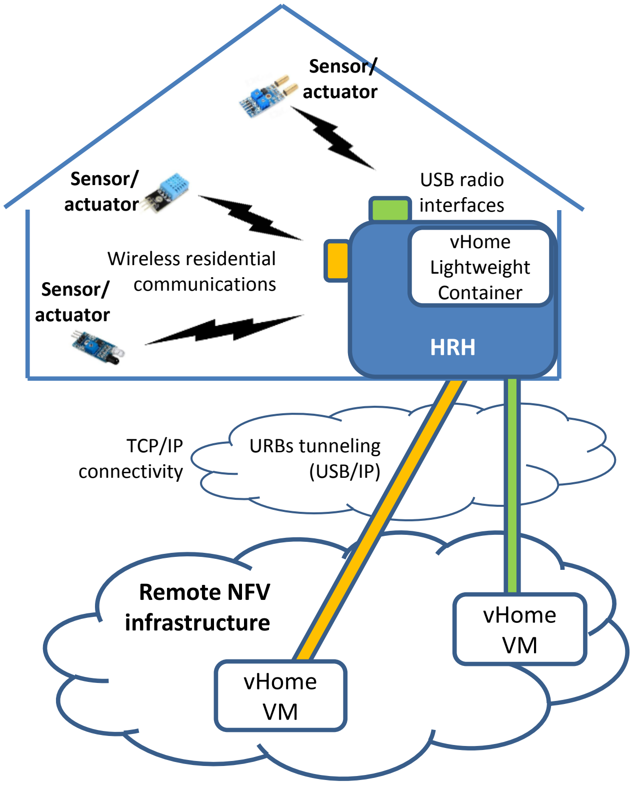

Figure 2. High-level architecture for the remote virtualization of wireless residential communications by means of USB tunneling.

With the work described in this paper we elaborate further on this architecture and decide to establish USB as the vendor-agnostic frontier between the in-home radio flows and the vendor-specific functionality that would be implemented as VNFs. In Figure 2 we show the high-level view of our proposal.

Our architecture is based on the following principles:

- The support of varied short-range wireless interfaces widely used by residential IoT products (e.g., WiFi, Bluetooth or ZigBee). These radio flows are processed locally at low level and exposed to the HRH as standard USB interfaces.
- The establishment of tunnels between the HRH and the virtualization infrastructure at the ISP side, in order to propagate transparently the USB Request Blocks (URBs) that are to be processed by VNFs. The establishment of these tunnels is programmable by using SDN, i.e., the HRH supports Openflow (or any other SDN-compliant southbound interface).
- The vendor-specific functionality, which is realized by means of one or several VNFs (represented as “vHOME” virtual machines in Figure 2), can be distributed if necessary. In this line, the HRH supports a lightweight virtualization environment on which a subset of the VNFs necessary for a service can be downloaded and executed, as commanded by the Virtualized Infrastructure Manager (VIM), see Figure 3.

In the following sub-sections, we elaborate on the implications, advantages and rationale behind each of our main architectural design decisions.

A. *Virtualization of USB Interfaces*

The rationale behind the virtualization of USB interfaces is threefold. Firstly, USB is widely supported by IoT vendors in residential environments. Secondly, the existence of USB dongles at reasonable prices and easily interchangeable is very convenient for the residential market. Lastly, fast prototyping of proof of concepts becomes possible with general purpose equipment (see section IV below).

Moreover, virtualizing the functionality above the USB interface would make various existing IoT products immediately available through our proposed schema. New products also supporting USB would become available to customers almost in a “plug-and-play” manner as long as ISPs supported the adequate virtualized drivers.

Each tunnel would correspond to a new IoT product that utilizes a specific wireless technology (see Figure 2), and the specific drivers and all upper functionality would be placed at the other end of each tunnel, on the ISP side. To implement this idea, we propose to use USB/IP [11], a means of sharing USB devices over a TCP/IP network by encapsulating USB messages between a server (the equipment with the USB device physically connected) and a remote client.

B. *SDN-Programmable Data Path*

In order to provide a flexible configuration, the URB flows must be dynamically provisioned and managed. The HRH would benefit from a generic datapath that is programmable by following the SDN principles. The concrete policies to be applied to the establishment of the tunnels would be implemented and enforced by a SDN controller, and a southbound Openflow-programmable

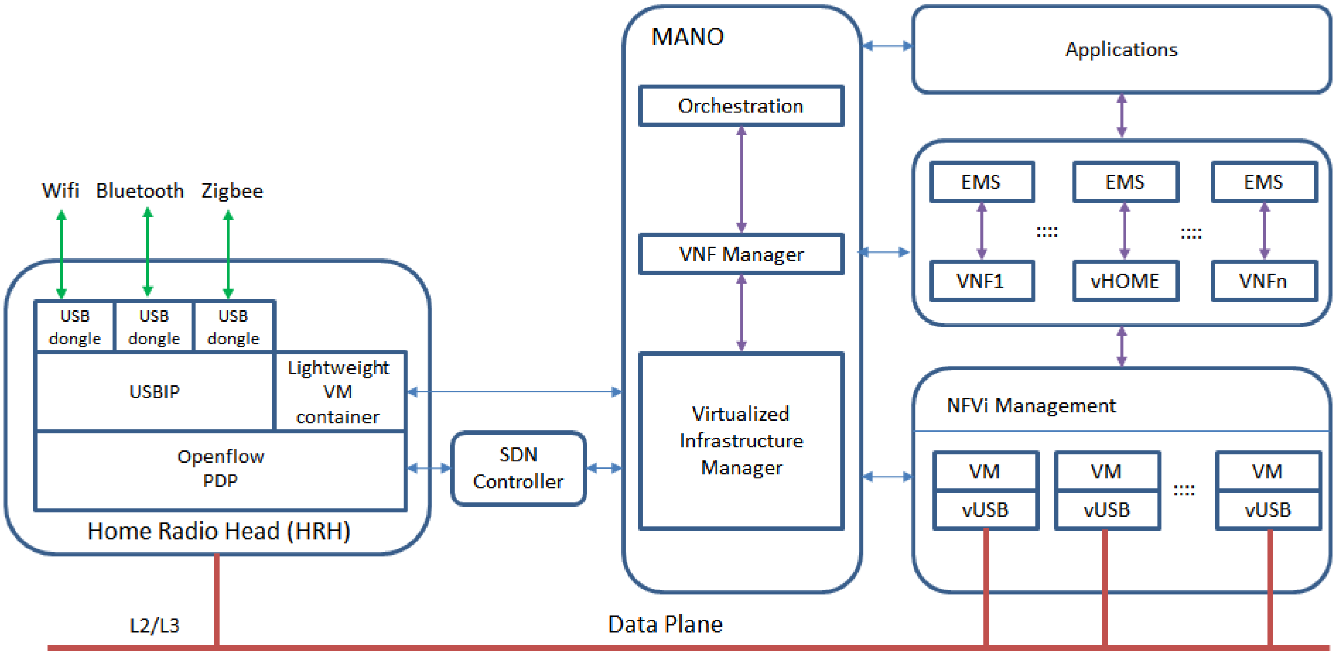

Figure 3. SDN and NFV as considered in our proposal.

datapath has to be supported by the HRH. This way, SDN advantages brought by the software definition of networking configuration are present in our scenario. Also, our HRH would be more easily integrated with a SDN-based residential gateway as proposed in [9].

C. Lightweight Docker Containers

Under certain circumstances it might be convenient or even necessary that a subset of the vendor-specific functionality is carried out inside the customer premises. This might be the case for complying with stringent delay requirements or for saving uplink bandwidth. We propose to provide a light virtualization environment, based either on light virtual machines or on Docker containers, inside the HRH, in which specific modules can be downloaded and executed locally when commanded by the NFV management and orchestration layer.

This distribution of functionality has to be done transparently, without the user being aware of the decomposition of the global service into different modules that may be executed at different points.

D. SDN/NFV Relationship

Our HRH follows the principles of both SDN and NFV. As such, it contains on one hand a generic and programmable networking datapath, and on the other hand a lightweight virtualization environment. The former offers a standard SDN southbound interface so that the SDN controller can provision the USB tunnels dynamically. The latter is formally part of the virtualized infrastructure that has to be managed by the VIM, as per the NFV architecture.

Figure 3 is an enhanced version of a figure we included in our previous paper [5]. We have completed the modules and technologies inside HRH, and we have also made the virtualization capabilities of HRH explicit. The SDN controller that is functionally located between the VIM and the HRH (see Figure 3) can itself be implemented as another VNF, this way leveraging the existing NFV infrastructure.

IV. Experimental Setup and Results

We have implemented the experimental setup shown in Figure 4. To act as HRH, we have equipped a Raspberry Pi 3 with a USB/IP server running on Raspbian OS. This HRH is located inside the Smart Home that the Universidad Politécnica de Madrid has in its South Campus. Both a Bluetooth USB dongle and a USB mass storage device (i.e., a USB pendrive) are connected to the HRH. The ISP side is emulated by means of a Windows PC equipped with a VirtualBox hypervisor. On top of this virtualization infrastructure, a guest OS is run which contains a USB/IP client. This client is in charge of terminating the USB tunnels and offering the virtualized USB dongles as if they were local to the guest OS. On top of these virtualized USB dongles, the concrete specific functionality can be deployed.

The first test we have performed is purely functional. The remote Bluetooth USB dongle is perceived as local at the ISP side. The BT dongle inside the Smart Home receives periodical temperature measurements obtained from a sensor connected to an Arduino board.

These measurements are available at the ISP side thanks to the USB tunneling mechanism. The second test is actually a group of different measurements. The objective of this test is to estimate the bandwidth that would be available through the USB tunneling infrastructure for different local-remote networking scenarios. To perform this estimation, we have run several write and read tests on the regular USB pendrive and have measured the performance of those operations. Four local-remote setups have been considered:

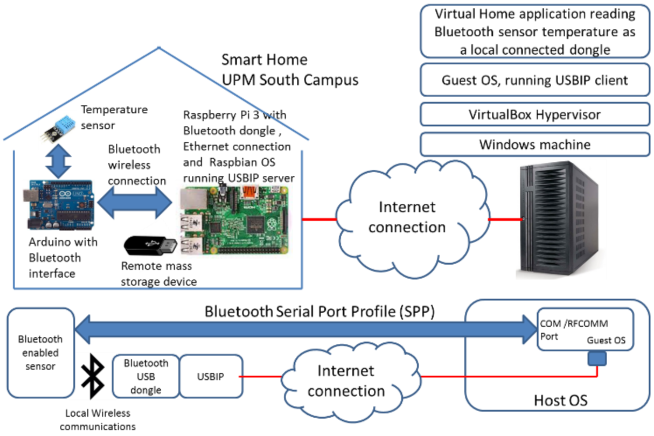
Virtual Home application reading Bluetooth sensor temperature as a local connected dongle
Guest OS, running USBIP client
VirtualBox Hypervisor
Windows machine
Smart Home
UPM South Campus
Temperature sensor
Raspberry Pi 3 with Bluetooth dongle , Ethernet connection and Raspbian OS running USBIP server
Bluetooth wireless connection
Arduino with Bluetooth interface
Remote mass storage device
Internet connection
Bluetooth Serial Port Profile (SPP)
Bluetooth enabled sensor
Bluetooth USB dongle
USBIP
Internet connection
COM /RFCOMM Port
Guest OS
Host OS
Local Wireless communications

Figure 4. Experimental setup.

- Local connection: This is the baseline that gives us the actual write/read capacity of the device. Both operations are performed with a pendrive connected to the same device.
- Same network: The HRH and the emulated ISP are connected to the same Ethernet network. In this case it is a 100BASE-TX wired Ethernet connection.
- Madrid-Alcalá: The HRH is located inside the Smart Home in the South Campus of UPM, whereas the emulated ISP is connected to a residential network in Alcalá de Henares, a village on the outskirts of Madrid. They are around 30 km. apart from each other. The residential network has a 50 MB/s Hybrid fiber-coaxial (HFC) internet connection.
- Madrid-Galicia: The HRH is located inside the Smart Home in the South Campus of UPM, whereas the emulated ISP is connected through a WIFI access point in a public library in Galicia (North of Spain). They are around 500 km. apart from each other. The internet connection of the library, which is provided by a public ISP, is a 100 MB/s HFC access shared by all users.

These setups are chosen to consider scenarios in which not only the distance but also the expected quality of the networking accesses is varied. We have compiled the obtained results in Table I below.

TABLE I. WRITE / READ PERFORMANCE UNDER SEVERAL LOCAL-REMOTE CONFIGURATIONS

Configuration	*Write*	*Read*
Local conection	7.34 Mb/s	17.5 Mb/s
Same network	4.32 Mb/s	6 Mb/s
Madrid-Alcalá	0.9 Mb/s	1 Mb/s
Madrid-Galicia	0.4 Mb/s	0.5 Mb/s

These bandwidth measures should be taken as a starting point. They have been obtained in controlled environments that implement per-user traffic shaping or bandwidth limits policies in order to control peer to peer communications. Nonetheless, even in the most disadvantageous scenario, the bandwidth estimations show that for sporadic or periodical sensor readings (such as temperature or humidity) and for short actuator orders (such as lights on/off control), it is feasible to execute all virtualized functions remotely. Even low-rate video can be remotely tunneled if necessary: as an example a Youtube video with a resolution of 360p consumes around 0.3/0.4 Mb/s. However, in the case of more intensive multimedia traffic, such as a video camera output with higher resolution, it might be necessary to download and execute some of the processing functionality into the HRH, in order to consume less bandwidth towards the ISP. This decision might be made on the basis of bandwidth or delay measurements with each residential subscriber that could be easily carried out by the ISP.

To better assess the reproducibility of these experiments, we highlight here that all the hardware used is inexpensive and off-the-shelf, and all the software is open source. We have also specified the concrete type of Internet access that is present at each setup location to give a better idea of the influence that this might have on the final perceived figures included in Table I. We are aware that different specific locations would have thrown different numbers, but we consider that the objective of demonstrating the feasibility of virtualizing and executing remotely many of the usual residential functions is reasonably well attained.

V. CONCLUSIONS AND FUTURE WORK

This paper describes a basic implementation which shows that the HRH strategy is feasible through the tunneling of USB blocks. The proposed approach, based on common hardware and open software, has several benefits such as its low-cost, low-complexity, easy programmability and alignment with some of the current networking virtualization trends. However, several important issues need further research in order to improve some aspects, for instance the management of uplink communications bandwidth between the HRH and the remote virtualization infrastructure.

As future work we will aim to implement a complete prototype of the proposed architecture and its seamless integration in an existing SDN/NFV infrastructure including its standardized control plane protocols. As regards the distribution of the virtualized functions, and given the specific QoS requirements of video traffic, we consider video preprocessing an ideal candidate to be runtime deployed in the HRH docker container as a VNF. This way, if the overall composed video processing service can be divided into modules with clear interfaces, it could be easily deployed across the NFVi infrastructure, including the HRH.

In our opinion, the demonstration of this scenario would constitute an interesting proof of concept mixing the service chaining capabilities often mentioned for composite virtualized services and the seamless distribution of components, that is one of the basis of fog computing.

ACKNOWLEDGMENT

This work has been supported by Spanish Ministerio de Economía y Competitividad under the grant TEC2015-67834-R.

REFERENCES

[1] T. Zachariah et al., The Internet of Things Has a Gateway Problem, In Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications (HotMobile '15). ACM, New York, NY, USA, 2015, pp. 27-32.

[2] N. Bizanis and F. A. Kuipers, SDN and Virtualization Solutions for the Internet of Things: A Survey, 2016, 4, IEEE Access pp.5591- 5606.

[3] N. Omnes, M. Bouillon, G. Fromentoux, and O. L. Grand, A programmable and virtualized network & IT infrastructure for the internet of things: How can NFV & SDN help for facing the upcoming challenges, *Intelligence in Next Generation Networks (ICIN), 18th International Conference on*, Paris, 2015, pp. 64-69.

[4] M. Niedermeier and H. De Meer. Constructing Dependable Smart Grid Networks using Network Functions Virtualization, Journal of Network and Systems Management 24, 2016, pp. 449-469.

[5] A. B. García, A. Da Silva, L. Bellido, F .J Ruiz, and D. Fernández. Virtualization of Residential IoT Functionality by Using NFV and SDN, In 2017 International Conference on Consumer Electronics (ICCE), Las Vegas, USA, January, pp. 8-10, 2017.

[6] F. Ramalho and A. Neto, Virtualization at the network edge: A performance comparison, IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), Coimbra, 2016, pp. 1-6.

[7] S. Abdelwahab et al., Cloud of Things for Sensing as a Service: Sensing Resource Discovery and Virtualization, In 2015 IEEE Global Communications Conference (GLOBECOM). IEEE, pp. 1–7.

[8] J. Rossey, I. Moerman, P. Demeester, and J. Hoebeke, Wi-Fi helping out Bluetooth smart for an improved home automation user experience, Symposium on Communications and Vehicular Technologies (SCVT), Mons, 2016, pp. 1-6.

[9] R. Flores, D. Fernández, and L. Bellido, A user-centric SDN management architecture for NFV-based residential networks, Computer Standards & Interfaces, Available online 27 January 2017, ISSN 0920-5489 pp. 279-292.

[10] Y. Gittik. Distributed Network Functions Virtualization (White paper),. March 2014.

[11] T. Hirofuchi, E. Kawai, K. Fujikawa, and H. Sunahara. USB/IP - a Peripheral Bus Extension for Device Sharing over IP Network, 2005 USENIX Annual Technical Conference, pp. 47-60.