SDN&NFV contribution to IoT objects virtualization

Article *in* Computer Networks · November 2018

DOI: 10.1016/j.comnet.2018.11.030

8 authors, including:

Luigi Atzori
Università degli studi di Cagliari
270 PUBLICATIONS **19,424** CITATIONS
SEE PROFILE

Raffaele Bolla
Università degli Studi di Genova
248 PUBLICATIONS **3,167** CITATIONS
SEE PROFILE

Giacomo Genovese
Consorzio Nazionale Interuniversitario per le Telecomunicazioni
6 PUBLICATIONS **31** CITATIONS
SEE PROFILE

Antonio Iera
Università della Calabria
358 PUBLICATIONS **21,072** CITATIONS
SEE PROFILE

SDN&NFV contribution to IoT objects virtualization

L. Atzori[*£], J. L. Bellido[§], R. Bolla[*‡], G. Genovese[*#], A. Iera[*#], A. Jara[§], C. Lombardo[*‡], G. Morabito[*¥]
[‡] DITEN - University of Genoa - Genoa, Italy
[*] CNIT, Italy
[#] University of Reggio Calabria, Italy
[¥]University of Catania
[£]University of Cagliari
[§] Hop Ubiquitous S.L. - Murcia, Spain

***Abstract*—The Internet of Things paradigm will certainly be one of the main drivers of the tomorrow's 5th generation (5G) wireless networks. In order to support the algorithms required for real time exchange and analysis of great amounts of data among the involved smart devices, Virtual Objects (VO) have become a key component to improve the objects energy management efficiency, as well as to address heterogeneity and scalability issues. Following the research trend of exploiting on-demand cloud computing resources to augment the processing and storage capabilities of IoT devices, this paper addresses the design of a novel infrastructure and paradigm to support the deployment of new personal IoT services inside the infrastructure provider premises. The goal is to bring cloud-computing services much closer to the end-users and to be able to replace physical IoT devices with their "Virtual Images". Among other benefits, this approach will ensure a longer lifetime to IoT constrained devices and enable the inclusion of new protocols making the development of logic and the configuration of IoT smart application environment technology-agnostic. To achieve this goal, we have developed an open-source software platform that exploits OpenStack APIs and leverages a web interface providing all the functionality typically available in a home gateway through the OpenWRT Linux distribution. Results show this approach brings a manifest reduction in the amount of data transmitted, with benefits in terms of reduced workload and power consumption as well as extended device lifetime.**

***Keywords*—Internet of Things, Network Functions Virtualization, Virtual Object, Mobile Edge Computing.**

I. INTRODUCTION

In recent years, network technologies and architectures are facing a deep revolution in order to meet tomorrow's 5th generation (5G) wireless networks [1] requirements, such as the support for low latency and personalized service delivery. Although 5G is not only conceived for the Internet of Things (IoT), it will certainly be the main driver of the new era of connected objects.

In future IoT applications, devices will likely be requested to perform real time analysis and frequent exchanges of great amounts of data with other involved objects. Implemented algorithms would require too much energy and computational capabilities that constrained physical devices might not support. To overcome the highlighted limitations, Virtual Objects (VO), the digital counterpart of any physical entity, in current IoT platforms have become a key component to support the discovery and mash up of services, to foster the creation of complex applications, to improve the objects energy management efficiency, as well as to address heterogeneity and scalability issues [2].

In this direction, the manifold approaches investigated in the literature exploit on-demand cloud computing resources to augment the processing and storage capabilities of IoT devices. Most of them offer processing and storage capability for time-critical services closer to end-users, to reduce latency in demanding application scenarios. In this direction, Fog computing [3] [4] and Mobile Edge Computing (MEC) [5] paradigms along with the cloudlet concept [6], have recently promoted the idea to rely on a middle tier, between end-devices and the remote cloud, consisting of purpose-built servers, access points (APs) or base stations. This offers fast access to cloud services and the additional benefit of offloading the network/cloud infrastructure, likely unable to sustain the growing demand for mass ICT services. This approach also follows the vision of the Open Fog Consortium [7], whose ultimate objective is to support a ubiquitous Fog computing ecosystem for a wide range of IoT platforms and applications.

The deployment of these paradigms will allow overcoming the main technological issues that may hinder the widespread of IoT services, namely the physical constraints, in terms of computational capacity and energy consumption, of the sensors and actuators. In fact, the current strategy to extend the low processing/storage/communication capabilities of the "things" with virtual counterparts, which act on behalf of the physical objects when involved in a service. Today, the virtual instances of physical objects are executed in specialized application/vendor-dependent servers or central units/controllers, which have to be physically deployed close to physical things (e.g., inside home networks) in order to overcome the poor networking capabilities available on things. These IoT functionalities could greatly benefit from a shift towards the edge of the network, since it would allow maximizing the interworking among the physical objects and their virtual counterparts, along with their accessibility, management, and customization from the users.

With reference to the depicted scenario, the research in this paper addresses the design of a novel infrastructure and paradigm to support the deployment of new personal IoT services inside the infrastructure provider premises. The main goal is replacing physical smart devices usually placed in users' homes (e.g., network-attached storage servers, set-top boxes, etc.), or deployed around for monitoring purposes (e.g., sensors), with their virtual images, providing them to users "as a Service". The chosen approach leverages

both in-network Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) models. Besides, Network Functions Virtualization (NFV) [8] and Software Defined Networking (SDN) [9], widely accepted as key enablers of 5G systems [10], are exploited to support personal IoT services by means of Personal Networks, which are virtual overlay networks associated to each user and employed to interconnect the user to his virtual images with the same level of isolation and security available in the Local Area Network (LAN), independently of the actual user location. An example of this approach can be found in [11].

The Internet of Things is always questioned about its viability to satisfy the security and privacy requirements that are needed in critical and personal scenarios. Thanks to the Personal Network, access to the sensors/actuators for specific functions (enabled management and advanced APIs) is denied to external entities that are not enabled to be visible from the same Layer 2. Thereby, it also ensures the protection and limitation to access from third party Internet devices. The ultimate goal of our research is to bring cloud-computing services much closer to the end-users and to be able to replace physical IoT Devices with their "Virtual Images", thus envisioning the concept of "Smart Device-as-a-Service" (SDaaS), far beyond the classical service model (IaaS, PaaS, and SaaS). More specific objectives are:

- virtualizing typical Network Functions offered to IoT devices by the user's home gateway, by implementing them in the form of software instances that run in commodity computing facilities deployed in the Telco Operator's edge network or in any other edge node infrastructure provider.
- allowing users and Telco operators-service providers to provide and manage cloud services as close as possible to the end users in order to enable datacenter network offloading and offer lower latency reactiveness to offered services.
- enabling an effective, scalable and sustainable fruition of IoT personal Cloud services
- providing a secure and trusted virtual overlay network, capable of handling the interconnection of users' smart devices with standard Layer-2 (L2) protocols and operations equivalent to the ones available today in a user home network, independently of their location (inside/outside the user's home) or their nature (physical/virtual).

It is expected that the deployed solutions will be capable of addressing Internet of Things key challenges in the domains of *Dynamic Resource Allocation* and of *Interoperability among different protocols/technologies*, as well as in the domain of *Security and Privacy*.

As for the first domain, the objective is to address solutions to ensure a longer lifetime to IoT constrained devices by effectively leveraging outsourcing (external allocation) of several functions such as data logging store, algorithms for data processing / aggregation, and reduction of communications. As for the second domain, the specific objective is to enable the inclusion of new protocols in the home management system without (almost) any intervention of the final user, thus making technology-agnostic the development of logic and the configuration of IoT smart application environment (such as for example Home Automation). Last, objectives related to the third domain are to reduce the direct access from the Internet to the physical sensors / actuators, as it happens in Industrial Internet solutions, and to provide mechanisms to carry out Firewall as an access control method to filter and limit the access to authenticated ranges of addresses and well-known domains.

In order to support the deployment of illustrated services in the edge network, we have developed an open-source software platform that will be described in the remainder of this paper. It exploits OpenStack APIs to provide a trusted and well-known communication mean among the involved stakeholders and to allow isolating the physical resources hosting the Fog pico-datacenters while guaranteeing the desired levels of QoS/QoE. On the user side, it leverages a web interface providing all the functionality typically available in a home gateway through the OpenWRT Linux distribution. Last, our solution also offers a networking approach via SDN to provide a Local Network of IoT devices, i.e., a common Layer 2 between the physical IoT objects and the applications, and network functions to limit access to objects from external entities not enabled to be visible from the same Layer 2.

The proposed solution embraces a large number of well-known open-source software projects, offering novel data- or control-plane capabilities, to provide scalable and virtualized networking technologies able to integrate cloud services, both personal and federated in a native fashion. Modularity has been further boosted to cope with the presence of remote network managers interacting with the proposed control plane and to support the migration of the personal service instances across servers in the edge network. The only assumptions required for our solution are the provisioning of containers for processing, persistent storage (virtual images) and enablement of a common L2 layer for the integration of the smart devices and virtual images.

Last point, a testbed has been implemented to carry out a performance evaluation campaign that is illustrated to give the reader a clear idea on the effectiveness of the proposed solutions in achieving the main goals. Tests results show significant reductions in both service latency and network hops, with additional benefits in the amount of data transmitted that decreased by up to 68%, with positive repercussions on battery lifetime.

II. STATE OF THE ART

NFV is a network architecture paradigm that uses IT technologies to virtualize entire classes of network node functions (e.g., router or middle-box functions) into building blocks that may be chained together to create communication services. The NFV technology takes advantage of infrastructure and networking services (IaaS and NaaS) to form the network function virtualization infrastructure (NFVI) [12]. It is the focus of the ETSI NFV industrial study group [13] whose activity is finalized "to enable and

exploit the dynamic construction and management of network function graphs or sets, and their relationships regarding their associated data, control, management, dependencies and other attributes". Interesting guidelines to design and implement NFV in the core segment of telecommunication networks are also provided in [14].

Obviously, the scientific literature has addressed different aspects that highlight how NFV can be an enabler for future IoT platforms. For example, the study in [15] presents some IoT challenges towards the implementation of effective network and IT infrastructures and highlight how the agility brought by the combination of NFV and SDN is essential to face the IoT revolution, by pointing out NFV and SDN related benefits from a network operator point of view. The research in [16] designs an SDN-IoT architecture with NFV implementation with specific choices on where and how to adopt SDN and NFV approaches to address the new challenges of the Internet of Things. A similar objective is pursued in [17], wherein the authors present a general SDN-based IoT framework with NFV implementation for the IoT architecture design. Differently, in [18] an SDN/NFV-enabled edge node for IoT Services by means of orchestration of integrated Cloud/Fog and network resources is proposed. Network connectivity is provided between IoT gateways and deployed virtual machines allocated at the edge node.

Thanks to the offered degrees of freedom, virtualized network functions can be chained to compose network services that provide different trade-offs and functionality to diverse ecosystems, which is the basis of the network slice concept [19]. Of However, such benefits could be nullified in the absence of efficient algorithms to properly address, among other potential issues, service placement [20], scaling [21] and load balancing [22] aspects.

A network slice can be created on demand, for example to allows the isolation of IoT use-cases from other ones [15], based on the available network, compute and storage resources and the requirements of service providers/vertical industries. Also Fog Computing [3] [4], Mobile Edge Computing (MEC) [5], and the Cloudlet [6] concepts have recently been the subject of several investigations and project deployments. The researches in [23] and [24], in addition to the one described in [2], focus on the precise role of Fog and Edge computing in supporting IoT and highlight that these can help: to meet the stringent timing requirements of many IoT systems; to enable hierarchical data processing along the Cloud-to-Things continuum; to reduce IoT device complexity, lifecycle costs, and energy consumption; to ensure non-interrupted services even in the presence of intermittent network connectivity to the Cloud; to face most of the new IoT security challenges [24]. Some interesting insights with regards to the latter issue are also available in [25], while, with reference to IoT resource allocation, it emerges from the literature that the most efficient use of physical resources is accomplished when an application is running on bare metal, followed by when it runs in a container and lastly when it runs in a VM [26]. In distributed virtualized environments for IoT, also application orchestration plays a key role to use of these new distributed resources. In [27] the authors present a solution to an optimization problem encountered in the resource allocation across distributed resources at the edge of the network. In 2012, [28] investigated smart devices and their role into the IoT progression, nowadays NFV and SDN application turned upside down the role of smart device in the IoT world as a unique physical device connected to the network infrastructure.

As for the issue of IoT object virtualization, authors of [29] have highlighted the property of object virtualization techniques to allow quick deployment of new network elements and architectures, while issues such as slicing implementation on a common infrastructure and continuous object reachability have been addressed in [30] and [31], respectively. Besides, Virtualization allows heterogeneous objects interoperability and self-management of network objects through the use of semantic descriptions [32] [2]. Several IoT platforms available from research works and projects leverage object virtualization; among them are: ETSI M2M and oneM2M [33], FIWARE [34] with FogFlow component, the SENSEI project [35], the IoT-A project [36], The COMPOSE [37] architecture, the iCore framework [38].

Depending on the project vision, Virtual Objects (VOs) may differ from each other but they can be described as [39]: "a digital representation, semantically enriched, of a real world object able to acquire, analyze and interpret information about its context, to augment the potentialities of the associated services".

The type of correspondence between the Physical Device (PD) and its virtual counterpart, as well as the kind of established relationship, may strongly depend on the reference service scenario. Therefore, it is possible to associate:

- A single physical object (or device) to one VO, like in [12] [34]; in this case the VO is in charge of receiving and processing all the requests for the PD;
- A single physical object to multiple VOs [36] [37] [40]; the virtual entities associated to the same physical object are dedicated to the provision of different services;
- Multiple elementary physical objects to a single VO [41] able to integrate and process the data of heterogeneous objects, and exposing them through a homogeneous framework for interaction with external services;
- Many physical objects to many virtual objects through a double level of abstraction that involves the use of VO and Composite VO (CVO), like in [42]; the aggregation of multiple VOs aims at satisfying the requirements of applications outside the initial domain of the same VOs.

Regardless the type of correspondence, the virtualization of physical devices can be performed in different points of the network, which are not univocally defined. Nowadays, most solutions prefer to deploy the virtualization layer of such resources within Cloud

platforms, to exploit a greater computational capacity that allows integration with different tools for analytics, data storage, user visualization and data processing. Alternatively, different solutions choose to use hardware equipment already available at the edge of the network, such as Routers / Home Gateways, wherein the virtualization level is deployed by creating a middleware layer between the service(s) that use/process data and physical devices that generate them. The former solution exploits all the advantages of Cloud solutions but suffers from: (*i*) latency; (*ii*) bandwidth consumption; (*iii*) network traffic overload. The latter, however, is one of the first solution in the Edge Computing domain that overcomes the limitations in terms of latency and bandwidth of the cloud platforms although it is greatly affected by limitation in resources available at the hardware devices used, by scalability issues, and by difficulty in handling mobility.

The limits described in the cited solutions assume an increasingly relevant role of network functionalities in supporting IoT ecosystems populated by smart objects connected always and everywhere in mobility. This scenario increases the infrastructural complexity of the systems and poses many problems for the fulfilment of new IoT applications requirements.

In recent times, the evolution of SDN and NFV systems opened to a new placement of object virtualization layer that is still based on the network infrastructure but, differently from Cloud, is deployed in different and not fixed Points of Presence (PoP) of the network ever closer to real objects. As a result, new opportunities will open for vertical industries to deploy their IoT applications by opportunistically exploit the sensing/actuating/computing/networking resources as-a-Service offered by multiple infrastructure domains. This new network configuration will enable VO real-time management and mobility through container migrations and new VO management mechanisms such as clustering. Nowadays, worldwide, Network Providers observe a reduction in their profit and need to find new markets for their business. Consider SDaaS as a native service of its network could open up new opportunities. This latter is thus the approach followed in this paper.

III. NFV TECHNOLOGY AS IOT ENABLER

The cloud-based architecture has satisfied, until now, the needs of most of the IoT applications but it is clear that it will not be able to deal with new real time requirements and the higher and higher volume of data yield by the growing number of connected devices. Besides, it will be further challenged by the increasing of mobility [43] and ubiquitous presence in emerging use-cases such as Smart Cities, connected cars, wearables and home automation.

This work aims at contributing to the evolution of the Internet "brain" beyond current limitations due to obsolete IP network paradigms, which operate most of the time on top of proprietary and specialized firmware/hardware components, require a large number of active packet processing engines across the network, by moving cloud services much closer to end-users and smart-devices. This evolution will be accomplished by introducing intelligence and flexibility ("in-network" programmability) into the edge network point of presence, and by enabling them to host cloud applications (Service_Apps) capable of cooperating with the IoT devices using NFV. ETSI NFV aims to define an architecture and a set of interfaces so that physical network functions, like routers, firewalls, CDNs and telco applications, can be transformed: from software applications designed to run on specific dedicated hardware into decoupled applications – called Virtual Network Functions (VNFs) – deployed on Virtual Machines (VMs) or containers (such as in the Platform as a Service – PaaS- paradigm), on generic servers. In this respect, this work will enforce the following added values of NFV technology in addressing the IoT challenges, with particular attention on the first two aspects:

- **Security and privacy protection for connected personal devices via the brokering with VOs**: this aspect consists in optimizing resources and avoiding attacks such as DDoS due to limitation of resources (constrained devices). The former aspect, offered by the NFV technology architecture, ensures the control of direct access from the Internet to the physical devices, by means of the instantiation of VOs, which are located in high capacity computing facilities at the network edge. An example of these proxy capabilities, as approached in this paper, are presented in the example of SDaaS, which is a VO for managing operations and data on behalf of the IoT device.

- **Capacity extension via virtualization and distributed allocation**: this consists in the allocation of resources next to the sensors/actuators, services for data processing (Data Management), and the reduction of the sensors' complexity and communications with the provisioning of different services such as the Historical Data Cache (HDC) that will be presented in Section IV, which extends the storage capacity of the sensor over an edge service; similarly further resources beyond memory, such as processing capabilities and specific hardware capabilities (e.g. GPUs), can be also offered via the edge nodes.

- **Functionalities extension via virtualization**: NFV enables additional network functions over the edge nodes, which can carry out access control policies to filter and limit the access to authenticated ranges of addresses and well-known domains; some clear examples of these network functions are being focused on Intrusion Detection Systems (IDS) in project such as FORTIKA [44] for IoT networks.

In addition to meeting the mentioned challenges, NFV provides increased network performance values in terms of Latency, Bandwidth usage, service continuity and operational efficiency. NFV enhances the quality of experience (especially in terms of latency) for services, such as Data Visualization as presented in the reference use-case in the next subsection. The main reason for a reduced latency is the availability of VO and HDC, that overcome personal devices limitations (devices work on a duty cycle basis, which introduces latency and limit reliability, and, at the same time, they usually communicate over wireless mediums with reduced bandwidth / speed compared to wired links. Finally, reliability is also optimized since the personal devices are resource-constrained;

this extends their battery lifetimes, due to the load reduction, and in parallel enhances security by avoiding potential DDoS attacks which can make sensors collapse and run out of energy very fast.

Further positive side effects of an NFV based approach are the offered opportunities of performing a more intelligent data management in terms of data aggregation, events detection, behavior modelling, and the opportunity to synchronize data to/from the cloud in off-topic hours.

Figure 1 describes the reference architecture of our research (and the architecture model that we have implemented to validate the described concept of NFV for IoT). This architecture offers the integration of physical devices (e.g., IoT sensors) with edge nodes (base station from telecommunication operators) and cloud services over the public Internet. The figure depicts the instantiation of a personal IoT service for a user (User A in the example). A Platform-as-a-Services (PaaS) instance containing all VOs belonging to User A is hosted in the edge of the network and made accessible to the user through the Personal Network. Additional Back-End Networks, which are isolated L2/L3 broadcast network domains, can be used, if needed for intercommunication among all instances composing a service. The same deployment is replicated for all additional users. Additionally, virtual network functions within the Personal Network provide the firewall and Network Address Translation (NAT) capabilities required for a secure interaction with the public Internet.

Section IV presents the details of how NFV services have been developed to reach and demonstrate the mentioned added values of NFV as an IoT enabler. All the details of the implementation for the SDaaS and HDC, which extend the IoT devices and enhance the performance for user apps such as a Data Visualization App, are described as well.

IV. IoT Services Deployment and Issues

In this section, we describe the service applications, Service_Apps, specifically designed to be integrated with NFV and SDN technologies into our Fog environment and to provide (*i*) virtualization of IoT devices and (*ii*) an IoT platform for data storage/mining (*iii*) and user interfaces [45].

A. *Virtualization of Personal IoT Devices*

We defined our VO solution as a *Smart Devices as a Service* (SDaaS), which hosts digital instances of devices and their related meta-data. From the perspective of this work, device means a physical object with all its equipment of sensors and actuators; we adopt a correspondence of a single VO for each physical device. The SDaaS implements a native IoT service, which enables the presence of VOs at the Edge/Fog Point of Presence (PoP) of the network. The SDaaS has been developed by using Java version Google APP Engine (GAE) [46] [47] to the purpose of directly providing data and metadata without continuously accessing the corresponding physical device, thus prolonging device lifetime and service provisioning.

To support and improve flexibility, we propose to virtualize physical devices by leveraging a labor-saving and “agile” Platform-as-a-Services (PaaS) model, which allows an easy deployment of applications in Cloud computing environments and reduces the VO implementation and deployment burden.

PaaS allows to run web applications on the hosting infrastructure. App Engine applications are easy to build, easy to maintain, and easy to scale according to traffic and data storage needs. Our research envisages to move a PaaS from the Cloud to the Edge of

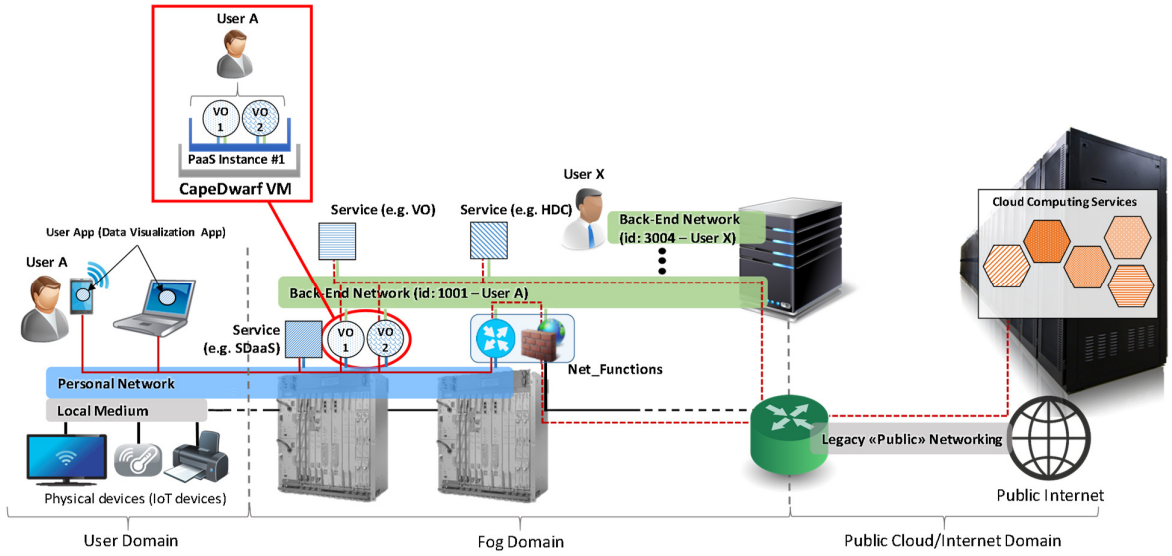


Figure 1. General architecture to deploy and run NFV for a cloud/fog hybrid environment to enable virtualized services (VOs) for physical devices.

the Network. The PaaS adopted in our architecture is Capedwarf, an Application Server that allows deployment of Google App Engine applications in own servers. It proved to be more suitable to our needs than others solution (AppScale) thanks to its flexibility, the possibility it offers to cluster instances, and the large number of interfaces made available. Besides, Capedwarf provides RESTful APIs that rely on the HTTP protocol for management purposes. NFV and SDN Management Control Services use these interfaces to manage PaaS and SDaaS services. According to such interfaces, primitives have been implemented in a Fog environment to perform operations of resource monitoring and migration triggering services.

In the model we propose, the Fog environment that hosts virtual objects is composed of three levels:

- IaaS, implemented through OpenVolcano Virtual Machines (VM)
- PaaS, powered by Capedwarf [48].
- SaaS/SDaaS, deployed by Virtual Objects [28].

The management of instantiation and deployment processes is properly handled through ad-hoc defined primitives that allow to operate on each single VO in the VMs of the Fog platform. Each VM can host several independent PaaS Instances and each PaaS Instance can host several independent VOs (Figure 2). This configuration model is driven by the needs of scalability and flexibility at the Edge of the network. However, it is worth highlighting that the installation of several PaaS instances into the same Virtual Machine can brought to a complex management of IP addressing to provide complete separation and isolation of traffic.

The logical scheme of the developed VO is depicted in Figure 3. It presents a Southbound interface, the Physical Device Interface, which is dedicated to communications with physical devices, and a NorthBound interface, SDaaS APIs, which expose a set of APIs to applications that require services from the VO. The SDaaS core, Virtualization Enhancement Layer (VEL) and OMA LwM2M Hardware Abstraction Layer (OMA-HAL), allows enhanced virtualization, request management, Semantic description and context awareness. In the remainder of this Section, each functional module is better detailed.

Physical Device Interface (PDI) - The task of this module concerns the interaction with the physical counterpart. All the messages from/to the PD are delivered/received through this interface. The IoT world is composed by heterogeneous devices that utilize different communication protocols. Hence, the VO interface has been developed in a modular way so to be adapted to several protocols. Software modules required for the communications through HTTP, MQTT, CoAP, etc protocols are implemented, which interact with the Hardware Abstraction Layer (HAL) in order to manage the available resources.

The HAL Middleware - This software module is implemented when the PD does not support the OMA-LwM2M standard described below. It works as an intermediate level for the translation of resource and interfaces to the LwM2M.

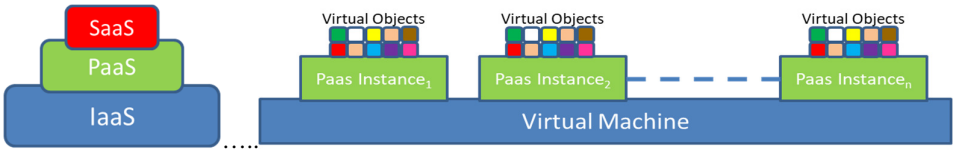


Figure 2. Representation of the hosting of PaaS and VOs within VMs.

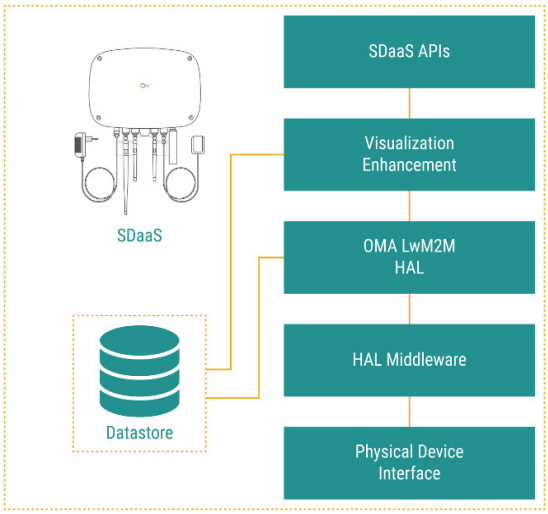


Figure 3. Logical scheme of the developed VO.

The OMA LwM2M HAL - The Open Mobile Alliance (OMA) has developed a Lightweight Machine to Machine (LWM2M) standard. OMA-LwM2M defines interfaces, for Server and Client, to allow the management of IoT devices through the use of an efficient data model, which enables device management and service enabling for M2M devices [49]. Such a model has been integrated in the SDaaS, described in this work, by using the Eclipse Foundation [50] implementation of OMA-LwM2M Server. The LwM2M Object Model purpose is to provide a homogeneous set of resource models for Applications. The mayor contribution to Object Model comes from the IPSO Alliance [51] that defines models for smart city/building applications. Anyway, any developer can create their own objects to be used by their applications or even to be suggested to OMA for standardization [52]. Each PD has a LwM2M Client which is in charge of manage device's resources. Thanks to the LWM2M Enabler, each primary information (value, unit, description) made available by the LWM2M Client is a Resource. The Resources are further logically organized into Objects. The LWM2M Client can have any number of Objects, each of which owns its resources as described in Fig X. Moreover, each device can contain more Objects of the same type and each Resource can be uniquely addressed inside the Client by an URI composed by */{Object ID}/{Object Instance}/{Resource ID}*. This URI is then used by the Application to require specific services to different Objects or Resources through Oma-LwM2M Server APIs.

The LwM2M Enabler interface provides access to resources through the use of "Create", "Read", "Write", "Delete", "Execute", "Write-Attributes", or "Discover" operations. The operations that Resource supports are defined in the Object definition though the Object Template described above. Moreover, the OMA-LwM2M HAL has an *Information Reporting Interface* used to monitor (observe) any change in a Resource, Object Instance, or Object of a Leshan Client.

Virtualization Enhancement - The Virtualization Enhancement layer provides features already present in the IoT architecture but their implementation into the VO enhances the quality of services. The SDaaS has been implemented with additional features that allow to aggregate multiple requests for the same service received by the VO into a single request for the PD. The observer function, for instance, is performed only once on the same resource, if a second application will request the same service, this second request will not be forwarded to the PD but it will be managed by the VO without burdening the physical device. This enhancement is adopted by our SDaaS solution to both save energy of PD and avoid unnecessary traffic crossing the network. Furthermore, the SDaaS has been developed to support and integrate services that can support the Social IoT paradigm [53] adding a social behaviour to our SDaaS.

The SDaaS APIs -The SDaaS APIs are aimed at third-party applications that intend to use the services offered by the VO. While LWM2M is used for device management operations according to OMA standard [49] [54], the APIs provided by the SDaaS have been enriched. For instance, the VO provides APIs to access its datasource, short period storage, querying among the resources, objects and their values by name or key word simply through HTTP request. Furthermore, an additional interface provides the possibility to choose whether to request the last value saved in the SDaaS or to opt for a fresh real-time value of the resource. In the first case, for applications that do not require real-time values, this allows to avoid further access to the PD thus enabling additional savings.

Summing up, the *SDaaS* we propose in this paper may contribute to the improvement of techniques for the virtualization of physical devices by improving the aspects listed below:

1) *Re-usability of code:* The SDaas is *Application independent* because its services are exposed by APIs and can be easily expanded depending on scenarios. The SDaaS is a service that can be deployed inside the network independently from the Applications that will use it. Moreover, it is *Physical Object independent:* the HAL create its device model based on a defined and preload set of LwM2M objects, standard or customizable, dynamically during the OMA LwM2M device registration process. In such a way, every SDaaS can bind to every object and the connection of a new Physical device does not require the writing of a dedicated code, a specific VO, to be bound with.
2) *Scalability:* SDaaS can support from one to multiple physical objects. Can be instantiated individually or grouped with several VO per PaaS instance, depending on the resource management policies. The resources dedicated to VO con be easily monitored and scaled thanks to the PaaS environment.
3) *Flexibility:* SDaaS can be deployed/removed in real-time into every PaaS instances present in the network infrastructure, similarly to container enabling IoT virtual service mobility.
4) *Maintenance:* PaaS provides RESTful APIs that rely on the HTTP protocol that can be used for management of PaaS Instances and SDaaS.
5) *Network resource preservation:* This VO solution has been developed to reduce the traffic from physical device to the edge where VO is located and from the VO to the Cloud. The traffic to the core of the network can be managed in order to take advantage of period of low load. If we consider the huge number of IoT devices that will be connected pushing data to the network, this solution can contribute to reduce the effect caused by billions of IoT small packets that will flood the Network.

B. Data Storage/Mining and User Interfaces: The Historical Data Cache (HDC)

The main aim of this web service is to provide a REST API for data storage and consulting for OMA-LwM2M resource data. Since it has been developed as a web service running into a VM container, it can be used from any device with Internet access and a web browser and can be deployed by using virtualization tools such as Docker. The HDC counts on a database that can be consulted through by the REST API the service provides. It has been developed for integration with the Virtual Object (VO).

Nevertheless, the HDC web service is independent from the VO API and its enriched APIs. Because of that, a new micro-service called Virtual Object Data REST Manager (VODRM) has been developed; it counts on a database to store the relation between each VO and its Physical device, that means that it can work with many VOs at the same time. Since the Virtual Object (VO) is implemented by following the OMA-LwM2M protocol, the implementation of the relationship between both services have been easy and, as a result, we have a scalable standard solution. In the simple example in Figure 4, if we want to perform a GET petition to a OMA-LeM2M resource from a IoT Sensor, the request from the user would be forwarded to the VO by the micro service VODRM acting as a bridge, with the response information provided to the user and stored in the HDC as well.

Below we provide a deeper description of the micro service (VODRM).

We have talked above about its possibilities; for an easier use of the application by the final user in the Personal Network, a web graphic interface has been also developed. As a web application, it can be used from any device via an Internet browser (smart phones, tablets, PCs, …).

The actions that actually can be managed by the VODRM APIs that are consistent with the VO OMA specification[1] and included in the web app are:

- RESOURCE OBSERVATION: The VODRM will receive a POST request carrying the data about which resource the user wants to observe. It will store the submitted observer in its database and will forward the petition to the pertinent VO depending on the device endpoint.
- Instance OBSERVATION: The VODRM will receive a POST request carrying the data about which instance the user wants to observe. It will store/ the submitted observer in its database and will forward the petition to the pertinent VO depending of the device endpoint.
- DELETE OBSERVATION: The VODRM will receive a DELETE request carrying the data about which observer the user wants to delete. It will forward this petition to the pertinent VO depending on the device endpoint. The VO will stop forwarding information about the resource or instance the deleted observation was about.
- Device OBSERVATIONS: The VODRM will receive a GET request with the endpoint of the device the user wants to retrieve the observer's data about. And will reply with the observer's description information.
- write Instance: The user will be able to write in a OMA LwM2M device instance. The VODRM will forward this petition to the VO that will be in charge to notify the OMA device.
- Write Resource: The user will be able to write in a OMA LwM2M device resource. The VODRM will forward this petition to the VO that will be in charge to notify the OMA device.
- Execute Resource: Some object for device managing are able to be executed in order to produce some functionalities. For example, the LwM2M resource that allows to reset physical devices.

Other Functionalities offered by the VODRM for managing the IoT environment:

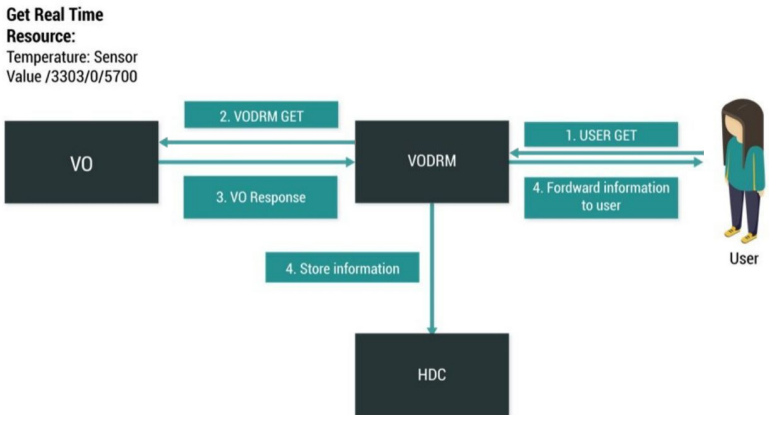


Figure 4. Example of a GET petition to a OMA-LeM2M resource from a IoT Sensor.

- ADD DEVICE: The VODRM service will receive a POST request from the user that will be carrying the endpoint name of the new device. The service will be in charge to manage the available VOs that can be assigned to the new device and sending the binding request the Leshan proxy, the functionality of such proxy will be described in section V. Consequently, this process is fully automatic.
- GET EVENTS BY ENDPOINT: The VODRM can provide data about the events registered from the VO about the physical device status. The user can have a complete report about the device and know when is working or not.
- GET VALUES BY ENDPOINT: The VODRM can provide data about the object, instances and resource values received from the VO about the physical device. The user can have a complete historical data report about the values that his/her sensors are registering. This data is presented with charts in order to provide a full descriptive overview.

V. PERFORMANCE EVALUATION

The use-case chosen to evaluate the performance of our framework consists in the monitoring of different parameters and collection of many data in a Smart Home environment. The user will have all this information available through the deployed services in his/her Network. As shown in Figure 5, the Historical Data Cache Dashboard is the service that offers data and many functionalities to the user: by using this dashboard, she can easily connect new devices to the network, read data from the devices, check historical data stored in the database, etc. The dashboard has also a box that displays the RTT (Round Trip delay Time).

The SDaaS is placed between Application Services and Resources to act like an enhanced physical device with respect to the services (either Service Apps or DC Apps) interacting with it. The VO interacts with a REST API that is shared with other applications, not only with the Historical Data Cache.

This feature allows every application able to perform HTTP requests to use SDaaS services, thus simplifying the way that developers and final users connect their devices to a new application.

The use case design includes two key cloud services that has been deployed into two different Virtual Machine; these services are (Figure 6):

- IoT-VM: PaaS deployment including the SDaaS(s) that is connected with the device(s).
- HDC-VM: The Historical Data Cache deployment that runs the Historical Data Service and the User Interface.

The IoT-VM, depicted in Figure 7, contains the PaaS environment, where SDaaS(s) is deployed and running, and a Proxy. In this solution, the SDaaS implementation relies on the Java application server CapeDwarf, which is a Jboss Wildfly extension. Thanks to the modularity of this PaaS, the most advanced software functionalities are made available to the VO, such as clustering, load balancing, data persistence, servlet container, etc.

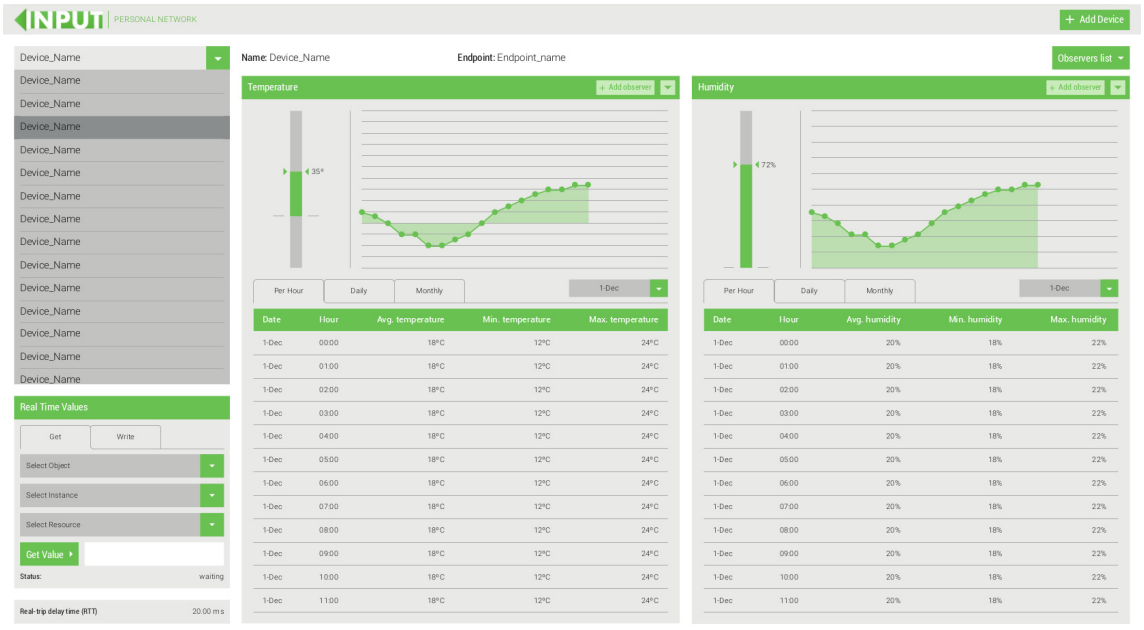


Figure 5. View of the Historical Data Cache dashboard.

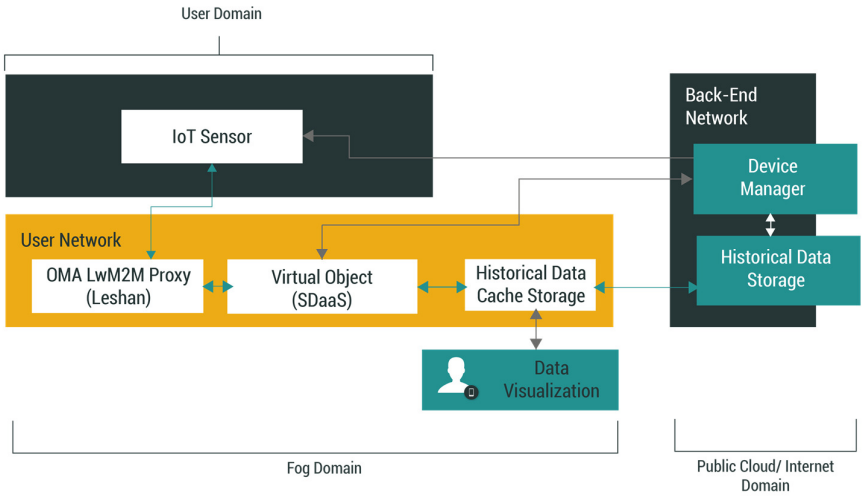


Figure 6. Virtualization scheme of IoT services.

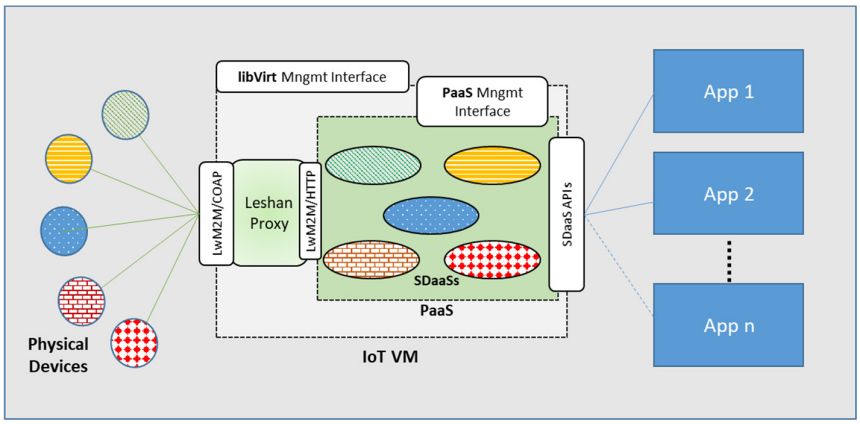


Figure 7. The IoT-VM.

In order to reduce the impact of numbers of VO into the Infrastructure PoP and to optimize the hardware resource usage, several SDaaS instances have been grouped into single PaaS instances. In this specific use-case, for the management of many SDaaSs into the same PaaS instance, and more PaaS instances into the same VM, we use a Proxy middleware based on Leshan server project [55] that we call Leshan Proxy. his Proxy receives the CoAP messages from a physical OMA-LwM2M device and forwards them by HTTP to the related VOs by ad hoc developed rules. Moreover, according to OMA-LwM2M specifications [54], the proxy establishes a secure communication channel with the physical device.

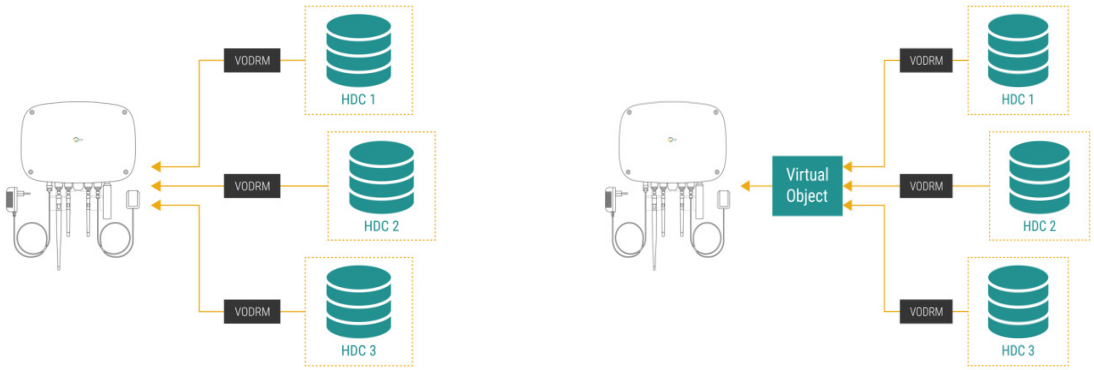
The IoT-VM runs an Ubuntu OS distribution, 16.04.3 LTS Xenial with 1,6 GB of virtual disk and 1GB of RAM allocated in the PoP of network infrastructure. The PaaS istance run a Java Virtual Machine (JVM) with maximum Java Heap Size of 512 MB. The SDaaS is deployed through by a Web application ARrchive(WAR) file of 20,4 MB.

The HDC and the VDRM are both developed in Java language by using Spring framework to implement the REST APIs that supports the functionalities. For this reason, the VMs wherein these services are running needs to be able to execute the Java virtual machine. This VM runs an Ubuntu OS distribution, 16.04.3 LTS too, with 10 GB of virtual disk, in order to have enough space for the data storage, 2 GB of RAM allocated in the PoP of network infrastructure. The VM is running the Java JRE 1.8.0_121-b13 and it executes a MYSQL server too, not accessible from the exterior.

A. *Tested Scenarios*

The main goal of the following test campaigns is to measure:

(*i*) The performance of an IoT device with a keen attention to the amount of data transmitted with and without the presence of VOs: To evaluate this, two test environments have been configured, both present a LwM2M IoT device which is connected to three



(a) (b)

Figure 8. Test configuration in the presence of aLwM2M IoT device connected to three services (a), and using a VO (b).

services, three HDCs each of them belonging to a different user, that are consuming data from it. In first configuration, the device will have to reply directly to each data request from every service, this is illustrated in Figure 8(a). Every time a service requests data the actual workload of the device will thus consist on these four tasks: take a temperature measurement and forward it three time to every service that required the data. Because of this reason, and to measure the VO capabilities for relieving the device workload the second environment consists in the same situation with three Historical Data Caches sharing a same IoT device, but this time a VO is in the middle of the communication acting as a proxy, as can be seen in Figure 8 (b). Thanks to the VO datastore, every time a data request is received it will be cached and forwarded if another request is received in a short period of time according to the rules fixed in the VO. Taking into consideration the characteristics of the two different environments, the test consists in:

The performance of a fixed number of real-time data requests by the three HDCs simultaneously, repeating it in several iterations, increasing the total request number. For every one of those iterations the amount of data transmitted by the physical device is being measured, with the help of OMA object "7", as it is said in the OMA Specification "***This LWM2M Objects enables client to collect statistical information and enables the LWM2M Server to retrieve these information, set the collection duration and reset the statistical parameters***".

With this, we can have an idea of the workload supported by a physical device and how is being relieved with the VO solution. From a theoretical point of view, using a virtual object implies that the amount of data transmitted in each iteration is reduced by 3 times; and this means reduced workload and power consumption as well as extended device lifetime.

(*ii*) The Round Trip Time (RTT) between the elements that form the IoT chain: As it has been said before, the Virtual Object has to work in synchronization with a physical IoT device, so to allow it to extend the physical device functionality and relieve its workload. In relation with the previous test, another excellent benefit is the service latency, because the first value will be provided by the physical device, while the following two will be provided by the VO, which means we are reducing network hops too and related Round Trip Time. In our experiments, the Physical Device is connected to the PoP through a WiFi Access Point.

B. Test Results

1) Phase 1

The plot in Figure 9 represents the amount of data transmitted by the PD when, within a pre-defined valid expiration time, one thousand requests are performed to retrieve the temperature value on resource 3303/0/5700.

As it can be seen, the amount of data transmitted grows and follows a linear progression. The different progressions between the two lines demonstrate that our solution allows data transmission saving of about 110KB with respect to a direct access to the physical device.

This comes from publish-subscribe functionalities offered by the virtualization layer, which is not resource constrained and can adds/adopts different protocol such as those we have, LwM2M which uses Observe method or, for instance, Message Queue Telemetry Transport (MQTT) [56]. The SDaaS can enable device with Pub/Sub capabilities without increasing the PD effort and could help to overcome the limitations for such IoT pattern in Cloud based application as highlighted in [57]. For instance, the SDaaS could implement device ad-hoc policy for message delivery or data discovery, by exposing a defined set of resources to subscribers, depending on the device owner, location or Social behavior to guarantee that messages reach a specific number of subscribers.

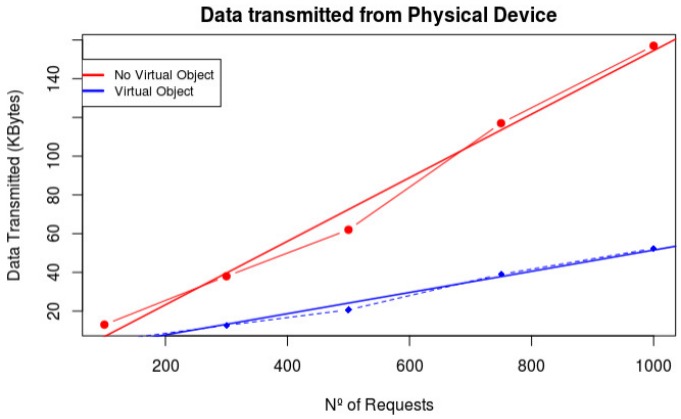


Figure 9. Physical Device Data Transmitted with (blue line) and without (red line) SDaaS.

Regarding those tests, taking a look to the power consumption specs of testing devices WIFI modules, aspects about energy savings can also be taken into consideration. One of the most popular modules is the one that the ESP32 board mounts (By Espressif) and, as the official datasheet [58] says, in a normal configuration it has the power consumption described in Table 1. Taking into account that the VO is avoiding a WIFI reception and a transmission from the PD for each cached value in the VO datastore, almost 300mA are being saved in such situations.

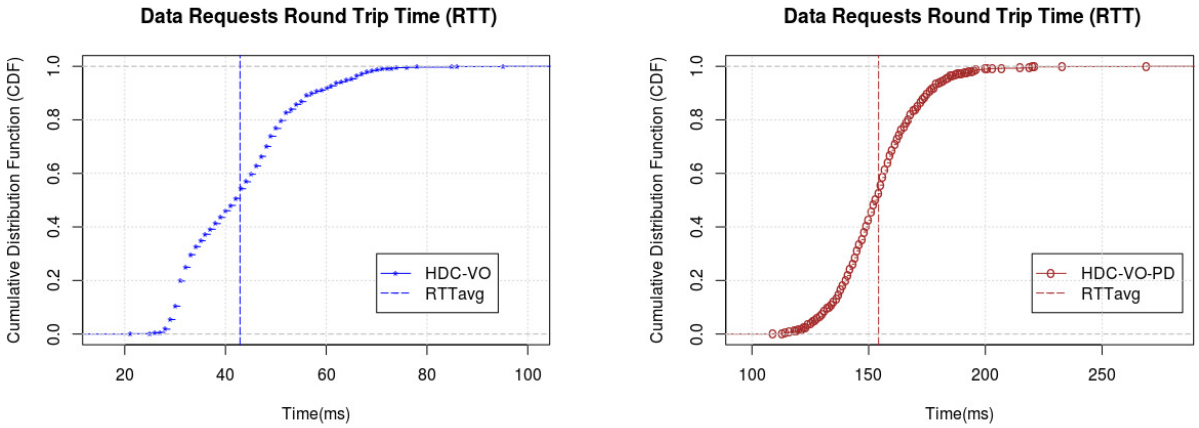
Table 1. WIFI Module Power Consumption

MODE	POWER-CONSUMPTION
TRANSMIT 802.11G, OFDM MCS7, POUT = +14 DBM	180 mA
RECEIVE 802.11B/G/N	95 ~ 100 mA

2) Phase 2

In Figure 10(a) the RTT dataflow test for first phase is presented as a plot, this one is representing the path of a data sample between the PD and the user dashboard running in the HDC VM, the same way in Figure 10(b) the same for the second phase, which only includes the path between the VO and the HDC VMs. To represent this data, the cumulative distribution function [59] has been used to have a statistical idea of the provability regarding requests RTT. In short, this measure refers the provability that a random variable or distribution function X will take values less than or equal that x.

The presence of a service (SDaaS) running in the middle of the path between data consumption services and physical devices has demonstrated to not to add a remarkable extra latency. Moreover, our tests have demonstrated to reduce the RTT when many data



(a) (b)

Figure 10. Cumulative Distribution Function for Data Request Time in NFV environment between Services (a), and from HDC Service to Physical Device (b).

samples are being requested. This is applicable to the first testing phase, where the first value is coming directly from the PD, but the following two are being forwarded immediately from the VO.

VI. CONCLUSIONS

Beyond current limitations in IP network technologies, such as common IP connections with multitude of network hops and typical data processing services running into centralized cloud architectures, in order to evolve the Internet "brain" and to meet tomorrow's 5th generation (5G) wireless networks requirements, network functions virtualization (NFV) and Software Defined Networking (SDN) have emerged. These technologies act as key enablers to support the requirements of future IoT applications, providing efficient data processing, analysis and storage paradigms, reducing the total amount of data which actually needs to be sent to be transported to the cloud.

This work aimed to contribute to such evolution by bringing cloud-computing services much closer to the end-users and to be able to replace physical IoT Devices with their "Virtual Images", envisioning the concept of "Smart Device-as-a-Service" (SDaaS), far beyond the classical service model (IaaS, PaaS, and SaaS). The proposed SDaaS may contribute to the improvement of techniques for the virtualization of physical devices by improving, among other aspects, flexibility, scalability, reusability of the code and reduction of traffic in the network. Moreover, in a NFV infrastructure ready to be deployed in a fog environment where the number of network hops between client and server are drastically reduced. the possibility of data interception by a third entity is also reduced, and the elasticity that offers virtualization platforms such as Open Volcano or OpenStack and NFV deployments in general, allows fast recovery facing DDosS attacks [60], therefore the security aspect is also being potentiated.

Tests results focused on comparing the performance of an IoT device obtained with and without the presence of VOs, and the round trip time within the IoT chain has shown a significant reduction in both the service latency and the network hops, with additional benefits in the amount of data transmitted that decreased by up to 68%. Due to the workload in relieved in physical device, the lifetime of batteries and small hardware is being extended too.

ACKNOWLEDGMENT

This work has been supported by the INPUT (In-Network Programmability for next-generation personal cloUd service supporT) project funded by the European Commission under the Horizon 2020 Programme (Call H2020-ICT-2014-1, Grant no. 644672).

REFERENCES

[1] 5G Vision - The 5G Infrastructure Public Private Partnership: the next generation of communication networks and services", URL: https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf

[2] Nitti, M., Pilloni, V., Colistra, G., & Atzori, L. (2016). The virtual object as a major element of the internet of things: a survey. IEEE Communications Surveys & Tutorials, 18(2), 1228-1240.

[3] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM, 2012, pp. 13–16.

[4] Mouradian, C., Naboulsi, D., Yangui, S., Glitho, R. H., Morrow, M. J., & Polakos, P. A. (2017). A Comprehensive Survey on Fog Computing: State-of-the-art and Research Challenges. IEEE Communications Surveys & Tutorials.

[5] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile Edge Computing - A Key Technology Towards 5G," ETSI White Paper, vol. 11, 2015.

[6] U. Shaukat, E. Ahmed, Z. Anwar, and F. Xia, "Cloudlet Architectures, Applications, and Open Challenges to Deployment in Local Area Wireless Networks," in Journal of Network and Computer Applications. Elsevier, 2015.

[7] https://www.openfogconsortium.org/

[8] M. Chiosi et al., "Network Functions Virtualization: An Introduction, Benefits, Enablers, Challenges and Call For Action," In Proceedings of the SDN and OpenFlow World Congress, Darmstadt, Germany. ETSI White Paper. URL: https://portal.etsi.org/nfv/nfv_white_paper.pdf

[9] Software-Defined Networking: The New Norm for Networks, Open Networking Foundation (ONF)," White Paper, Apr. 2012.

[10] 5G PPP Architecture Working Group, "View on 5G Architecture (Version 2.0)," URL: https://5g-ppp.eu/wp-content/uploads/2017/07/5G-PPP-5G-Architecture-White-Paper-2-Summer-2017_For-Public-Consultation.pdf.

[11] R. Bruschi, F. Davoli, P. Lago, A. Lombardo, C. Lombardo, C. Rametta, G. Schembra. "An SDN/NFV Platform for Personal Cloud Services," IEEE Transaction on Network and Service Management (TNSM), vol. 14, no. 4, Dec. 2017, pp. 1143 - 1156.

[12] ETSI, "Network Function Virtualization: Architectural Framework", http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf, 2013.

[13] ETSI ISG web portal. URL: https://portal.etsi.org/tb.aspx?tbid=789&SubTB=789,795,796,801,800,798,799,797,802.

[14] Hawilo, H., Shami, A., Mirahmadi, M., & Asal, R. (2014). NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC). IEEE Network, 28(6), 18-26.

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

[16] Ojo, Mike, Davide Adami, and Stefano Giordano. "A sdn-iot architecture with nfv implementation." Globecom Workshops (GC Wkshps), 2016 IEEE. IEEE, 2016.

[17] Li, J., Altman, E., & Touati, C. (2015). A general SDN-based IoT framework with NVF implementation. ZTE communications, 13(3), 42-45.

[18] Vilalta, R., Mayoral, A., Pubill, D., Casellas, R., Martínez, R., Serra, J., ... & Muñoz, R. (2016, March). End-to-End SDN orchestration of IoT services using an SDN/NFV-enabled edge node. In Optical Fiber Communication Conference (pp. W2A-42). Optical Society of America.

[19] NGMN Alliance, Description of Network Slicing Concept, January 2016, URL: https://www.ngmn.org/fileadmin/user_upload/160113_Network_Slicing_v1_0.pdf

[20] L. Wang, Z. Lu, X. Wen, R. Knopp, and R. Gupta, "Joint optimization of service function chaining and resource allocation in network function virtualization," IEEE Access, vol. 4, pp. 8084–8094, 2016.

[21] X. Fei, F. Liu, H. Xu, and H. Jin, "Adaptive vnf scaling and flow routing with proactive demand prediction," in IEEE INFOCOM, 2018.

[22] T. Wang, H. Xu, and F. Liu, "Multi-resource load balancing for virtual network functions," in Proc. IEEE ICDCS, 2017.

[23] Yu, W., Liang, F., He, X., Hatcher, W. G., Lu, C., Lin, J., & Yang, X. (2018). A Survey on the Edge Computing for the Internet of Things. IEEE Access, 6, 6900-6919.

[24] CHIANG, Mung; ZHANG, Tao. Fog and IoT: An overview of research opportunities. IEEE Internet of Things Journal, 2016, 3.6: 854-864.

[25] Alrawais, A., Alhothaily, A., Hu, C., & Cheng, X. (2017). Fog computing for the internet of things: Security and privacy issues. IEEE Internet Computing, 21(2), 34-42.

[26] W. Felter, A. Ferreira, R. Rajamony, J. Rubio. "An Updated Performance Comparison of Virtual Machines and Linux Containers" [Online]. Available at: http://domino.research.ibm.com/library/cyberdig.nsf/papers/0929052195DD819C85257D2300681E7B/$File/rc25482.pdf, 21 July 2014.

[27] Hegyi, A., Flinck, H., Ketyko, I., Kuure, P., Nemes, C., & Pinter, L. (2016, September). Application orchestration in mobile edge cloud: placing of iot applications to the edge. In Foundations and Applications of Self* Systems, IEEE International Workshops on (pp. 230-235). IEEE.

[28] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton, "Smart objects as building blocks for the internet of things," IEEE Internet Computing pp. 30–37, January/February 2010.

[29] N. M. K. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges," Communications Magazine, IEEE, vol. 47, no. 7, pp. 20–26, 2009.

[30] N. Niebert, S. Baucke, I. El-Khayat, M. Johnsson, B. Ohlman, H. Abramowicz, K. Wuenstel, H. Woesner, J. Quittek, and L. M. Correia, "The way 4ward to the creation of a future internet," in Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on. IEEE, 2008, pp. 1–5.

[31] N. Chowdhury and R. Boutaba, "A survey of network virtualization," Computer Networks, vol. 54, no. 5, pp. 862–876, 2010.

[32] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," Communications Surveys & Tutorials, IEEE, vol. 16, no. 1, pp. 414–454, 2014.

[33] oneM2M. [Online]. Available: http://www.onem2m.org/

[34] FI-WARE. [Online]. Available: http://www.fi-ware.org/

[35] SENSEI, "Integrating the physical with the digital world of the network of the future," 2008. [Online]. Available: http://www.ict-sensei.org/

[36] IoT-A, "Internet of things - architecture," 2010. [Online]. Available: http://www.iot-a.eu/

[37] COMPOSE, "Collaborative open market to place objects at your service," 2012. [Online]. Available: http://www.compose-project.eu/

[38] iCore, "Empowering iot through cognitive technologies," 2011. [Online]. Available: http://www.iot-icore.eu/

[39] 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. PP, no. 99, pp. 1–1, 2015.

[40] SenaaS: An event-driven sensor virtualization approach for Internet of Things cloud, Sarfraz Alam; Mohammad M. R. Chowdhury ; Josef Noll. Networked Embedded Systems for Enterprise Applications (NESEA), 2010 IEEE International Conference

[41] The SENSEI Project: Integrating the Physical World with the Digital World of the Network of the Future, Mirko Presser, Payam M. Barnaghi, Markus Eurich, Claudia Villalonga. IEEE Communications Magazine,Volume: 47, Issue: 4, April 2009

[42] Giaffreda R. (2013) iCore: A Cognitive Management Framework for the Internet of Things. In: Galis A., Gavras A. (eds) The Future Internet. FIA 2013. Lecture Notes in Computer Science, vol 7858. Springer, Berlin, Heidelberg

[43] Charles C. Byers, Patrick Wetterwald "Fog Computing Distributing Data and Intelligence for Resiliency and Scale Necessary for IoT: The Internet of Things" (Ubiquity symposium), Volume 2015 Issue November, November 2015 Article No. 4.

[44] FORTIKA is an EU funded project under Horizon 2020 (grant agreement No. 740690) focused on Cyber Security Accelerator for trusted SMEs IT Ecosystem. http://fortika-project.eu 2018.

[45] In-Network Programmability for next-generation personal cloUd service supporT. (INPUT), URL: http://www.input-project.eu/

[46] https://cloud.google.com/appengine/docs/java/

[47] ZAHARIEV, Alexander. Google app engine. Helsinki University of Technology, 2009, 1-5.

[48] CapeDwarf open source Google App Engine URL: http://capedwarf.org/.

[49] Klas G., Rodermund F., Shelby Z., Akhouri S., Holler J., "OMA Whitepaper LightweightM2M", 2014

[50] Eclipse Foundation http://www.eclipse.org/

[51] IPSO Alliance http://www.ipso-alliance.org/

[52] OMA Lightweight M2M Resource Model, Joaquin Prado (jprado@omaorg.org) – OMA Technical Director

[53] Atzori, L., Iera, A., Morabito, G., & Nitti, M. (2012). The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. Computer networks, 56(16), 3594-3608.

[54] Lightweight Machine to Machine Technical Specification v.1.0.2- 09 Feb 2018

[55]]https://www.eclipse.org/leshan/ -OMA Lightweight M2M server and client in Java.

[56] A. Banks and R. Gupta, "MQTT Version 3.1.1," Oct. 2014, OASIS Standard.

[57] HAPP, Daniel; WOLISZ, Adam. Limitations of the Pub/Sub pattern for cloud based IoT and their implications. In: Cloudification of the Internet of Things (CIoT). IEEE, 2016. p. 1-6.

[58] Espressif Systems. (August, 2018). ESP32 Series DataSheet:https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[59] Eberly College of science. (August, 2018). Lesson 14: Continuous Random Variables, Cumulative Distribution Functions. Retrieved from: https://onlinecourses.science.psu.edu/stat414/node/98/

[60] Margaret Rouse. (May, 2013). Distributed denial of service (DDoS) attack. Retrieved from https://www.digitalattackmap.com/understanding-ddos/