

Received January 28, 2020, accepted February 13, 2020, date of publication February 17, 2020, date of current version February 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2974504

# NFV and SDN-Based Differentiated Traffic Treatment for Residential Networks

RICARDO FLORES MOYANO<sup>1</sup>, (Member, IEEE), DAVID FERNÁNDEZ<sup>2</sup>, NOEMÍ MERAYO<sup>3</sup>,  
CARLOS M. LENTISCO<sup>2</sup>, AND ANDRÉS CÁRDENAS<sup>2</sup>

<sup>1</sup>Departamento de Ingeniería en Ciencias de la Computación, Universidad San Francisco de Quito, Quito 170901, Ecuador

<sup>2</sup>Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, 28040 Madrid, Spain

<sup>3</sup>Grupo de Comunicaciones Ópticas, Universidad de Valladolid, 47002 Valladolid, Spain

Corresponding author: Ricardo Flores Moyano (rflores@usfq.edu.ec)

This work was supported by the Spanish Ministry of Economy and Competitiveness and the Spanish Ministry of Science and Innovation in the context of GREDOS project under Grant TEC2015-67834-R, ONOFRE-2 under Grant TEC2017-84423-C3-1-P, Go2Edge under Grant RED2018-102585-T, and DISRUPTIVE project under Grant 0677\_DISRUPTIVE\_2\_E.

**ABSTRACT** Residential networks play a critical role in assuring that services or applications such as tele-work, tele-education, medical care, entertainment, home automation, among others, have the required resources to obtain an optimal performance. Although current residential gateways try to meet the Quality of Service (QoS) demands, the traditional networking paradigm does not have the appropriate mechanisms to address the heterogeneous and dynamic nature of the services running at home. In this context, a feasible solution consists of leveraging the flexibility and adaptability of the Software Defined Networking (SDN) and Network Functions Virtualization (NFV) paradigms to provide a differentiated traffic treatment intended to improve the QoS support of residential networks. The proposal takes advantage of the Service Function Chaining (SFC) concept intrinsic to NFV as well as the capacity of an SDN-based residential gateway to differentiate the traffic of a certain application. Thus, an association between an SFC and the differentiated traffic is established to apply a specific treatment. Besides, a comprehensive architecture composed of the software defined residential network (SDRN), the software defined access network (SDOAN) and the NFV-compliant ISP's edge cloud infrastructure is envisioned. This architecture would allow dramatically improving the life cycle management of the residential network from a centralized point which follows a user-centric approach.

**INDEX TERMS** Software defined networks, network functions virtualization, quality of service, service function chaining, residential network management applications.

## I. INTRODUCTION

Nowadays, most of human being activities are performed using different services or applications that require the local and the Internet connectivity service that residential networks provide. For example, tele-work, tele-education, medical care, entertainment, home automation, among others. In order to obtain an optimal performance that guarantees a good level of Quality of Experience (QoE), these high-level applications or services have specific connectivity requirements related to bandwidth, delay, jitter, availability, among others. As it can be noticed, a certain level of Quality of Service (QoS) must be provided by the residential network to meet these heterogeneous demands.

The associate editor coordinating the review of this manuscript and approving it for publication was Ting Wang<sup>1</sup>.

In this context, different Standards Development Organizations (SDOs), like the Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T) and Broadband Forum have proposed residential network architectures for the sake of accomplishing the QoS demands. For instance, a generic home network architecture with support for multimedia services has been proposed by ITU-T [1] while Broadband Forum has proposed a multi-service delivery framework for home networks [2]. Both proposals argue that residential networks should provide local and Internet connectivity service with QoS support to high-level services. Nonetheless, in practice, the residential network is not able to provide QoS support per service or application type in full, due to the following reasons.

On one hand, ITU-T and Broadband Forum architectures have been designed from a technical point of view. That is,

just users with technical and networking background are able to configure the residential network according to the QoS demands. On the other hand, the configuration options or functionalities embedded in the Residential Gateway (RGW) does not provide a comprehensive set of network tools to meet the QoS requirements of high-level services.

Besides, as new functionalities cannot be easily added, the RGW is far from being able to keep pace with the variable and future demands of applications or services.

In order to address these important drawbacks regarding the residential network, in a previous work, we have proposed a new management paradigm following a user-centric approach [3]. This proposal represents the first step towards the re-definition of residential networks considering both dimensions: management and network architecture. With the aim of improving the management of the network, residential users are provided with Residential Network Management (RENEMA) apps. The proposal leverages the Software Defined Networks (SDN) and Network Functions Virtualization (NFV) paradigms to provide a tailor-made, agile and flexible network environment that successfully responds to the users' needs or preferences. Furthermore, considering that residential networks, by definition, provide a connectivity service with QoS support, the differentiated traffic treatment based on the Service Function Chaining (SFC) concept of NFV [4] has been explored as well. This approach allows adding new network functionalities to create a specific SFC intended to improve some connectivity aspect of the residential network such as Security or QoS.

In the present work, the differentiated traffic treatment approach has been further developed to successfully enrich or extend the connectivity service catalog that residential networks offer. This represents an important achievement towards the provision of residential networks capable of successfully supporting advanced multimedia applications or services. The differentiated traffic treatment approach allows fine tuning the connectivity requirements to guarantee a certain level of QoS and, therefore, improving the QoE levels. On the other hand, the provision of agile, smart and flexible residential networks allows the ISPs optimizing the deployment, management and maintenance processes while facilitating the creation of new business models.

To achieve this, an interaction model between the user-centric management architecture and the NFV platform has been defined. This model is intended to orchestrate the SDN and NFV components that reside on the virtualized Management and Networking Domain (vMANDO). In the proposed interaction model, a RENEMA app uses the NFV Management and Orchestration (MANO) interfaces [5] to coordinate the deployment of VNFs and the creation of SFCs. In addition, the app must also coordinate the creation of traffic containers in the SDN-based RGW. These traffic containers are steered to an appropriate SFC which is responsible for applying a specific traffic treatment. The differentiated traffic treatment functionalities of the RENEMA app are included in the corresponding Application Programming Interface (API).

This API has been defined taking into account the principle of associating similar traffic flows into a traffic container as well as the life cycle operations of the different information elements that MANO interfaces provide. As a result, the residential network is able to adapt the connectivity service per traffic container to meet the QoS demands of the high-level services or applications in a flexible and dynamic way. For instance, a WAN accelerator packaged as VNF can be added to the SFC to improve the video streaming service performance.

Certainly, considering the big picture of the residential network, the access network plays an important role towards the provision of a flexible and dynamic ecosystem. With respect to a Software Defined Access Network (SDAN), Kerpez *et al.* [6] indicate that this proposal aims to abstract the entire access network through a unified and common interface for management and control that allows the agile creation of new services adapted to the user needs. In this light, the differentiated traffic treatment proposal can be further improved by leveraging the SDAN programmability.

Considering that the SDAN's control system API allows modifying different characteristics such as the minimum guaranteed bandwidth (upstream and downstream), maximum packet loss rate, jitter, delay, among others; the RENEMA app is in charge of using this API to further improve the QoS support of the residential network.

The remainder of the paper is structured as follows. In Section 2, the state of the art is analyzed and the particular features of the proposed approach are remarked. In Section 3, the differentiated traffic treatment approach to improve the QoS support of the residential network is described. In Section 4, the RENEMA app that interacts with the MANO interfaces and the SDN-based RGW is introduced, in order to implement the differentiated traffic treatment approach. In Section 5, the architectural design of the software defined optical access network is presented, paying attention to the set of functionalities that the API must include to improve the QoS support. To better understand this approach and to validate this proposal, a network scenario composed of the software defined residential network, the SDAN and the NFV-ready edge cloud infrastructure managed by the Internet Service Provider (ISP) is presented in Section 6. Finally, the conclusions of the paper and future works are reported.

## II. STATE OF THE ART

According to a report prepared by OPNFV and OpenDayLight [7], the virtualization of the Customer Premises Equipment (CPE) or RGW has become the most popular use case of NFV and SDN applicability.

The European Telecommunications Standards Institute (ETSI) describes the virtualization of the home environment [8] as the process of replacing the standard RGW by an L2 (layer 2) device. The remaining high-level functions, such as routing or NAT, are provided as Virtualized Network Functions (VNFs) in the NFV infrastructure (NFVI) of the ISP. The analysis of the state-of-the-art shows that the proposals

of Broadband Forum and Central Office Re-architected as a Data Center (CORD) initiative are the most significant examples of the vRGW approach. Thus, these proposals are presented below.

The Network Enhanced Residential Gateway (NERG) architecture [9] proposed by Broadband Forum adopts the vRGW approach. In this architecture, the virtual Gateway (vG) is defined as a logical component that contains network functions which are outsourced from the standard RGW. A Multi-Service Broadband Network Gateway (MS-BNG) or an associated NFVI hosts the different vGs and their corresponding functions, either VNFs or Physical Network Functions (PNFs). NERG also considers that a specific traffic in the vG might need to use a different group of services or chain of service. Broadband Forum has further developed this approach in TR-345 [10] where an NFVI is deployed to increase or replace the services provided by the BNG with VNFs. A classification function located at a physical MS-BNG is responsible for steering the subscriber traffic to a service graph or SFC. In turn, the SFC can be applied to all subscriber traffic or specific flows.

The CORD initiative [11] takes advantage of SDN, NFV and Cloud technologies to transform the Central Office (CO) into a data center. CORD replaces closed and proprietary hardware with software running on commodity servers, switches, and access devices. CORD's residential proposal (R-CORD) includes a Software-Defined Optical Access Network (SDOAN) consisting of an Optical Line Terminal (OLT) enabled for OpenFlow and the software complement designated as virtual OLT (vOLT). R-CORD also virtualizes the RGW or CPE, placing a "bare-metal switch" at the user's home and deploying a virtual Subscriber Gateway (vSG) in the cloud of the central office. The vSG contains a set of network functions selected by the user. In order to support a multitude of virtualized access technologies at the edge of the carrier network (not only PON but also G.Fast or DOCSIS) the CORD initiative has developed SEBA<sup>1</sup> (SDN-enabled Broadband Access).

In consonance with the approach of transforming the legacy central office into a data center, Broadband Forum has proposed the Cloud Central Office (CloudCO) architectural framework [12]. The CloudCO's functionality can be accessed through a Northbound API which follows a Platform-as-a-Service (PaaS) style. This approach allows operators, or third parties, to consume its functionality, while hiding how the functionality is provided from the API consumer. A CloudCO specializes in management, control and orchestration of access and edge functionality. The framework highlights the existence of different CloudCO, non-CloudCO, SDN and non-SDN domains as well as the need of a Service Orchestrator to coordinate the end-to-end service provision.

A clear consensus regarding the NFV-based vRGW approach is observed from the state-of-the-art analysis.

The concepts around this approach have gained an important momentum thanks to the advances performed by different stakeholders, either from the academia or the industry. Nonetheless, some aspects must be considered in order to consolidate this approach. For instance, as Broadband Forum focuses on showing the benefits of NERG and specifying some requirements to achieve them, a detailed description on how to add new network functions to the vG is omitted. With respect to the use of SFCs explained in TR-345, the 5-tuple or application (L7) based traffic steering that the classification function of the MS-BNG performs, does not represent an accurate mechanism that allows identifying specific traffic flows of the residential network. Therefore, the capacity of meeting the QoS requirements of a certain application or service is restricted.

The CORD initiative goes beyond the Broadband Forum proposals and presents a unified platform based on the principles of "Cloud Networking". However, the creation of custom SFCs per traffic flow and per residential network is not considered by R-CORD proposal, which notably restricts the QoS support. Moreover, the kind of device to be placed at the user's home is not defined in full. According to the vSG description, the RGW or CPE is replaced by a "bare-metal switch" while according to the vOLT description, the end-to-end system proposal includes an OpenFlow-controllable CPE. Nonetheless, an application in charge of controlling the CPE is not included in the ONOS controller. The vOLT application is only responsible for controlling the OpenFlow-enabled OLT.

Similar to CORD, CloudCO argues that TR-317 style virtual gateways (vGs) can be deployed inside the NFVI following a "subscriber-as-a-service" model. Regarding SFCs, CloudCO embraces network virtualization overlays. Some network functions run as VMs or containers, while network overlays perform basic L2/L3 steering of packets in-between and to certain network functions. Although CloudCO embraces the NFV paradigm, it is not clear how NERG fits into the CloudCO model. A Network QoS Framework is also proposed for fair and proper sharing of network resources. The framework considers an agreed profile between providers and consumers to guarantee a certain level of QoS. Nonetheless, this QoS level is applied to the main Internet connection. QoS requirements of specific applications or services are not fulfilled. The QoS framework of CloudCO should implement a fine-grained traffic policy definition for the sake of truly providing QoS support in residential networks.

While it is true that NFV leverages the ability of cloud technologies to dynamically provide different kind of services with specific computational requirements, this paradigm focuses on providing a virtualized replica of a custom-designed network device implemented with specific hardware. Therefore, the vRGW approach by using a simple L2 device restricts the steering capabilities required to accurately control traffic flows. For instance, filtering unwanted traffic in the vRGW placed at the ISP facilities is not effective

<sup>1</sup><https://www.opennetworking.org/seba/>

as it can be done at the residential network. Besides, a traffic class belonging to a certain application or service might be allocated a priority before it enters the vRGW. In short, the vRGW approach partially addresses the limitations of the residential network to truly meet the QoS demands of applications or services.

The concept of network slices is gaining a special momentum in order to provide a certain level of QoS per service or application. Ordonez-Lucena *et al.* [13] define network slices as end-to-end (E2E) logical networks that can be created on demand over a common underlying infrastructure. Network slices are mutually isolated, so control and management functions are carried out in an independent manner. Regarding 5G networks, network slices are intended to simultaneously accommodate the wide range of services that vertical-specific use cases will demand. Ordonez-Lucena *et al.* mention that SDN and NFV technologies are key enablers to manage the life cycle of network slices and its constituent network services. Besides, Afolabi *et al.* [14] state that network slicing seeks to assure service customization. Authors describe a network slice as a unification of virtual resources wherein a set of VNFs are instantiated and connected via virtual networks.

Han *et al.* [15] propose creating end-to-end (E2E) network slices to carry specific applications with QoS guarantees. By means of SDN, a physical Access Point (AP) is virtualized to create different virtual APs (vAPs). A vAP represents a network slice. An NFV-based Mobile Edge Cloud (MEC) is used to improve the QoS of the applications running in network slices. Although promising results are reported, all the NFV benefits cannot be properly exploited as the proposed architecture does not follow the NFV architectural framework proposed by ETSI [5]. For instance, an NFV platform such as Open Source MANO (OSM<sup>2</sup>) or ONAP<sup>3</sup> might be used instead of the NFV controller running on top of ONOS<sup>4</sup> SDN controller. Moreover, the bandwidth allocation should be more fine-grained controllable. In the proposal, only the MEC domain's switch is able to limit the access bandwidth of any vAP. A flexible multi-tier traffic control is highly desirable in order to improve the overall QoS support.

On the other hand, using only the SDN paradigm to improve the QoS support might lead to marginal improvements. For instance, Chaabnia and Meddeb [16] propose to slice the software-defined smart home. Network slices created with FlowVisor are allocated to queues of the SDN-based RGW. Queues are intended to guarantee a certain bandwidth, prioritize traffic and to provide an effective bandwidth sharing between slices. Nonetheless, as the NFV paradigm is not considered, SFCs cannot be used to enrich the QoS support already provided by the software defined residential network.

Regarding the access network, as the optical transmission media has become the preferred option, the access

node (i.e., the OLT) must be compliant with the SDN principles. Nowadays, SDN-based OLTs that natively implement OpenFlow are not usually available. As mentioned by Thyagaturu *et al.* [17], to turn legacy OLTs into OpenFlow-controllable devices a hardware abstraction layer must be defined. Different proposals are aligned with this approach. For instance, Clegg *et al.* [18] present an OpenFlow-enabled Gigabit-capable Passive Optical Network (GPON). The hardware abstraction layer presents a single distributed switch to the external OpenFlow controller. The switch includes several ports, one for the OLT and the remaining ports for each Optical Network Terminal (ONT). Similarly, in order to transform the entire GPON system into a virtual switch, Lee *et al.* [19] propose embedding an OpenFlow agent within the OLT. The agent communicates with the external OpenFlow controller and the GPON ONT Management and Control Interface (OMCI) module.

Virtual OLT Hardware Abstraction (VOLTHA<sup>5</sup>) represents the CORD's proposal with respect to next-generation optical access networks based on SDN/NFV technologies. VOLTHA will be analyzed in greater detail in Section V.

### III. DIFFERENTIATED TRAFFIC TREATMENT BASED ON SDN AND NFV

As mentioned before, the residential network should be able to fulfill the QoS requirements of high-level applications or services. In contrast with the proposals of Broadband Forum and CORD, the provision of residential networks based on SDN and NFV paradigms [3], [20], allows a fine-grained definition of traffic classes to which a specific SFC will be applied. As a result, the connectivity service with QoS support that residential networks provide is significantly improved in accordance with user preferences and needs.

#### A. SDN AND NFV ORCHESTRATION

In order to provide a differentiated traffic treatment with QoS support, the user-centric management architecture proposal [3] must orchestrate the SDN and NFV components. To this end, a comprehensive approach is proposed as depicted in Fig. 1 in which two components are distinguished:

##### 1) MANAGEMENT & CONTROL

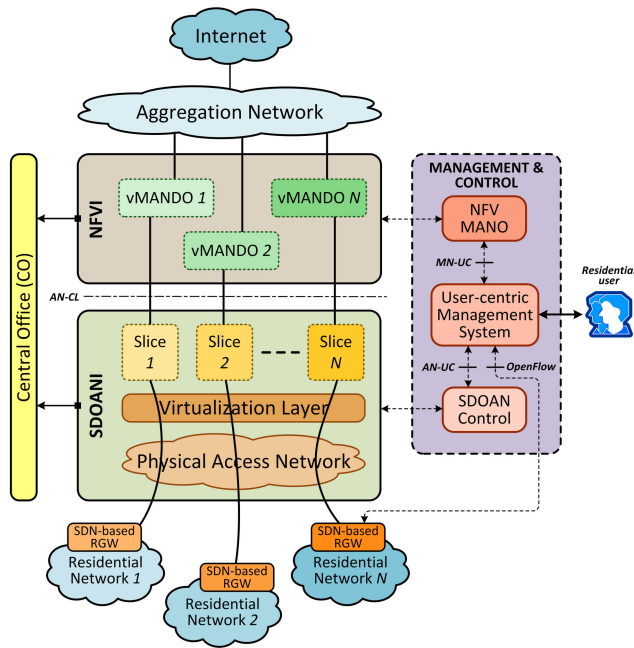
This component is composed of NFV-MANO, the user-centric management system and the SDOAN control. This component defines the functional characteristics of virtual and physical elements that comprise the residential network based on the user's preferences and needs. To achieve this functionality, NFV-MANO and SDOAN control provide different APIs. MANO interfaces allow instantiating VNFs, SFCs and virtual networks required to compose the vMANDO internal structure. The SDOAN control interfaces define the characteristics of the access network such as the minimum guaranteed bandwidth levels in both the upstream

<sup>2</sup><https://osm.etsi.org/>

<sup>3</sup><https://www.onap.org/>

<sup>4</sup><https://onosproject.org/>

<sup>5</sup><https://wiki.opencord.org/display/CORD/VOLTHA>



**FIGURE 1.** Joint orchestration of SDN and NFV. The user-centric management system interacts with SDOAN control and NFV-MANO to coordinate the operation of SDN and NFV based residential network components.

and downstream channel, delay, jitter, maximum packet loss rate, traffic prioritization or classification, among others.

As depicted in Fig. 1, the management system interacts with the SDOAN control through the AN-UC (Access Network – User-centric management system) reference point, and with the NFV-MANO through the MN-UC (MANO – User-centric management system) reference point.

## 2) NETWORK AND COMPUTING RESOURCES

The SDN-based RGW, the SDOAN infrastructure (SDOANI) and the NFV infrastructure (NFVI) comprises the Network and Computing resources to be controlled by the Management & Control component. The SDN-based RGW is used to define specific traffic flows to which a certain set of VNFs will apply a differentiated treatment in the NFVI. The user-centric management system uses OpenFlow to define the requested traffic flows.

Following the concept of NFVI, the access node or Optical Line Terminal (OLT) includes virtualization capabilities in order to provide an SDOAN infrastructure. As depicted in Fig. 1, the virtualization layer provides a customizable virtual access network slice per residential network. After the instantiation of an access network slice, the corresponding functional parameters are managed through the API that SDOAN Control exports in the AN-UC reference point.

Considering a cloud-based management approach, the ISP Cloud infrastructure is in charge of hosting several vMANDOs which are composed of the management system instance and the outsourced virtualized network functions of the residential network. Taking into account a massive instantiation of vMANDOs, the centralized ISP Cloud

infrastructure might not be efficient enough to satisfy the clients’ demands and ensure an adequate QoS level. Moreover, due to the extra processing time introduced by the virtualization layers of NFV (VFNs and SFCs), the latency of applications or services might be significantly increased, and the responsiveness of RENEMA apps might also be constrained.

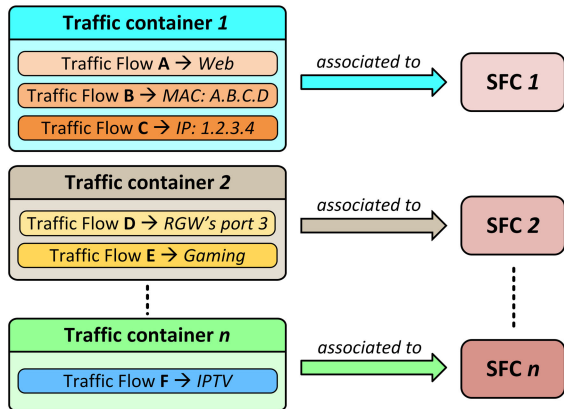
In order to overcome these potential drawbacks, different approaches are encountered in the literature. For instance, Qu et al. [21] propose a scheduling model for VNFs. The model considers both, the latency of the VNFs’ workloads and the latency of the virtual links that connect the SFC. The model improves the scheduling overall performance to accomplish the strict delay requirements of services. Krishnaswamy et al. [22] propose a hierarchy of distributed data centers to flexibly place VNFs in different parts of the hierarchy. The VNF placement is carried out considering different performance requirements such as latency, cost or resource availability. Similarly, the Edge Cloud concept mentioned by Chang et al. [23] or the Edge Computing concept mentioned by Shi and Dustdar [24], represent a novel alternative to centralized data centers located in the ISP’s core network. An Edge Cloud provides shorter response times, reduces the bandwidth consumption of the core network, reduces the overall power consumption and provides an initial stage for traffic processing. CORD and CloudCO are examples of Edge Clouds and can also be considered as examples of hierarchical distribution of data centers.

Following this approach, the ISP Cloud resources can be distributed. To achieve this, a Cloud platform (NFVI) is co-located with the access node (SDOANI) in different Central Offices (COs). Thus, based on the availability of computational resources and the requirements of the latency-sensitive services, each Edge Cloud would be in charge of hosting a certain quantity of vMANDOs. As depicted in Fig. 1, each access network virtual slice is connected to a specific vMANDO through the AN-CL (Access Network – CLOUD) reference point. Considering the physical and logical location of the different SDOANI and NFVI components, the AN-CL reference point might require the implementation of physical and/or virtual networks to achieve the expected functionality.

## B. DIFFERENTIATED TRAFFIC TREATMENT CONCEPT

The differentiated traffic treatment consists in applying a specific SFC to a certain traffic type of an application that runs in the residential network. This approach provides flexibility to apply different traffic processing strategies as well as to adjust a desired QoS. To achieve this functionality, the user is able to specify the applications such as the video on-demand or P2P downloads that require a specific treatment through the definition of *traffic flows*. Moreover, traffic flows with similar characteristics can be grouped through the definition of *traffic containers*.

As an SDN-based RGW is used, multiple ways to define traffic flows can be considered. For instance, using RGW’s



**FIGURE 2. Differentiated traffic treatment. Different traffic flows are defined and grouped in traffic containers. Then, a specific SFC is associated per traffic container in accordance with the traffic processing strategy or the requested level of QoS.**

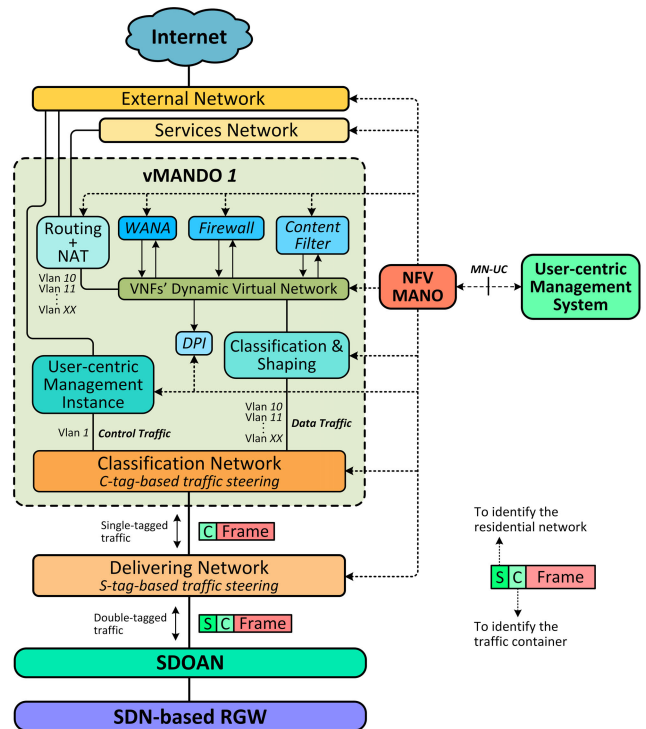
physical ports, IP or MAC addresses or TCP/UDP ports. An example of the definition of different traffic flows identified from A to F are depicted in Fig. 2. Traffic containers identified from 1 to  $n$  are defined to group similar traffic flows. Besides, for each traffic container an SFC is created and also identified from 1 to  $n$ .

An Id is required to identify the traffic container and the corresponding SFC. This Id is also used to identify the virtual channel in charge of conveying and steering the traffic flows from the residential network to the appropriate SFC. Different mechanisms can be used to differentiate traffic containers such as VLANs, VxLAN or tunnels. With the aim of better explaining the proposal, an approach based on VLANs as mentioned by Kim *et al.* [25] is embraced. The residential network traffic is double-tagged using the 802.1ad standard (Q-in-Q). As Fig. 3 shows, the outer *S-tag* is used to identify the residential network while the inner *C-tag* is used to identify the traffic container.

Certainly, the proposed mechanism to differentiate traffic allows creating virtual or logic network slices that run over the top of a common NFVI. In this light, the differentiated traffic treatment proposal is fully aligned with the SDN/NFV based network slicing approach mentioned in the state-of-the-art section. Nevertheless, beyond the creation of a network slice, the present proposal outlines how the network services (collection of SFCs) must be defined and deployed per network slice (a traffic container may be considered as a network slice) over the NFVI to meet the QoS requirements. As mentioned by Afolabi *et al.* [14], the possibility of creating on-demand and customized network slices for the dynamic provision of heterogeneous services, is considered an important feature of 5G networks.

**C. INTERNAL STRUCTURE OF VMANDO**

In order to support the differentiated traffic treatment, vMANDO requires a specific structure composed of several VNFs, virtual networks and SFCs as depicted in Fig. 3. Besides, vMANDO must be connected to *External*,



**FIGURE 3. Internal structure of vMANDO and VLAN-based traffic differentiation mechanism. NFV-MANO instantiates different VNFs to create a specific SFC in accordance with the specifications received from the user-centric management system.**

*Services* and *Delivering* complementary networks to achieve the required functionality. External network provides Internet connectivity and Services network provides connectivity to common services such as DHCP or AAA. Based on the S-tag, the Delivering network steers the traffic of a certain residential network to its corresponding vMANDO. As the traffic arrives at the edge cloud, two levels of classification are required due to the in-band control used. The *Classification* network represents the first level and classifies the control and data traffic. The *Classification & Shaping* VNF represents the second level and binds the traffic container to the corresponding virtual channel.

Two kinds of VNFs are distinguished in vMANDO. Firstly, fixed VNFs are mandatorily deployed per residential network in order to provide a base or default service. *Routing + NAT* and *Classification & Shaping* are fixed VNFs. The former executes the network functions of a traditional RGW while the later employs the rate limiting and traffic classification functions to create the required number of virtual channel terminations. Secondly, following an on-demand model, extra VNFs can be added to create specific SFCs. By means of implementing this approach, the connectivity service with QoS support that the SDOAN and the SDN-based residential network provide can be significantly enriched. For instance, Fig. 4 shows that a WAN Accelerator (WANA) can be added to create the SFC A, which is applied to the Torrent traffic. On the other hand, a Content Filter and a Firewall can be added to create the SFC B, which is applied to the Web traffic.

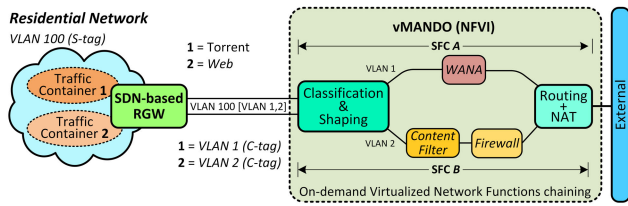


FIGURE 4. Example of differentiated traffic treatment with QoS support. SFCs A and B are associated to traffic containers 1 and 2, respectively.

A. TRAFFIC CONTAINER HANDLER

According to the operational mode describe before, this functional block is in charge of creating different traffic flows. Considering that traffic flows can have similar characteristics, these can be grouped and allocated to a traffic container which must be previously created.

Based on the mechanism used to identify such containers, this functional block coordinates their creation on the SDN-based RGW. Similarly, the virtual channel required to connect the traffic container with its corresponding SFC is also created. After the creation of traffic flows and their respective container, this functional block also controls the remaining CRUD operations. Thus, containers and traffic flows can be updated, deleted or read.

It is important to mention that deploying a new residential network implies providing two default traffic containers. The former being associated to the *internal control traffic* (default traffic flow) while the latter being associated to the *data traffic* (default traffic flow), both following a 1:1 relationship. Once all the residential network components have reached the operational state, the app aids the user in creating new flows and traffic containers as well as in their corresponding allocation. For instance, the app would suggest the user how to improve the Quality of Service (QoS) or Quality of Experience (QoE) by classifying the traffic on the network and the subsequent application of specialized traffic treatment services on the classification performed.

Considering the use of VLANs as identification mechanism and based on the traffic containers already created, the Traffic Container Handler instructs the *Switching* service of the SDN Application layer how to tag the traffic as depicted in Fig. 5. The tagging process can be simple or double and it depends on the ISP’s policies. For instance, the outer *S-tag* can be added by the ONT and the inner *C-tag* can be added by the *Switching* service. Alternatively, the *Switching* service can add both tags.

Note that in response to the creation of a new traffic container, the classification and traffic steering process must be updated. Thus, the VLAN Id that identifies the traffic container is added to the *Classification Network* and a new virtual channel termination is created in the *Classification & Shaping* VNF. The Traffic Container Handler interacts with the NFV platform at the *MN-UC* reference point, using the *NS* and *VNF* APIs that MANO provides as depicted in Fig. 5.

B. SFC HANDLER

This functional block is in charge of coordinating the creation of an SFC. This process requires specifying the traffic container to be processed by the SFC. To this end, a VNF Forwarding Graph Descriptor (VNFFGD) is required. According to ETSI [5], a VNFFGD describes the network service topology by referencing VNFs and PNFs and virtual links (VLs) that connect them. In order to create the VNFFGD, certain functional parameters of both VNF descriptors and traffic containers are required.

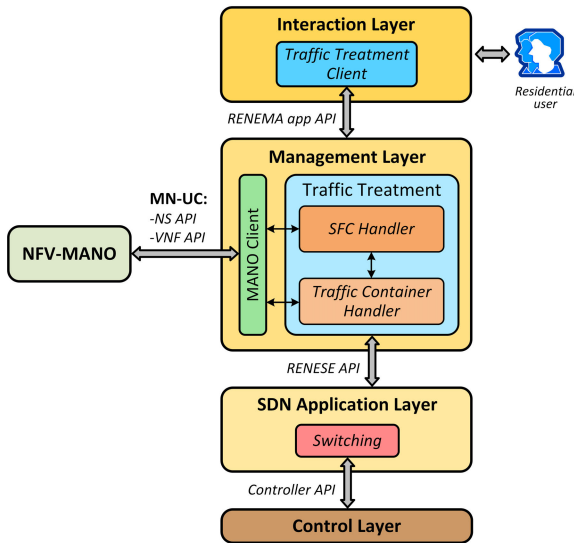


FIGURE 5. Traffic Treatment app. The functional blocks of the application use the APIs that NFV-MANO provides to control the CRUD operations of SFCs. Besides, the RENESE protocol is used to control the CRUD operations of traffic containers and flows on the SDN-based RGW.

With this network configuration proposal, the SDN-based RGW, the SDOAN and the Classification & Shaping VNF lay down a three-tier mechanism to define traffic shaping policies for the virtual channels created between the residential network and the External network. Each level enables the definition of queues for both, upstream and downstream traffic. Furthermore, these policies are intended to improve the QoS level of applications that run on the residential network and services provided by the ISP such as video on demand, IPTV, voice, among others.

IV. RENEMA APP FOR THE DIFFERENTIATED TRAFFIC TREATMENT

In the context of the proposed user-centric management architecture [3], the *Traffic Treatment app* is defined to provide the differentiated traffic treatment functionalities described in the previous section. In general, the app includes two functional blocks as depicted in Fig. 5. On one hand, the *Traffic Container Handler* is in charge of controlling the CRUD (Create, Read, Update and Delete) operations of traffic containers and flows. On the other hand, the *SFC Handler* is in charge of controlling the CRUD operations of SFCs and constituent VNFs.

The VNFs specified in VNFFGD represent new functionalities intended to enhance the connectivity services provided by the residential network. The VNFs to be part of the SFC are chosen through the Traffic Treatment app. Moreover, an association between a traffic container and a specific SFC, as well as to specify a classification policy for the VNFFGD, are required. The classification policy is used by the SFC's service classification function to determine the traffic's path through the VNFs deployed in vMANDO. As the use of VLANs is envisaged, the *C-tag* value that identifies the traffic container must be used as classification policy in VNFFGD. Thereby, the association between the traffic container and the SFC is established.

After the creation of a valid VNFFGD, the SFC Handler incorporates the descriptor to the Network Services catalogue of the NFV platform. Apart from creating instances of SFC, this functional block also controls the rest of CRUD operations. The same as the Traffic Container Handler, the NFV-MANO interfaces are used to provide the described functionalities as depicted in Fig. 5.

Considering that vMANDO concentrates different kind of traffic, a specific SFC for each of them is required. In this light, the following SFCs to be specified in different VNFFGDs are defined:

- *Default SFC for the data traffic:* In the absence of SFCs defined by the user, this SFC steers the data traffic of the residential network to Internet. The corresponding descriptor only includes the fixed VNFs that constitute the SFC.
- *SFC for a specific traffic treatment.* This SFC is created in response of specifying a new treatment schema to be applied to a traffic container.
- *SFC for the internal control traffic.* As an in-band control is being used, an SFC to steer the traffic between the Management System and the SDN-based RGW is defined.
- *SFC for the external control traffic.* Considering that a RENEMA app client must connect to its back-end deployed at vMANDO, an SFC to steer the traffic between Internet and the Management System VNF is required.
- *SFC for the services network traffic.* The ISP includes common network services such as DHCP or DNS. Thus, this SFC steers the traffic of the residential network that needs to reach this kind of common services.

### C. OBJECTS AND API OF THE APPLICATION

The Traffic Treatment app uses objects to represent traffic flows, traffic containers, VNFs and SFCs as depicted in Fig. 6. The app includes several operations to control the objects. These operations are exposed from the Management layer through a REST API, so that the corresponding client can use them. According to Fig. 6, the creation of a certain object depends on the previous creation of another. For instance, in order to create an SFC, a traffic container must

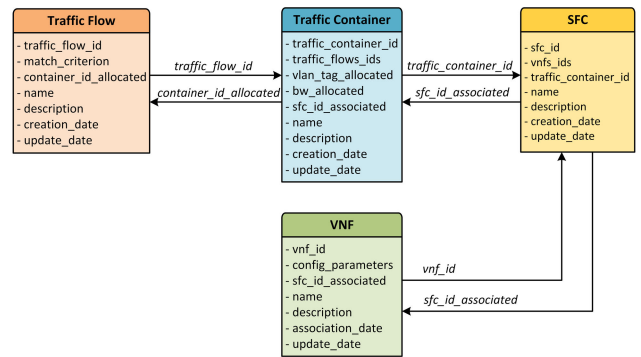


FIGURE 6. Objects defined to provide a differentiated traffic treatment and the relationship between them.

be previously created and the VNF catalogue must have valid descriptors. In turn, to create a container, a traffic flow must be previously created.

As mentioned before, the functional blocks of the app are in charge of controlling the CRUD operations of such objects which are included in the REST API. In general, the API includes five base operations that can be applied to the different objects handled by the app as summarized in Table 1. It is important to mention that the proposed API is based on the definition made by ETSI [5] regarding the operations that MANO provides through NS and VNF interfaces. As these interfaces are intended to control the life cycle of descriptors and subsequent instances of NS, VNF and VNFFG, the Traffic Treatment app uses them to provide the required functionality.

In order to use a certain operation of the API, a comprehension of the implicit functionality is required. Besides, it is important to consider the request parameters as well as to properly interpret the response parameters. The API structure and operation are further explained in Section VI where the proposals are validated in a real testbed.

### D. OPERATIONAL DETAILS

This section shows an example of the interaction between the app, the *Switching service* and *NFV-MANO*. In the NFV environment, vMANDO represents a Network Service (NS) that is instantiated by using the corresponding descriptor. On the assumption that an instance of vMANDO and all its components are properly running, the creation of new SFCs requires the update of vMANDO. To this end, the update operation of NS API is used. The process of applying a Differentiated Traffic Treatment to a specific traffic container can be summarized in five fundamental steps as depicted in Fig. 7.

- In the first step, the traffic flows to be part of a traffic container are created (*Create traffic flow*). For the creation process, the required parameters are provided via the REST API. The number of flows depends on the user needs. For instance, the P2P (BitTorrent) traffic can be allocated with a bounded bandwidth in order to not affect the rest of user traffic. Considering that a



TABLE 1. REST API provided by the Traffic Treatment app.

URI	Method	API
/traffictreatment/flows	GET	List traffic flows: lists traffic flows created
/traffictreatment/flows/<traffic_flow_id>	GET	Show traffic flow: shows information of a specific traffic flow
/traffictreatment/flows	POST	Create traffic flow: creates a traffic flow by providing the match criterion, name and description. The match criterion may be simple or compound, considering, for instance, physical port, device's MAC address, IP address destination or TCP port destination
/traffictreatment/flows/< traffic_flow_id>	PUT	Update traffic flow: updates the values of certain attributes of the traffic flow
/traffictreatment/flows/< traffic_flow_id>	DELETE	Delete traffic flow: deletes a specific traffic flow
/traffictreatment/containers	GET	List traffic containers: lists traffic containers created
/traffictreatment/containers<container_id>	GET	Show traffic container: shows information of a specific traffic container
/traffictreatment/containers	POST	Create traffic container: creates a traffic container by providing the traffic flow(s) to be part of the container, bandwidth to be allocated, name and description
/traffictreatment/containers<container_id>	PUT	Update traffic container: updates the values of certain attributes of the traffic container
/traffictreatment/containers<container_id>	DELETE	Delete traffic container: deletes a specific traffic container
/traffictreatment/sfcs	GET	List SFCs: lists SFCs created
/traffictreatment/sfcs/<sfc_id>	GET	Show SFC: shows information of a specific SFC
/traffictreatment/sfcs	POST	Create SFC: creates an SFC by providing the VNF(s) to be included in the chain, name, description and the traffic container Id the SFC will be applied to
/traffictreatment/sfcs/<sfc_id>	PUT	Update SFC: updates the values of certain attributes of the SFC
/traffictreatment/sfcs/<sfc_id>	DELETE	Delete SFC: deletes a specific SFC
/traffictreatment/vnfs	GET	List VNFs: lists the available VNFs on the NFV platform
/traffictreatment/vnfs/<vnf_id>	GET	Show VNF: shows information of a specific VNF
/traffictreatment/vnfs/<vnf_id>	PUT	Update VNF config: updates the VNF's configuration parameters
/traffictreatment/vnfs/restore/<vnf_id>	GET	Restore VNF config: restores the default VNF's configuration parameters

Traffic Flows, Traffic Containers, SFCs and VNFs are represented as objects as shown in Figure 6. In this light, the REST API is in charge of controlling the corresponding CRUD operations.

default flow for the data traffic exists, an additional flow for the P2P traffic is needed. Regarding the additional flow, the destination TCP port as match criterion for OpenFlow can be used.

- In the second step, the traffic container is created (*Create traffic container*). Certain parameters such as the traffic flow to be associated, the dedicated bandwidth, among others, are specified via the corresponding REST API. The process requires to add the traffic container to the *Switching service*, the *Classification and Delivering networks* and the *Classification & Shaping VNF*. Each entity notifies the app about the result of the operation carried out.
- The third step consists of defining and then creating the SFC (*Create SFC*). To this end, a list of available VNFs and their capabilities is shown. This list is obtained from the catalogue of VNFDs that the NFV platform provides. In order to build the SFC descriptor (VNFFGD), the app uses certain specification parameters of the VNFs previously chosen and the traffic container already created.

The SFC creation process involves deploying the instances of the constituent VNFs. As new VNF instances will be added to the vMANDO instance, an update process of NS must be performed. According to

the NFV Management and Orchestration document proposed by ETSI [5], the update process of a NS requires verifying dependencies with other running VNF instances as well as verifying the availability of NFVI resources supporting the NS instantiation.

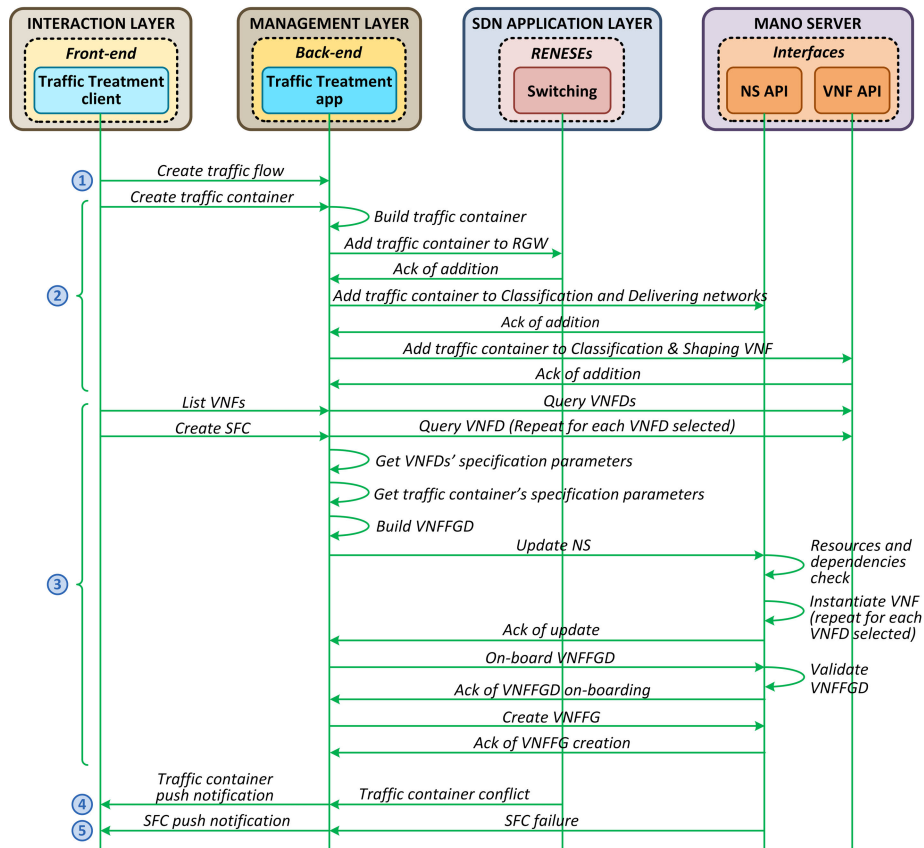
At the end of the instantiation process, the app on-boards the VNFFGD to the NS catalogue for validation. Then, the SFC is created into vMANDO, using the Id of the descriptor previously on-boarded.

- The fourth and fifth steps show the sending of notifications about the state of traffic containers and SFCs (*Traffic container push notification y SFC push notification*). As notifications represent an asynchronous communication started by the back-end, they are sent to the client of the app using the “push” method.

Although the aforementioned explanation is focused on the creation process of traffic flow, traffic container and SFC, the remaining CRUD operations of the different objects follow a similar pattern.

## V. SOFTWARE DEFINED OPTICAL ACCESS NETWORK ARCHITECTURAL DESIGN

As mentioned above, the access network plays an important role in the creation of new programmable residential networks capable of addressing the QoS requirements of



**FIGURE 7. Differentiated traffic treatment implementation process. The app communicates with the Switching service using the Residential Network Service protocol [3]. Moreover, the app uses the services provided by NFV-MANO through the NS and VNF interfaces.**

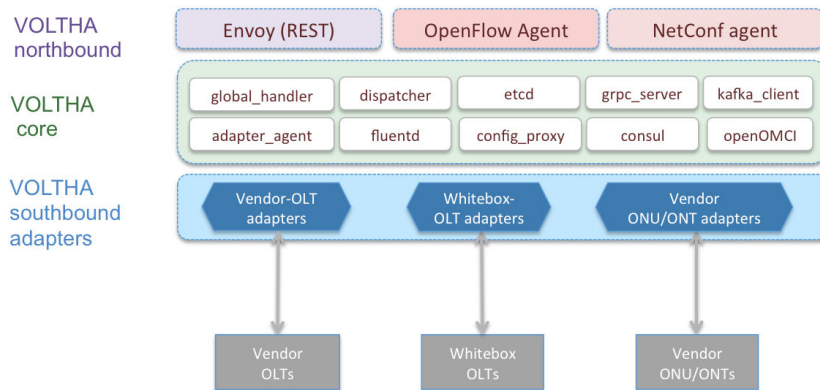
the applications that run at home. The new emerging services, such as real time virtual/augmented reality, online videogames, 4K/8K television, tactile Internet or Internet of Things (IoT) applications, demand highly restrictive network requirements in terms of latency, high bandwidth, packet loss ratio or network resilience. Besides, these applications requirements are typically variable in time. Therefore, new software defined access network architectures are required in order to dynamically and quickly react to fulfil applications requirements, efficiently administering and sharing access network resources among customers.

Optical access networks based on GPON standard are the preferred option nowadays, with new standards and technologies to come that will increase the bandwidth and functionality available, like XGS-PON or NG-PON2. However, most of the currently deployed PON networks do not provide direct and flexible programmable control of service parameters. For example, it is not generally possible to dynamically obtain a temporal increase in bandwidth to use a high demanding video application. Moreover, legacy GPON equipment does not allow a dynamic and fast service configuration since each time a new service is contracted by a residential user, a synchronization time is needed between the OLT and the ONT during which the service connection is interrupted.

In this light, several proposals are being developed in order to integrate access networks into software defined architectures. One of the most mature is VOLTHA, proposed in the context of the CORD project. VOLTHA proposes to decouple the control and data plane functions implemented in nowadays OLTs, virtualizing the control function into a virtual OLT entity (vOLT) and moving it to standard x86 servers hosted in central offices.

VOLTHA follows the *Network as a Switch* paradigm [19], modelling the access network as a virtual OpenFlow switch, having one port for each ONU and one uplink that connects the switch to the NFVI infrastructure where the virtualized network functions of each residential network run (i.e., the virtual residential gateway or vSG defined by CORD). The switch is controlled through a standard OpenFlow interface, which allows SDN controllers to manage the access network as if they were pure SDN networks, creating the flows needed to route the traffic between each residential gateway and the corresponding VNFs.

VOLTHA architecture (Fig. 8) provides isolation between PON low level details, allowing the creation of vendor agnostic management systems. Vendors just need to provide a controller (named Vendor-OLT adapter in Fig. 8) that offers the southbound interface defined in the architecture and



**FIGURE 8. VOLTHA Architecture<sup>7</sup>.** VOLTHA provides three northbound APIs: OpenFlow, for the management of networks flows; and the REST and NETCONF APIs, for the general configuration and management of the network.

translates the standard requests coming from VOLTHA into the vendor-specific management interface. A standard vendor adapter (OpenOLT) designed to be used with white-box OLTs (i.e., OLTs that follow the open specifications published by the Open Compute Project<sup>6</sup>) is also provided by the architecture.

The use of OpenFlow protocol to control the access network as a virtual switch facilitates its integration into a software defined architecture. Events in the PON network can be mapped to existing OpenFlow primitives, for example, when a new user ONT is connected a port status message can be generated to inform the controller. However, as mentioned by Čejka and Krejčí [26], OpenFlow protocol is mainly focused on the flow-based control of a switch and lacks the functionality to configure other more durable parameters, like the ones related to port configuration, the QoS queues, etc. For that purpose, other protocols and APIs should complement OpenFlow to allow the complete configuration of the access network. In this light, there are some initiatives than propose to use available fields in OpenFlow protocol to carry additional information that is used by the vOLT agent to configure the PON services, for example, the METADATA field in flow entries. Other initiatives propose to extend OpenFlow protocol and messages in order to cope with the PON-related parameters as mentioned by Khalili *et al.* [27].

On the other hand, VOLTHA defines two additional northbound APIs to complement OpenFlow: REST and NETCONF (Fig. 8). The NETCONF API is an interface based on the NETCONF configuration protocol defined by IETF in RFC6241. NETCONF provides mechanisms to install, manipulate and delete the configuration of network devices, by means of XML-based messages. This interface is supposed to be used with the YANG data models being defined for Passive Optical Networks by the Broadband Forum [28]. However, at present, it is still under definition and it is not yet implemented in the reference implementation of the CORD

project (OpenCORD). The proposal to use NETCONF for this purpose is aligned with the proposal made by ONF to use the OF-Config protocol, (which is NETCONF based) for the configuration of OpenFlow based switches.

The VOLTHA REST interface is an ad-hoc interface designed to programmatically configure a PON. For that purpose, it includes primitives, for example, to define the type and parameters of the services offered to the customers, in terms of the concepts defined in GPON terminology: GEM ports (flows) and T-CONT (traffic classes) [28]. The REST API is also available as a command line interface. As mentioned, both VOLTHA interfaces (NETCONF and REST) allow the dynamic modification of the service offered to customer networks. Therefore, both could be used by the user-centric management system to dynamically control the QoS in the access network though the SDOAN Control entity depicted in Fig. 1. Besides, the integration of access networks into the architecture will require a deeper study about the correspondence of service chains implemented in the central office and the GEM ports and T-CONT traffic classes defined in GPON.

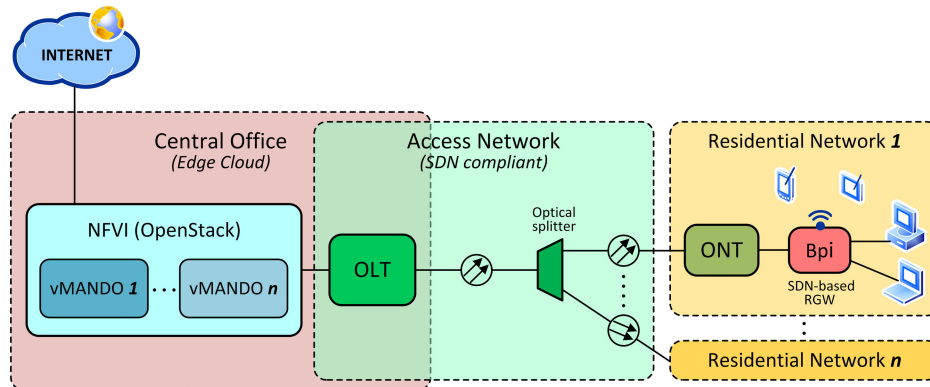
Integration of access network is at present an ongoing task. However, due to the lack of maturity of VOLTHA APIs, as well as the lack of reference implementations, it was decided to define a simple QoS API interface to be able to control the QoS parameters of the validation scenarios based on present GPON network access technology. This initial version of the API and the tests made so far are described in the next section.

## VI. VALIDATION SCENARIO

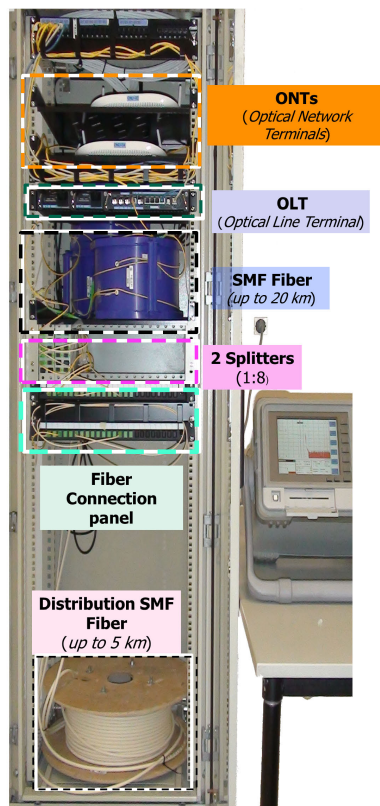
As mentioned in Section III, the provision of a differentiated traffic treatment requires integrating different SDN and NFV components in a single architecture for residential networks (Fig. 1). In this light, the main components of this comprehensive architecture such as the SDN-based RGW, the SDAN and the SFCs to be deployed at the NFVI are included in the global scenario designed to validate the architecture depicted in Fig. 9.

<sup>6</sup><https://www.opencompute.org>

<sup>7</sup><https://wiki.opencord.org/display/CORD/VOLTHA>



**FIGURE 9.** Proposed validation scenario. Note how the residential network, the access network and the NFVI located at the ISP's central office (CO) are integrated. For the sake of simplicity, the management and control components have not been included in the figure.



**FIGURE 10.** Topology of the deployed GPON access network in the experimental testbed.

The GPON access network comprises the OLT “Smart OLT240” and the L2 ONT “WaveAccess 511” from the TELNET-RI<sup>8</sup> vendor as depicted in Fig. 10. The OLT includes four ports, each supporting up to 64 ONTs. As 802.1q and 802.ad standards are supported, *C* and *S* tagging can be performed as previously described in Section III. Two optical splitters 1:8 have been included to enable the configuration of a two-stage splitting topology if desired. The connection between the OLT and the optical splitters

includes three spools of Standard Single Mode Fiber (SSMF) of different lengths (two of 5 Km and one of 10 Km). Besides, the splitters are connected to the ONTs using distribution fibers and connection panels, from 100 meters up to 5 Km.

In the user's premises a Banana Pi<sup>9</sup> (Bpi) is placed to perform the SDN-based RGW functions. The Banana Pi is a router-based development board that includes four Ethernet Gb ports (LAN ports), an Ethernet Gb port (WAN port) and a WLAN interface. Considering that Bpi does not natively support OpenFlow, an Open vSwitch (OVS) is installed, associating OVS ports to the physical LAN and the WLAN ports. It is expected that in the near future both the OLT and the ONT natively support OpenFlow or any standard protocol of the southbound interface of the SDN architecture.

vMANDO has been deployed over an emulated software-defined central office by using Open Source Mano (OSM), which is an orchestration and management framework of network services based on ETSI NFV architecture. OSM is connected to a Virtualized Infrastructure Manager (VIM) emulator named Vim-emu [29], [30], which is responsible for instantiating the VNFs composing vMANDO as docker containers. Initially, Tacker<sup>10</sup> was considered and experimented with as the NFV orchestration framework running over a full OpenStack VIM, but, although Tacker allowed the deployment of one SFCs per Traffic Container, the deployment of several SFCs could not be verified due to the limitation of the management and orchestration capabilities of this NFV platform.

Although several tests have been already made integrating all the main components of the validation scenario using real equipment (i.e., the GPON access network and Bpi based RGW)<sup>11</sup>, the effort has been focused on evaluation the feasibility of implementing the differentiated traffic treatment proposal over available open source NFV platforms, as described in the next subsections.

<sup>9</sup><http://www.banana-pi.org/r2.html>

<sup>10</sup><https://wiki.openstack.org/wiki/Tacker>

<sup>11</sup><http://oa.upm.es/52019/>

<sup>8</sup><https://www.telnet-ri.es/en/>

```

REQUEST PARAMETERS
{
  "traffic_container": {
    "traffic_flows_ids": [{"tf_id": "eafec398ce02"},
      {"tf_id": "ebfcc594ce03"}],
    "bandwidth_allocated": "4",
    "name": "downloads",
    "description": "tf for downloads"
  }
}

RESPONSE PARAMETERS
{
  "traffic_container": {
    "traffic_container_id": "acffc308da54",
    "traffic_flows_ids": [{"tf_id": "eafec398ce02"},
      {"tf_id": "ebfcc594ce03"}],
    "vlan_tag_allocated": "36",
    "bandwidth_allocated": "4",
    "sfc_id_associated": "7767f75100ac",
    "name": "downloads",
    "description": "tf for downloads",
    "creation_date": "14:26:56 30/10/2019",
    "update_date": "14:26:57 30/10/2019"
  }
}

```

**FIGURE 11.** Request and Response parameters of “Create Traffic Container” operation of the REST API. Note that the response parameters match with the parameters described in Fig. 6. In addition, the request parameters match with those described in Table 1.

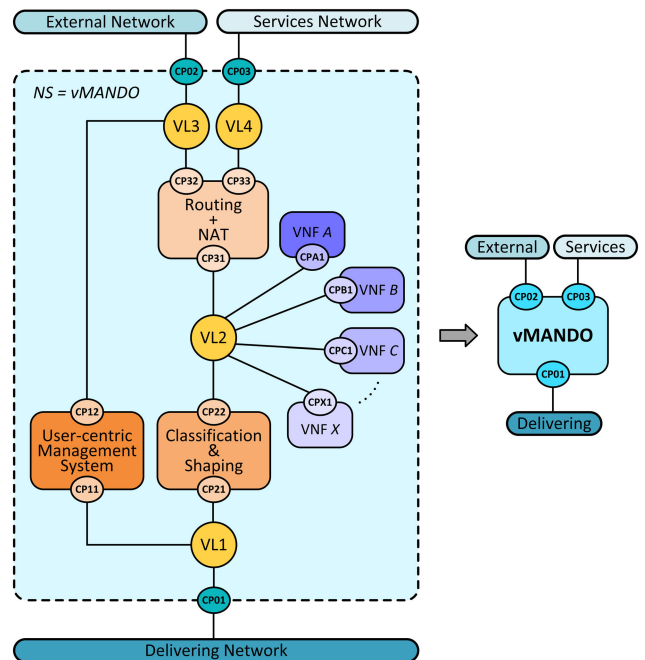
**A. DIFFERENTIATED TRAFFIC TREATMENT APP API**

As explained in Section IV, the traffic treatment app functions are exposed through a REST API, which provides a complete set of operations with their associated parameters. Based on the request parameters received, the app creates extra parameters to include them in the appropriate object, executes the operation and returns the response. Fig. 11 presents an example of the use of this API to create a traffic container, showing the request parameters to be specified. After the execution of the “create” operation, the REST API sends a response that includes the extra parameters that are associated with the object Traffic Container (bottom of the figure). Due to the extension of the CRUD operations that the REST API includes (Table 1), the complete set of request and response parameters have been omitted. Nevertheless, all the CRUD operations of the REST API follow the structure depicted in Fig. 11.

**B. IMPLEMENTING VMANDO AS A NETWORK SERVICE**

vMANDO has been implemented as a network service based on NFV. A network service can be depicted through a Network Connectivity Topology (NCT) graph and one or more VNFFGs. The definition of different VNFFGs makes it possible to apply a differentiated traffic treatment, since different SFCs can be associated to different Traffic Containers. Based on the logical structure of vMANDO depicted in Fig. 3, the resulting NCT is depicted in Fig. 12.

The Classification Network is replaced by the virtual link VL1. The virtual link VL2 interconnects the different



**FIGURE 12.** Network connectivity topology graph (NCT) for vMANDO.

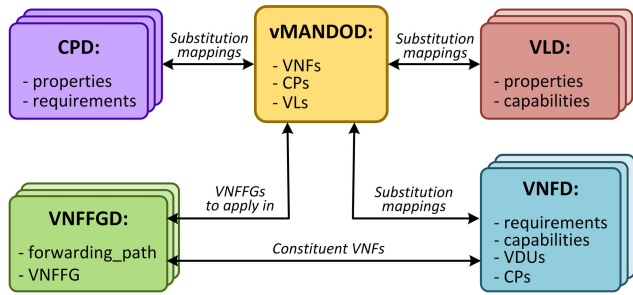
VNFs that are added to vMANDO. The virtual link VL3 interconnects the Management Instance and the VNFs to the External network. The virtual link VL4 provides connectivity to the Services network.

In the context of a network service, the Management Instance can be seen as a VNF that contains the user-centric management system. A VNF is connected to a virtual link through a Connection Point (CP), which represents a virtual and/or physical interface. Each vMANDO can be considered as an aggregated VNF with three endpoints (CP01, CP02 and CP03) that must be connected to three different networks to obtain the required behavior as the right part of Fig. 12 depicts.

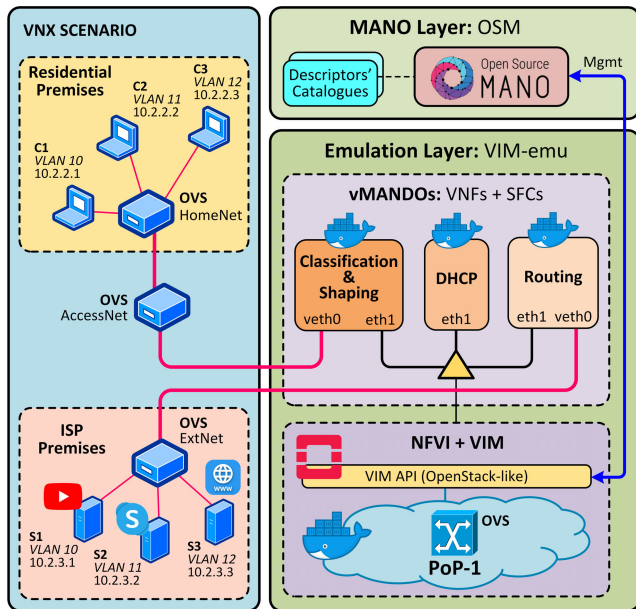
**C. DESCRIPTORS SPECIFICATION**

In order to specify the required descriptors shown in Fig. 13, the TOSCA<sup>12</sup> Simple Profile for NFV [31] and YAML are used. These descriptors must be on-boarded to OSM in order to create instances of NSs and VNFs. These descriptors have been added to a public repository in github [32]. Both vMANDOD and the different VNFDs describe the NCT of Fig. 12. To provide a differentiated traffic treatment, a VNFFG must be specified over the NCT by means of a VNFFGD. On the one hand, based on a classification policy, this descriptor specifies the forwarding path that the traffic takes. Considering the eventual use of VLANs as Traffic Containers identification mechanism, the classification policy might use the VLAN Id. On the other hand, the VNFFG itself is described by listing the descriptors of the constituent VNFs as well as the corresponding CPs and VLs.

<sup>12</sup>Topology and Orchestration Specification for Cloud Applications (TOSCA) defines a language and a meta-model to describe services, their components, relationships and management procedures.



**FIGURE 13.** Relationship between service templates or descriptors used to describe the components of vMANDO.



**FIGURE 14.** Experimental platform used for QoS and performance evaluation. The VNFs are automatically deployed using the MANO interfaces that OSM provides.

Together, vMANDOD, VNFDs, VNFFGDs and VLDs allow applying a specific traffic treatment to a Traffic Container of the residential network as depicted in Fig. 13. Moreover, it is important to point out that the endpoints of vMANDO (CP01, CP02 and CP03 in Fig. 12) can be described in independent service templates. The TOSCA Simple Profile for NFV does not consider this option. However, it would be very useful and can be considered as a functional requirement to optimize the vMANDOD specification. Thus, a service template designated as CPD might be included as depicted in Fig. 13.

#### D. QoS AND PERFORMANCE EVALUATION

An experimental platform based on the validation scenario depicted in Fig. 9 has been developed to evaluate the feasibility of implementing the differentiated traffic treatment proposal over standard NFV platforms and to carry out a QoS and performance evaluation (Fig. 14).

The experimental platform is based on the use of different open source virtualization tools. The first one, named Virtual Network over linux (VNXC) [33], has been used to emulate

both the residential, access and ISP networks. Residential network systems are emulated by VNXC using LXC containers, while the SDN-based RGW has been implemented as an SDN enabled OVS switch. The access and ISP networks are also emulated using OVS switches. Additionally, three servers supporting the three different applications used in the tests (video streaming, video conferencing and a file delivery service based on FTP) are also emulated by LXC containers deployed with VNXC. On the other hand, the management and orchestration platform used is OSM, which acts as both the NFV orchestrator and the VNF manager, controlling the NFVI resources which are emulated using Vim-emu.

As shown in Fig. 14, OSM uses the Vim-emu capabilities to deploy the VNFs that compose the vMANDO network service. Vim-emu deploys these VNFs packaged as Docker containers. In this scenario, a simplified version of vMANDO is considered, which consists of the following VNFs: Classification & Shaping, DHCP, and Routing. For that purpose, three different docker images have been created so that each VNF only includes the specific software components needed to support each virtual network function. The VNFs and network service descriptors have been defined using YAML and they have been on-boarded into OSM to instantiate the vMANDO network service. As it can be seen in Fig. 14, all of the VNFs composing vMANDO are internally connected to an OVS switch provided by Vim-emu and hosted inside one of the docker containers that implement the Vim-emu functionality.

Regarding the differentiated traffic treatment proposal, the SDN-based RGW and the routing VNF have been programmed to tag and untag the traffic with the corresponding 802.1Q tags of each application, with the aim of implementing QoS policies over the access network. A Hierarchical Token Bucket (HTB) discipline implemented in the RGW and in the Classification & Shaping VNF is in charge of implementing these QoS policies using the Linux kernel Traffic Control (TC) functionality. The Routing VNF is in charge of connecting the residential LAN to the outside world, in this case represented by the three application servers.

Fig. 15(a) shows the throughput (Mbps) demanded by each test application when the Classification & Shaping VNF is not active in the vMANDO network service. Initially, the throughput demanded by application App\_A (FTP) is around 18 Mbps. Later, at second 30, Application App\_B (video conferencing) is initiated, so the available bandwidth needs to be shared between the two applications. As a consequence, the quality perceived by the video conference users would be degraded and the time to download the file using FTP would be increased. Finally, at second 70, a third application App\_C (video streaming) starts to request a multimedia content, consuming up to 14 Mbps. As expected, a bandwidth sharing mechanism is required in order to guarantee a minimum QoS for each application.

Fig. 15(b) shows how the activation of the Classification & Shaping VNF in the vMANDO network service provides the required differentiated traffic treatment functionality.

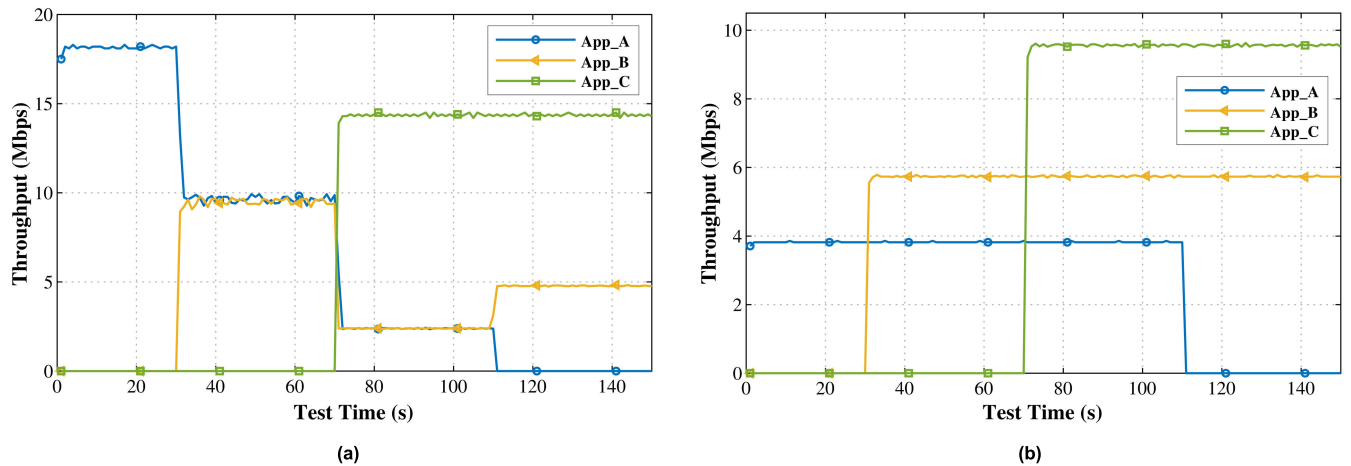


FIGURE 15. Behavior of the applications' throughput that run at the residential network. (a) Differentiated traffic treatment is not active. (b) Differentiated traffic treatment is active.

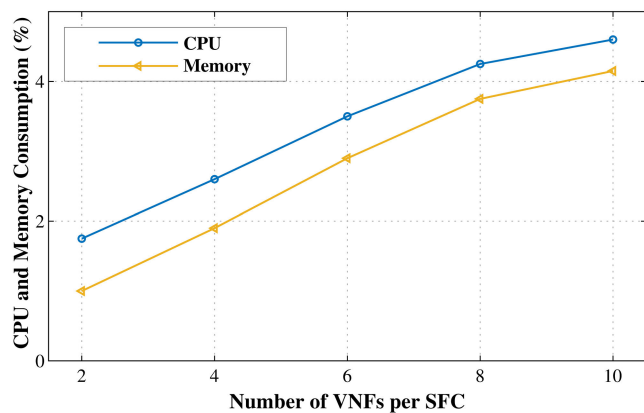


FIGURE 16. VNFs resource consumption.

The implementation of QoS policies based on an HTB discipline per service inside this VNF guarantees the bandwidth demanded by each application. As can be seen, App\_A, App\_B, and App\_C can achieve a maximum throughput of 4 Mbps, 6 Mbps, 10 Mbps respectively, avoiding the competition for the bandwidth previously shown.

The tests described have been complemented with a preliminary evaluation of the resources consumed by the proposed architecture when deploying multiple VNFs. The virtual scenario has been deployed on an Ubuntu 18.04 system equipped with an Intel Core i7 8700 (3.2 GHz) and 24 GB of RAM. Fig. 16 shows the memory and CPU consumption according to the variation of the number of VNFs deployed. As expected, there is a linear relationship between the number of VNFs deployed and the consumed resources. However, it is important to take into account that resources consumption mostly depend on the kind of service that the VNF provides. In this light, using containers to implement the VNFs included in vMANDOs represents an appropriate choice to keep resources consumption to a minimum. Besides, the use of Vim-emu, based on docker containers, made it possible to run the whole evaluation scenario in a single machine.

TABLE 2. API Proposed for the SDOAN.

Services	Arguments
Service Creation Request	<ul style="list-style-type: none"> <li>- Type of service (Internet, Multimedia, Telephony)</li> <li>- Upstream guaranteed BW</li> <li>- Upstream excess BW</li> <li>- Downstream guaranteed BW</li> <li>- Downstream excess BW</li> <li>- VLAN</li> </ul>
Service Creation Reply	<ul style="list-style-type: none"> <li>- Feasibility: "yes" or "not"</li> <li>- Service Id</li> </ul>
Service Update Request	<ul style="list-style-type: none"> <li>- Service Id</li> <li>- Upstream guaranteed BW</li> <li>- Upstream excess BW</li> <li>- Downstream guaranteed BW</li> <li>- Downstream excess BW</li> <li>- VLAN</li> </ul>
Service Update Reply	<ul style="list-style-type: none"> <li>- Feasibility: "yes" or "not"</li> <li>- Service Id</li> </ul>
Service Discard Request	<ul style="list-style-type: none"> <li>- Service Id</li> </ul>
Service Discard Reply	<ul style="list-style-type: none"> <li>- Feasibility: "yes" or "not"</li> </ul>
ONT ID Request	<ul style="list-style-type: none"> <li>- MAC Address ONT</li> </ul>
ONT ID Reply	<ul style="list-style-type: none"> <li>- ONT Id</li> </ul>
Service Subscription Request	<ul style="list-style-type: none"> <li>- ONT MAC address</li> <li>- ONT Id</li> <li>- Service Id</li> </ul>
Service Subscription Reply	<ul style="list-style-type: none"> <li>- Feasibility: "yes" or "not"</li> </ul>
Service Unsubscription Request	<ul style="list-style-type: none"> <li>- ONT MAC address</li> <li>- ONT Id</li> <li>- Service Id</li> </ul>
Service Unsubscription Reply	<ul style="list-style-type: none"> <li>- Feasibility: "yes" or "not"</li> </ul>

Services offered by the SDOAN (OLT's configurable services) for the creation, modification and deletion of communication services in the network

This simplified the testing process, compared to the Open-Stack based tests that needed additional servers.

### E. SDOAN API

As mentioned before, due to the lack of maturity and availability of reference implementations of APIs to dynamically

control the service QoS in access networks, it was decided to define and implement a simple QoS REST API interface to interact with the GPON equipment used in the validation scenario (Fig. 10). Table 2 summarizes the functionalities that the API provides.

This initial version of the API basically allows: the creation and modification of the services defined in GPON (Service Creation/Update primitives), specifying the QoS parameters associated to the service; and the subscription of ONTs to that services. The API is being implemented over the command line interface provided by the TELNET-RI OLT and it will be latter integrated and tested into the user-centric management architecture. Additional tests carried out in relation to QoS control in the access network are described in [34]. Therefore, an SDN-GPON solution using OpenFlow that permits residential users efficiently, fastly and dynamically adjust their bandwidth demands is proposed and experimentally validated. In fact, a dynamic network scenario in which users, by means of a management app, have a contracted basic bandwidth has been set up. In this scenario, users are allowed to increase the bandwidth temporarily when using highly demanding services.

## VII. CONCLUSIONS AND FUTURE WORKS

Traditional or legacy residential networks provide limited quality of service (QoS) support. As a consequence, services or applications performance is not as expected and, therefore, the QoE is poor. In addition, the ISP is restricted from offering new end services and fully meeting user demands.

The differentiated traffic treatment proposal allows providing better and new network connectivity services to meet the QoS requirements of applications that run at home. The SDN/NFV based network architecture designed to support the differentiated traffic treatment, represents a novel network ecosystem that upgrades the residential network architecture. Beyond addressing the current drawbacks of residential networks, the presented proposal provides an evolvable network environment to adapt itself to future requirements of users and multimedia services, as well as new emerging technologies.

The implementation of this approach consists of two stages. Firstly, the traffic belonging to an application is differentiated by creating a traffic flow on the SDN-based RGW. Secondly, a specific SFC is created at vMANDO and it is associated to the traffic flow previously created.

Certainly, the collaborative operation between SDN and NFV allows complementing their individual contributions and it makes possible the differentiated traffic treatment implementation. Nevertheless, the operations to be carried out both at SDN and NFV level must be orchestrated. In this light, the Traffic Treatment app is defined to interact with NFV-MANO and the RGW-based RGW to create the required traffic flows, traffic containers and SFCs following the user orders. With this approach, on one hand, the user specifies what application will have a specific treatment and, on the other hand, what will the specific treatment consist of.

Considering the NERG architecture and R-CORD as the most notable examples of the RGW virtualization paradigm, the differentiated traffic treatment with QoS support proposal in conjunction with the user-centric management approach represent an extraordinary improvement. However, despite the promising results obtained from the experimental validation, there are unresolved issues that require further analysis.

As the Traffic Treatment app functionalities depend on the life cycle operations of descriptors and instances that MANO interfaces provide, the NFV platform to be used must accurately implement every single functionality and operation defined by ETSI. In this light, NFV platforms like ONAP or OpenBaton<sup>13</sup> can be used in future experiments.

According to ETSI, after the instantiation of a VNF, a configuration interface is created to set the functional parameters of the service that the VNF provides. This interface is used by the Element Management (EM) or the VNFM of the reference architecture. Considering the user-centric approach, this configuration interface must also be used by the management system to allow the user to modify the VNF's configuration. In this light, a VNF client can be integrated to the Interaction layer of the Management architecture which would be in charge of communicating with the VNF configuration interface. This VNF client mainly would provide a graphical configuration interface and would perform the EM tasks.

The presented proposals show how, from a centralized location, all the functional components of the residential network such as the SDN-based RGW, the SDOAN and the SFCs can be managed and controlled. Moreover, by means of the vMANDO concept, the SDN and NFV components can be grouped into a single entity. These particular characteristics allow thinking that the provision of the Residential Network as a Service (RNaaS) is a true option. In this light, the ISP is provided with an agile and flexible environment to optimize the life cycle control of residential networks and the creation of new business models is encouraged. The RNaaS concept provides the user the ability of defining and customizing the functional characteristics of both the residential network and the management system. The RNaaS concept allows the residential network to adapt itself to the user needs.

Certainly, the user-centric approach aids in simplifying the management tasks of the residential network. Nonetheless, in advanced scenarios, Artificial Intelligence based components can be integrated to support the management tasks. For example, each VNF instance has a default or startup configuration which can be modified during the lifetime of the instance. This startup configuration might not be adequate considering the current state of the virtual environment where it will run. Thus, a mechanism can be proposed to infer a startup configuration considering different functional parameters of the VNF execution environment such as the available computational resources or network congestion. Besides, the user can be recommended to create a certain SFC

<sup>13</sup><https://openbaton.github.io/>



to improve the performance of the residential network. This SFC recommendation system would use network status information such as applications running at the user's home, type of user's devices, parental control policies or SFCs previously created to provide a specific recommendation.

## ACKNOWLEDGMENT

The authors would like to thank C. Vega for his contribution to the development and testing of the validation scenarios presented in this paper, carried out as part of his Master's thesis.

## REFERENCES

- [1] *A Generic Home Network Architecture With Support for Multimedia Services*, document ITU-T Rec. H.622, 2008, Accessed: May 2, 2019. [Online]. Available: <https://www.itu.int/rec/T-REC-H.622-200806-I/en>
- [2] Broadband Forum. (2004). *TR-094: Multi-Service Delivery Framework for Home Networks*. Accessed: May 2, 2019. [Online]. Available: <https://www.broadband-forum.org/technical/download/TR-094.pdf>
- [3] R. F. Moyano, D. F. Cambroner, and L. B. Triana, "A user-centric SDN management architecture for NFV-based residential networks," *Comput. Standards Inter.*, vol. 54, pp. 279–292, Nov. 2017.
- [4] R. F. Moyano, D. Fernandez, L. Bellido, N. Merayo, J. C. Aguado, and I. de Miguel, "NFV-based QoS provision for software defined optical access and residential networks," in *Proc. IEEE/ACM 25th Int. Symp. Quality Service (IWQoS)*, Jun. 2017, pp. 1–5.
- [5] ETSI. (2014). *Network Function Virtualisation (NFV); Management and Orchestration v1.1.1*. Accessed: May 22, 2019. [Online]. Available: [http://www.etsi.org/deliver/etsi\\_gs/NFV-MAN/001\\_099/001/01.01.01\\_60/gs\\_NFV-MAN001v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf)
- [6] K. J. Kerpez, J. M. Cioffi, G. Ginis, M. Goldberg, S. Galli, and P. Silverman, "Software-defined access networks," *IEEE Commun. Mag.*, vol. 52, no. 9, pp. 152–159, Sep. 2014.
- [7] OPNFV & OpenDayLight. (2016). *Virtualizing Customer Premises with Service Function Chaining*. Accessed: May 7, 2019. [Online]. Available: [https://www.opnfv.org/wp-content/uploads/sites/12/2016/11/opnfv\\_odl\\_vcpe\\_sfc\\_brief.pdf](https://www.opnfv.org/wp-content/uploads/sites/12/2016/11/opnfv_odl_vcpe_sfc_brief.pdf)
- [8] ETSI. (2013). *Network Functions Virtualisation, Use Cases (V1.1.1)*. Accessed: May 7, 2019. [Online]. Available: [http://www.etsi.org/deliver/etsi\\_gs/nfv/001\\_099/001/01.01.01\\_60/gs\\_nfv001v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/nfv/001_099/001/01.01.01_60/gs_nfv001v010101p.pdf)
- [9] Broadband Forum. (2016). *TR-317 Network Enhanced Residential Gateway*. Accessed: May 7, 2019. [Online]. Available: <https://www.broadband-forum.org/technical/download/TR-317.pdf>
- [10] Broadband Forum. (2016). *TR-345 Broadband Network Gateway and Network Function Virtualization*. Accessed: May 7, 2019. [Online]. Available: <https://www.broadband-forum.org/technical/download/TR-345.pdf>
- [11] L. Peterson, A. Al-Shabibi, T. Anshutz, S. Baker, A. Bavier, S. Das, J. Hart, G. Palukar, and W. Snow, "Central office re-architected as a data center," *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 96–101, Oct. 2016.
- [12] Broadband Forum. (2018). *TR-384 Cloud Central Office Reference Architectural Framework*. Accessed: May 13, 2019. [Online]. Available: <https://www.broadband-forum.org/download/TR-384.pdf>
- [13] J. Ordóñez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 80–87, May 2017.
- [14] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2429–2453, 3rd Quart., 2018.
- [15] K. Han, S. Li, S. Tang, H. Huang, S. Zhao, G. Fu, and Z. Zhu, "Application-driven end-to-end slicing: When wireless network virtualization orchestrates with NFV-based mobile edge computing," *IEEE Access*, vol. 6, pp. 26567–26577, 2018.
- [16] S. Chaabnia and A. Meddeb, "Slicing aware QoS/QoE in software defined smart home network," in *Proc. IEEE/IFIP Netw. Operations Manage. Symp. (NOMS)*, Apr. 2018, pp. 1–5.
- [17] A. S. Thyagaturu, A. Mercian, M. P. McGarry, M. Reisslein, and W. Kellerer, "Software defined optical networks (SDONs): A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2738–2786, 4th Quart., 2016.
- [18] R. G. Clegg, J. Spencer, R. Landa, M. Thakur, J. Mitchell, and M. Rio, "Pushing software defined networking to the access," in *Proc. 3rd Eur. Workshop Softw. Defined Netw.*, Sep. 2014, pp. 31–36.
- [19] S. S. W. Lee, K.-Y. Li, and M.-S. Wu, "Design and implementation of a GPON-based virtual OpenFlow-enabled SDN switch," *J. Lightw. Technol.*, vol. 34, no. 10, pp. 2552–2561, May 15, 2016.
- [20] R. F. Moyano, D. Fernández, L. Bellido, and C. González, "A software-defined networking approach to improve service provision in residential networks," *Int. J. Netw. Manage.*, vol. 27, no. 6, p. e1984, Nov. 2017.
- [21] L. Qu, C. Assi, and K. Shaban, "Delay-aware scheduling and resource optimization with network function virtualization," *IEEE Trans. Commun.*, vol. 64, no. 9, pp. 3746–3758, Sep. 2016.
- [22] D. Krishnaswamy, R. Kothari, and V. Gabale, "Latency and policy aware hierarchical partitioning for NFV systems," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2015, pp. 205–211.
- [23] H. Chang, A. Hari, S. Mukherjee, and T. V. Lakshman, "Bringing the cloud to the edge," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2014, pp. 346–351.
- [24] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, May 2016.
- [25] E.-D. Kim, S. Lee, Y. Choi, M.-K. Shin, and H.-J. Kim, "Evaluations of a multiple flow tables construction scheme for service function chaining," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2016, pp. 424–426.
- [26] T. Cejka and R. Krejci, "Configuration of open vSwitch using OFCONFIG," in *Proc. IEEE/IFIP Netw. Operations Manage. Symp. (NOMS)*, Apr. 2016, pp. 883–888.
- [27] H. Khalili, D. Rincón, S. Sallent, and J. R. Piney, "An integrated SDN-based architecture for passive optical networks," in *Broadband Communications Networks: Recent Advances and Lessons from Practice*. London, U.K.: IntechOpen, 2018, p. 215, doi: 10.5772/intechopen.72036.
- [28] Broadband Forum. (2019). *TR-385 ITU-T PON YANG Modules*. Accessed: Jul. 5, 2019. [Online]. Available: <https://www.broadband-forum.org/download/TR-385.pdf>
- [29] M. Peuster, H. Karl, and S. van Rossem, "MeDICINE: Rapid prototyping of production-ready network services in multi-PoP environments," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2016, pp. 148–153.
- [30] M. Peuster, M. Marchetti, G. García de Blas, and H. Karl, "Automated testing of NFV orchestrators against carrier-grade multi-PoP scenarios using emulation-based smoke testing," *J. Wireless Comput. Netw.*, vol. 2019, no. 1, p. 172, Dec. 2019.
- [31] OASIS Committee Specification Draft 03. (2016). *TOSCA Simple Profile for Network Functions Virtualization (NFV) Version 1.0*. Accessed: May 6, 2019. [Online]. Available: <http://docs.oasis-open.org/tosca/tosca-nfv/v1.0/csd03/tosca-nfv-v1.0-csd03.html>
- [32] C. Vega. (2018). *GREDOs Project Scenarios Repository*. Accessed: Jun. 5, 2019. [Online]. Available: <https://github.com/carlosv5/GREDOS>
- [33] D. Fernández, F. J. Ruiz, L. Bellido, E. Pastor, O. Walid, and V. Mateos, "Enhancing learning experience in computer networking through a virtualization-based laboratory model," *Int. J. Eng. Educ.*, vol. 32, no. 6, pp. 2569–2584, 2016.
- [34] J. Azofra, N. Merayo, J. C. Aguado, I. de Miguel, R. Durin, F. J. Ruiz, C. Vega, D. Fernandez, P. Fernandez, R. M. Lorenzo, and E. J. Abril, "Implementation of a testbed to analysis a SDN based GPON," in *Proc. Eur. Conf. Opt. Commun. (ECOC)*, Sep. 2018, pp. 1–3.



**RICARDO FLORES MOYANO** (Member, IEEE) received the degree in electronic engineering from Universidad Politécnica Salesiana (UPS), in 2006, and the M.S. and Ph.D. degrees in telematics systems engineering from the Universidad Politécnica de Madrid (UPM), in 2013 and 2018, respectively. He is currently a full-time Professor with the Department of Computer Science Engineering, Universidad San Francisco de Quito (USFQ). His research interests include SDN, NFV, cloud networking, the future Internet, and 5G networks.



**DAVID FERNÁNDEZ** received the M.S. degree in telecommunications engineering and the Ph.D. degree in telematics engineering from the Universidad Politécnica de Madrid (UPM), Spain, in 1988 and 1993, respectively. Since 1995, he has been an Associate Professor with the Department of Telematics Systems Engineering (DIT), UPM. His current research interests focus on software-defined networks, network virtualization, cloud computing datacenters technologies, and network security.



**CARLOS M. LENTISCO** received the M.S. and Ph.D. degrees in telecommunications engineering from the Universidad Politécnica de Madrid (UPM), Spain, in 2014 and 2019, respectively. He is currently an Assistant Lecturer with UPM, specializing in the fields of computer networking, multimedia services, and the Internet technologies. His current research interests include virtualization, software-defined networking, and mobile broadcast services.



**NOEMÍ MERAYO** received the degree in telecommunication engineering from Valladolid University, Spain, in February 2004, and the Ph.D. degree from the Optical Communication Group, University of Valladolid, in July 2009. Since 2005, she has been working as a Junior Lecturer with the University of Valladolid. She has been a Visiting Research Fellow with the University of Hertfordshire, London, working in the Optical Networks Group, in the group of Broadband Optical Transmissions (TOyBA), University of Zaragoza, and recently in the Technical University of Munich (TUM). Her research is focused on the design and performance evaluation of optical networks, especially passive optical networks.



**ANDRÉS CÁRDENAS** received the master's degree in network engineering and telematic services from the Universidad Politécnica de Madrid, Spain, where he is currently pursuing the Ph.D. degree with the Departamento de Ingeniería de Servicios Telemáticos (DIT). His researches focus on NFV/SDN environments, network slicing, and 5G Networks.

...