Building an NFV-Based vRGW: lessons learned

Jorge Proença, Tiago Cruz, Paulo Simões
CISUC - Dep. of Informatics Engineering
University of Coimbra
Coimbra, Portugal
{jdgomes, tjcruz, psimoes}@dei.uc.pt

Gonçalo Gaspar, Bruno Parreira, Alexandre Laranjeira, Fernando Bastos
Altice Labs
Aveiro, Portugal
{goncalo-n-gaspar, Bruno-m-parreira, alexander-s-laranjeira, fbastos}@alticelabs.com

***Abstract*—The residential gateway (RGW) is a widely deployed device in the context of telecommunication services such as triple play and internet access. Designed to make the connection between the customer home and the operator infrastructure, it provides wired and/or wireless connectivity capabilities, also handling services such as Domain Name Service (DNS) proxying, routing, Network Address Translation (NAT) or firewalling, among others. Its increased complexity, together with other factors – such as device-related costs, or the increased reliance on RGWs for providing critical services – has prompted an interest in its virtualization, also motivated by the evolution of network and service-centric virtualization concepts.**

This paper presents an approach for the virtualization of the residential gateway (vRGW). It starts by giving an overview of the concept, explaining it and pointing out its benefits. It also explains the virtualization techniques and paradigms that have pushed this movement, namely software defined networking (SDN), network function virtualization (NFV), and service function chaining (SFC). Additionally, it describes the implementation of the vRGW architecture, including a description of its components and their integration.

Keywords—Software-Defined Networking; Network Function Virtualization; Virtual RGW; Service Function Chaining

I. INTRODUCTION

Access network infrastructures are evolving at a rapid pace. Network service providers (NSP) are increasing the number of Fiber-To-The-Premises (FTTx) deployments, enabling better and faster connections from the customer premises to the operator's infrastructure. At the same time, services are getting increasingly more complex, which leads to more devices being deployed at the customer premises and consequently, an increased capital expenditure (CAPEX) for the operator [1].

One important component of this service is the residential gateway (RGW). These are feature-rich embedded systems, whose role and features have remained mostly unchanged for some time, being responsible for the mediation between the operator and customer premises domains, also providing a number of services to the home devices (such as firewall or web filtering). Despite its importance, RGWs represent a burden for the telecommunications operator, for several reasons, such as cost (both CAPEX and operating expense-related (OPEX)), reliability concerns or even management issues [2].

To overcome the limitations of the current RGW-based model, operators are looking at the virtualization of the residential gateway (vRGW) as a way of reducing complexity, minimizing costs, improving service quality, and reducing the time-to-market of new services as well.

The fundamental vRGW concept is built around the idea of transforming the original device into an abstract service entity composed by Virtual Network Functions (VNFs), while leaving behind a simple physical device to bridge the local network devices (computers, set-top-boxes, telephones, etc.) with the access network. While there are some efforts in this direction being made by operators such as Telefonica [3], there is little information regarding existing vRGW implementations.

This paper presents the architecture and prototype for an NFV-based vRGW that can significantly reduce service design time, while coexisting with current (legacy) architectures. The remainder of this paper is organized as follows: section 2 introduces the concept of the virtual residential gateway (vRGW), listing some drawbacks of the traditional model and expected benefits to be gained by its virtualization. It also provides an overview and description of some of the virtualization concepts that are pushing the development of the vRGW, namely software defined networking (SDN), network function virtualization (NFV), and service function chaining (SFC). Section 3 lists existing proposals related with the virtualization of the RGW. Section 4 presents an implementation of an NFV-based vRGW, describing the main solution components and their interactions. Section 5 describes the test scenario used to validate the implementation. Finally, section 6 concludes the paper.

II. VIRTUAL RGW CONCEPT

For telecommunication service providers (TSPs), the RGW is a device that, despite its critical role, represents a legacy heavily influenced by aspects such as the constrains of the IPv4 address space, the nature of mature service distribution models or even technological reasons. Despite this, the RGW is a cornerstone of modern broadband access networks, providing connectivity between the TSP and customer network domains, while hosting several services such as firewalling, DHCP (Dynamic Host Configuration Protocol), DNS, NAT (Network Address Translation), and other protocols and services related with the delivery of converged services such as IPTV (using for example Internet Group Management Protocol proxies) [4] or SIP (Session Initiation Protocol) [5].

Increased device complexity, together with the shift towards evolved service models and the increase in the number of subscribers and connected households, is turning the RGW into a burden for TSPs. In this perspective, the idea of

This work was partially funded by Altice Labs (in the scope of Project HolisticSDN) and by the ATENA H2020 EU Project (H2020-DS-2015-1 Project 700581).

streamlining the device using virtualization techniques constitutes a sound proposition for TSPs.

RGW virtualization consists primarily on moving the functions and services from the customer premises to the operator infrastructure. This also means that the device deployed at the customer premises can be replaced with a simpler and less complex device (such as a bridge, for that matter) that is solely responsible for establishing the connection between the customer and the operator domains. Although not new, this idea has been proving to be a difficult task for operators to develop and deploy for a number of years due to several reasons such as technical limitations (hosting a massive number of virtualized functions is a considerable challenge, considering that some operators have millions of subscribers), as well as managing them in an efficient manner.

A. Limitations of the Traditional RGW Model

The traditional model of developing and deploying a RGW has a few limitations and drawbacks. First, there is the high deployment cost that is involved in starting/upgrading a telecommunication service [6], where the RGW contributes with a significant share of the cost. There are also technical limitations of the solution: for example, it is a barrier for remote diagnostics and troubleshooting of devices and services within the customer domain (e.g., due to NAT translation).

The RGW plays a critical role for the introduction of new services, which are often dependent on specific device support. However, the high fragmentation and diversity of different hardware models, which may have different firmware versions within each model, may even result in a lower time to market of new services, as each RGW model requires its own specific customizations [3]. Furthermore, this diversity may also compromise uniformity and hamper troubleshooting and management operations, due to a lack of uniform service sets and/or management capabilities.

Overall, RGWs impose a considerable cost for the TSP (acquisition and operation), besides constituting a single point of failure for all the services offered to residential customers. In this perspective, RGWs are ideal candidates for virtualization, helping to relieve the TSP, while providing benefits to end-users, by easing the introduction of new and reliable services or even due to power consumption benefits (because of the improved energy efficiency of a virtualized RGW).

B. New Opportunities for the vRGW

Factors such as the reduction of the deployment costs or the evolution towards a simpler device with less services running would lead to a more economical RGW. Moreover, the simplification of the device would lead to a reduced failure rate [7], which would mean less expense for the operator and increased customer satisfaction and loyalty. This will in turn lead to the reduction of other expenses such as call centre cost, which some estimate a reduction up to 90%, and product return cost to 46% from the virtualization of the RGW [1]. Additionally, time to market of new services can be significantly reduced if part of the RGW is converted into a piece of software, since it will not have hardware-related dependencies such as in the traditional scenario.

Nevertheless, the implementation of these concepts in a production setting will have to fulfil to some requirements that were not relevant in the current RGW model and have been pointed out by some authors in [8] and [9] such as the increased load (in several aspects such as processing, storage and networking resources), coexistence with the existing legacy infrastructure, or the security and privacy of the instances that are going to be shifted from the customer to the operator domain, among other aspects. These proved challenging for several previously presented proposals for RGW virtualization – as an example, [10] addressed the limitations of virtualizing an RGW in a straightforward manner (as a single, self-contained, virtual machine instance), demonstrating the scalability constrains of such an approach.

C. Underlying Paradigms

Several problems faced by previous attempts for RGW virtualization were due to the lack of a framework for network and service-centric functional virtualization, required to fulfil the flexibility and scalability demands of the proposed concept, while helping to solve other important issues, such as functional component placement and deployment. This section introduces and describes these developments, namely SDN, NFV and SFC, which are proving instrumental in driving the recent research efforts towards the implementation of a vRGW.

1) Software Defined Netwoking

Software Defined Networking (SDN) is a recent paradigm of programmable networks [11], [12]. Among the available SDN supporting protocols, OpenFlow [13] is one of the most widespread. Having started as a research project at the University of Stanford, it eventually became one of the first SDN-enabling standards.

The main concept of SDN is the separation of roles within a network architecture. The control plane is moved from the forwarding elements (e.g., routers) and hosted in a logically centralized server. The forwarding elements, in turn, continue to host the forwarding plane, which is responsible for transmitting the packets to the next destination. As a result, the control plane (responsible for making the decisions) has a broad view of the network, due to being logically centralized.

Overall, an SDN compliant network has a number of benefits over a traditional one. For instance, network management is improved as well as network flexibility. In an OpenFlow-enabled scenario, traffic rule is flow based, and it can be dependent on a number of traffic characteristics (such as switch port or MAC address) either from the source or the destination. Also, its flexibility allows traffic to be steered *on the fly,* a concept which contrasts with the rather fixed nature of traditional networking, fulfilling the requirements for the implementation of SFC capabilities (despite not being the only technology capable of doing so, it is one of the best suited for that purpose).

2) Network Function Virtualization

For a number of years, TSP infrastructures have been built with appliances made up by close-coupled hardware and firmware. Recent developments have brought a change where hardware and firmware are separated and the latter moved to a commercial off the shelf (COTS) servers as software instances. This is the main idea of Network Function Virtualization (NFV), to move the network components (such as firewalls, load balancers or NAT) to generic software capable of running in common servers [14], [15], [16]. The Network Functions Industry Specification Group of the ETSI [17] is working on promoting the advance and standardization of NFV.

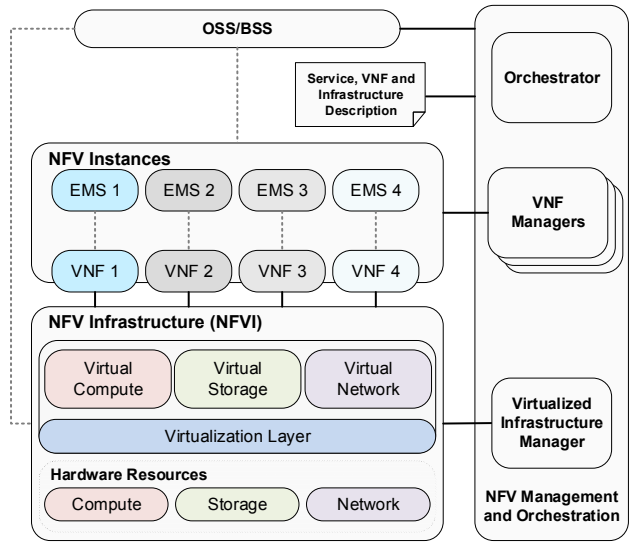

Fig. 1. ETSI NFV Architectural Framework

Fig. 1 depicts the ETSI NFV reference framework. It is composed by several modules, namely: the Network Function Virtualization Infrastructure (NFVI), providing the required resources (hardware/servers, accelerators and the virtualization layer) which support the VNFs; the VNF domain, hosting the VNF instances and their corresponding Element Management Systems (EMS) for integration with Operations Support Systems and Business Support Systems (OSS/BSS), when applicable; finally, the NFV Management and Orchestration (MANO or M&O) domain orchestrates and manages the lifecycle of physical and/or software infrastructure resources, as well as the lifecycle of services and their VNFs.

3) Service Function Chaining

The shift of functions to software components, together with the benefits introduced by SDN has enabled the possibility of changing the network structure and topology in a flexible way, allowing to forward traffic between VNFs in a dynamic fashion, improving and simplifying the provision of structured service abstractions composed of linked network functions (technically referred as function chains) in a cost-effective manner [18].

In an NFV-centric perspective, the concept of service function chaining (SFC) [19] is the interconnection of the functions that compose a virtual network service into a chain (which is an instantiation of a graph). In an NFV-enabled scenario, the individual nodes that compose a chain can be instantiated as VNFs running in the infrastructure. In turn, the interconnection of the chains can be enabled by SDN, which will be responsible for managing the switches for them to perform the correct traffic steering between the VNFs. An example of a VNF service chain is illustrated in Fig. 2 (the acronym PNF stands for "Physical Network Function", such as a carrier-grade NAT appliance).

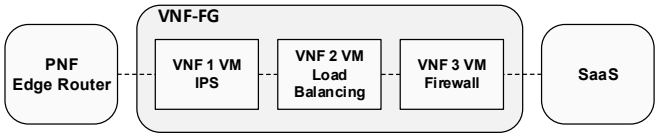

Fig. 2. Forwarding Graph of three VNF chained into a service

The IETF has created a group focused on this topic, the IETF Service Function Chaining Working Group (IETF SFC WG) [20]. One of its objectives is to produce architectures using SFC on networking scenarios.

III. Related Work

The idea of virtualizing the elements related to an RGW is not new and has been previously discussed in the literature. Some proposals have been published for example in [10], [21], [22], and [23]. Still, a number of issues were making it a very difficult task, such as technical challenges to handle issues like scalability and manageability. Recently, the rise of virtualization techniques such as NFV (described in section II.C) gave way of new proposals for the virtualization such as [9], [1], and [24]. Moreover, the Spanish telecom operator Telefonica has been performing trials in Brazil during 2014 [3].

However, the available literature is focused on the high-level architectural concept for such a device. This paper goes a step further by providing insights about a proof-of-concept implementation that was undertaken in a joint-effort between a telecommunications operator research lab and a university.

IV. Implementation of a Virtualized RGW (vRGW)

The implementation of the vRGW prototype takes a similar approach to the ETSI OSM initiative presented in [25]. In this initiative, an end-to-end orchestrator is used in collaboration with the NFV Orchestrator (NFVO). The first element establishes the connection point between BSS and OSS components and the network infrastructure while delegating to the second the management of the datacentre-based service components. In this hierarchical architecture, the integration of legacy components and novel cloud-based components is done seamlessly, making use of what is already available in terms of management systems in network operators.

A. vRGW Logical Architecture

A simplified version of a traditional RGW deployment is composed of the following elements:

- Optical Network Termination (ONT)/RGW – the device that performs routing between the customer home network and the provider optical network;
- Optical Line Terminal (OLT) – this device aggregates the traffic from all nearby ONTs (access network) and delivers them to the BNG;

- Broadband Network Gateway (BNG) – this device performs routing between the access/aggregation networks and the provider core network.

Although the end goal in the virtualization of the RGW is the movement of all the services provided by these components to a cloud environment, this prototype is focused only on the ONT services. With that in mind, a datacentre was deployed to instantiate the virtual resources that support these services. This datacentre connects to the BNG through a Multiprotocol Label Switching (MPLS) network, as shown in Fig. 3.

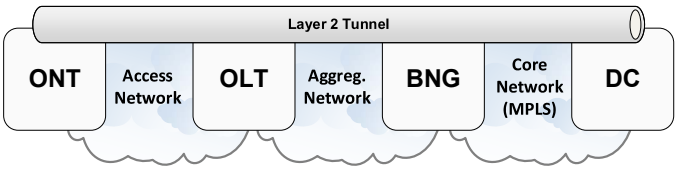

Fig. 3. vRGW Infrastructure

To connect the devices located in the local network to the services available in the datacentre, a layer 2 Generic Routing Encapsulation (GRE) tunnel is established between the ONT and the private domain within the datacentre where the virtual resources are deployed. This way, from the local network devices' point-of-view, there is no difference between traditional RGWs and vRGWs.

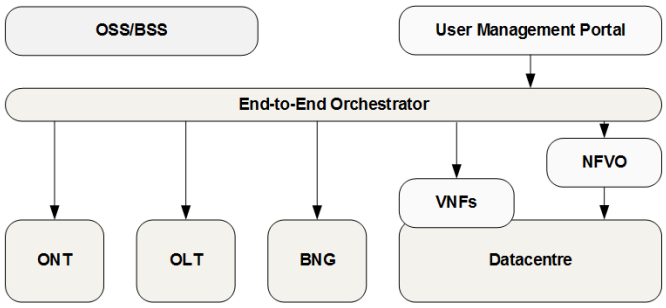

Fig. 4. vRGW Prototype

Fig. 4 shows the complete prototype logical architecture where the following components were added:

- User-management Portal – this Altice Labs proprietary component allows the end-client to configure the vRGW and the use of VNFs per device when connected to the home network;
- End-to-End Orchestrator – this component is responsible for the management of end-to-end services including the physical and virtual devices that support it. In this prototype a proprietary Altice Labs product, Network Activator (NA), is used to perform this function;
- NFVO – this component manages virtual resources in the datacentre that take part in network services. A customized version of Telefonica's OpenMANO [26] is used to realize this component;
- DC – this component provides virtual resources as a service for the deployment of VNFs. In this prototype the OpenStack platform [27] with some extensions is used to realize this component.

The following sections will provide further insight on the novel elements.

B. *Custom OpenStack*

In this prototype, the OpenStack platform (IceHouse release) was used, integrated with the following extensions:

- Attachment-Point – this functionality enables the establishment of the layer 2 tunnels, in this case GRE tunnels, to virtual networks in OpenStack. This extension is presented in [28];
- DHCP-Radius – this extension enables AAA on devices located in the local network by using the DHCP server to send Radius messages to an AAA server. This extension is presented in [29];
- Traffic Steering – this extension enables SFC on OpenStack through an ordered list of traffic redirections. This extension is covered in [30].

The first extension enables the crossing of multiple domains transparently and with minimum impact on traditional operations. This is important for the near-future coexistence of novel and legacy services on top of the same infrastructure. The other two components enable value-added services such as deploying content filtering VNFs for children devices only. This is possible by using per device authentication and SFC to redirect traffic to the URL filter VNF.

C. *User-management Portal*

This proprietary portal is deployed for the end-client to enable the configuration of the vRGW and register devices that are connected to the home network. By registering devices, users are not only able to configure the available services for them but are also able to set a default configuration for un-registered ones. This approach provides the means for implementing a shared-management model in which the end-client can manage the vRGW instance, akin to the local device management capabilities of conventional RGWs.

The management portal also provides the configuration interface for the included services, namely: content-filtering, application-filtering and firewall capabilities. The content-filtering service enables the user to choose which Internet content should be filtered for a specific device, (e.g., webpages containing adult content). The application-filtering service provides the means for the user to be able to block specific network-based applications from specific devices (e.g., online-gaming). Finally, the firewall service implements the traditional firewall capabilities available in most consumer routers.

Each time the users access the portal to set a specific device configuration, the portal communicates with the End-to-End Orchestrator through a secure interface. The latter will then interact with the infrastructure bellow to complete the configurations.

D. *OpenMANO*

In the ETSI model (see Fig. 1), the MANO domain oversees the virtualization-specific tasks, encompassing the NFV Orchestrator, that manages network services on the

operator domain; the VNF Manager(s), which take care of VNF instances: and the Virtualized Infrastructure Manager (VIMs), responsible for managing the NFVI computing, storage and networking resources. The NFV framework is to be driven using a set of metadata for describing the Service, VNF and Infrastructure requisites, providing the MANO with information about available resources. This enables resource providers (VNF or Infrastructure) to develop compatible solutions that can be integrated within the same framework

For the MANO role, the vRGW proof-of-concept prototype resorted to the OpenMANO project, which has developed an open-source implementation of the ETSI MANO architecture. OpenMANO is used as an orchestrator to instantiate and orchestrate the infrastructure resources that support the functions, such as virtual machines and tenant networks. The virtual infrastructure manager used by OpenMANO to manage the NFVI was OpenStack.

The OpenMANO framework encompasses three main components: *openvim*, *openmano* and *openmano-gui*, which respectively provide a VIM, NFV orchestrator and web management GUI capabilities. In this prototype, several modifications were made to the *openmano* orchestrator to support some features that were not available at the time (work was based on a master branch commit from September of 2015). More specifically, the orchestrator component was modified to create, manage, and delete OpenStack routers, and to manage floating IPs and authentication keys of the OpenStack VMs.

E. Virtual Network Functions

The virtual network functions used in the proof-of-concept prototype include an URL filter and firewall.

The URL filter VNF uses the DansGuardian [31] web content filter to block specific webpages content by defining URLs and phrase-matching configurations, while the firewall VNF uses the firewall application for the Alpine Linux distribution [32] to implement the firewall service and to block specific applications.

These services were selected to provide a minimal NFV-based implementation of a vRGW with enough capabilities for integration and functional validation.

V. Test Scenario/Testbed

The test scenario consists on the complete service lifecycle, which includes the following stages:

- Service Design – by using the openmano-gui web GUI component the service designer can use the already available VNFs to define a new vRGW service composed of two VNFs. Although the complete service design also involves designing the service in the End-to-End Orchestrator, this is almost a one-time only process because the actual service uses the legacy infrastructure in an overlay fashion;
- Service Provisioning – when the ONT is turned ON, it will send a DHCP request to the BNG, which will reply with an IP address after going through the AAA processes. Moreover, the BNG will signal the End-to-End Orchestrator to start the instantiation of the virtual resources in the datacentre. Finally, the openmano orchestrator will be requested to instantiate the necessary resources for the network service;
- Service Runtime – after the service has been provisioned, the client may use the User-management Portal to configure the vRGW;
- Service Destruction - when the client wishes to terminate the service, the BSS and OSS systems interact with the End-to-End Orchestrator for the destruction of the resources. At this moment, the latter will signal the openmano orchestrator to destroy the resources and will re-configure the network infrastructure elements.

A. Service Design

The prototype provides a significant flexibility in terms of service design. A solution can be tailor made to the customer needs by structuring the needed VNFs. The OpenMANO framework allows for an easier VNF-based service design with the use of its GUI component. By providing a drag-and-drop web interface (illustrated in Fig. 5), the custom-made service can be built in an intuitive manner.

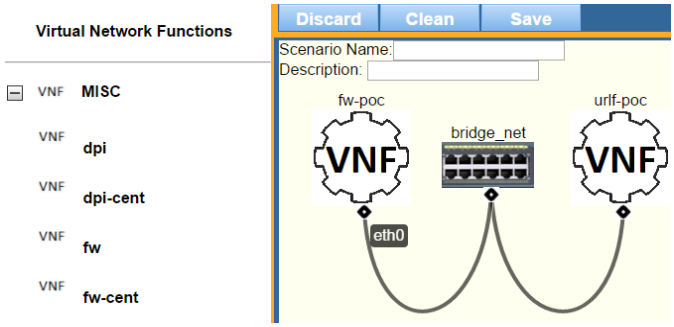

Fig. 5. OpenMANO GUI web interface

This is made by dragging the elements that compose the service (such as VNF components and networks) to the service area of the web interface and establishing the connections among them. This will allow the *openmano* orchestrator to provision all the necessary resources in the NVFI for the remaining components of the prototype to establish the connection between the customer functions and the operator's components, and to activate the service.

B. Functional Validation

Prototype validation was focused on the management and functional aspects of the platform in terms of provisioning and usability. Provisioning tests ensured the correct integration and operation of the MANO and OSS/BSS systems for resource and VNF instantiation, as well as the vRGW NFV service lifecycle management. Moreover, the User-management portal was also tested to demonstrate the possibility of blocking specific games and webpages on a per device basis. Each configuration took a couple of seconds to become active and without disrupting the service to other devices connected on the same wireless network.

Functional tests were focused on validating the usability of the prototype, using smartphones and laptops as consumer devices. Those devices were connected to the ONT wireless and wired networks and used to test/assess several use cases,

namely: webpage browsing, online gaming and consumption of media services. Such functional tests were executed with success, over several time spans and with different devices and users, in other to stress the platform.

VI. Conclusions

The present paper presented an implementation of an NFV-based virtualized residential gateway (vRGW). It started by presenting the concept of the vRGW and some recent developments that allowed operators to take a leap forward, namely, SDN and NFV. Additionally, the paper also presented an architecture and a proof-of-concept prototype for providing a vRGW where services are composed by NFV functions.

A key benefit of this architecture is the integration and co-existence with legacy infrastructures. This is an important point as it allows for an easier introduction of the new vRGW paradigm, which enables deployments on current infrastructures without having to go through major architectural changes in a short time. As a result, CAPEX is reduced since current deployments can go through the planned lifecycle with new concepts being deployed gradually.

In the future, there are more architectural components from legacy architectures that can be virtualized into the new paradigm. An example of such virtualization is the Optical Line Terminator (OLT), as pointed out as a use case in a proof-of-concept project by AT&T and ONOS: CORD (Central Office Re-architected as Datacenter).

References

[1] H. Xie et al., "vRGW: Towards network function virtualization enabled by software defined networking," *Proc. of the 21st IEEE Int. Conf. on Network Protocols (ICNP)*, 2013, doi: 10.1109/ICNP.2013.6733632

[2] T. Cruz, P. Simões, and E. Monteiro, "Optimizing the Delivery of Services Supported by Residential Gateways: Virtualized Residential Gateways," *Handb. Res. Redesigning Futur. Internet Archit.*, pp. 432-473, 2015. doi:10.4018/978-1-4666-8371-6.ch019

[3] R. Cantó Palancar, R. A. da Silva, J. L. Folgueira Chavarría, D. R. López, A. J. Elizondo Armengol, and R. Gamero Tinoco, "Virtualization of residential customer premise equipment. Lessons learned in Brazil vCPE trial," it - Information Technology., vol. 57, no. 5, pp. 285–294, 2015. doi: 10.1515/itit-2015-0028.

[4] H. Holbrook, S. Systems, and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast," RFC 4604, IETF, 2006.

[5] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler SIP:session Initiation Protocol, RFC 3261, IETF, 2002.

[6] Z. Bronstein and E. Shraga, "NFV virtualisation of the home environment," *Proc. of the IEEE 11th Consumer Communications and Networking Conf. (CCNC)*, 2014, doi: 10.1109/CCNC.2014.6940493.

[7] F. Sánchez and D. Brazewell, "Tethered Linux CPE for IP service delivery," *2015 1st IEEE Conf. on Network Softwarization (NetSoft)*, London, 2015, doi: 10.1109/NETSOFT.2015.7116166.

[8] ETSI GS NFV 002 V1.1.1, "Network Functions Virtualisation (NFV); Architectural Framework," 2013.

[9] J. J. Dustzadeh, "SDN: Time to Accelerate the Pace," keynote presentation at the Open Networking Summit 2013, Santa Clara, CA, US, April 4, 2013. Retrieved Jan 2016 from: http://www.slideshare.net/opennetsummit/ons2013-justin-joubine-dustzadehhuawei

[10] N. Egi, A. Greenhalgh, M. Handley, M. Hoerdt, L. Mathy and T. Schooley, "Evaluating Xen for Router Virtualization," *Proc. of 16th Int. Conf. on Computer Communications and Networks (ICCCN 2007)*. 2007, doi: 10.1109/ICCCN.2007.4317993

[11] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," in *Proc. of the IEEE*, V. 103, N. 1, 2015. doi: 10.1109/JPROC.2014.2371999

[12] K. Greene, "Tech Review 10 Breakthrough Technologies: Software-Defined Networking.," 2009. Retrieved May 2016 from: http://www2.technologyreview.com/article/412194/tr10-software-defined-networking/.

[13] B. Pfaff, B. Lantz, B. Heller, and others, "Openflow switch specification, version 1.3. 0," Open Networking Foundation, 2012.

[14] M. Chiosi et al., "Network Functions Virtualization, An Introduction, Benefits, Enablers, Challenges and Call for Action," SDN and OpenFlow SDN World Congress, 2012.

[15] M. Chiosi et al., "Network Functions Virtualisation - Network Operator Perspectives on Industry Progress," Updated White Paper, 2013.

[16] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network Function Virtualization: State-of-the-Art and Research Challenges," Commun. Surv. Tutorials, IEEE, vol. 18, no. 1, pp. 236–262, 2016. doi: 10.1109/COMST.2015.2477041

[17] ETSI Industry Specification Group, "Network Function Virtualization." http://portal.etsi.org/portal/server.pt/community/NFV/367.

[18] S. Sahhaf et al., "Scalable Architecture for Service Function Chain Orchestration," *2015 Fourth European Workshop on Software Defined Networks*, Bilbao, 2015, pp. 19-24. doi: 10.1109/EWSDN.2015.55.

[19] J. Halpern and C. Pignataro, "Service Function Chaining (SFC) Architecture," RFC 7665. IETF, 2015.

[20] "The Internet Engineering Task Force (IETF) Service Function Chaining (SFC) Working Group (WG)," 2015. Retrieved May 2016 from: https://datatracker.ietf.org/wg/sfc/charter/.

[21] N. Egi, A. Greenhalgh, M. Handley, M. Hoerdt, F. Huici, L. Mathy, and P. Papadimitriou, "A Platform for High Performance and Flexible Virtual Routers on Commodity Hardware," *SIGCOMM Computer Comm. Rev.*, vol. 40, no. 1, pp. 127–128, 2010. doi: 10.1145/1672308.1672332

[22] D. Basak, R. Toshniwal, S. Maskalik, and A. Sequeira, "Virtualizing Networking and Security in the Cloud," SIGOPS Oper. Syst. Rev., vol. 44, no. 4, pp. 86–94, 2010. doi: 10.1145/1899928.1899939

[23] D. Abgrall, "Virtual Home Gateway, How can Home Gateway virtualization be achieved?," EURESCOM, Study Rep. P.

[24] T. Cruz, P. Simões, N. Reis, E. Monteiro and F. Bastos, "An architecture for virtualized home gateways," *2013 IFIP/IEEE Int. Symposium on Integrated Network Management (IM 2013)*, Ghent, 2013, pp. 520-526.

[25] ETSI-OSM, "End-to-End Service Instantiation Using Open-Source Management and Orchestration Components - White Paper," Mobile World Congress 2016, 2016.

[26] Telefónica, "OpenMANO project page," 2015. Retrieved Jan 2016 from: https://github.com/nfvlabs/openmano.

[27] OpenStack, "OpenStack - Open Source Cloud Computing Software," 2016. Retrieved Jun 2016 from: https://www.openstack.org/.

[28] Igor D. Cardoso, "Network infrastructure control for virtual campuses," Master Thesis, University of Aveiro, 2014.

[29] V. A. Cunha, I. D. Cardoso, J. P. Barraca and R. L. Aguiar, "Policy-driven vCPE through dynamic network service function chaining," *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, Seoul, 2016, pp. 156-160. doi: 10.1109/NETSOFT.2016.7502463

[30] J. Soares et al., "Toward a telco cloud environment for service functions," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 98-106, Feb. 2015. doi: 10.1109/MCOM.2015.7045397.

[31] D. Barron, "DansGuardian - true web content filtering for all," 2016. Retrieved Jun 2016 from: http://dansguardian.org/.

[32] "Alpine Linux - Webpage," 2016. Retrieved Jun 2016 from: http://www.alpinelinux.org/.