An Architecture for Virtualized Home Gateways

Tiago Cruz, Paulo Simões, Nuno Reis, Edmundo Monteiro
DEI-CISUC – University of Coimbra
Coimbra, Portugal

Fernando Bastos, Alexandre Laranjeira
PT Inovação
Aveiro, Portugal

Abstract— the convergence of technical advances in the field of virtualization has enabled the consolidation and scaling of resources in a cost-effective way, a trend that has also found its way into the telecommunication operator infrastructure foundations, from data centers to networks alike. Starting from the core, the impact of these developments is reaching towards the edge of the infrastructure and into the access network.

In this spirit, we foresee the opportunity of virtualizing a key device of modern broadband access networks: the Residential Gateway (RGW). Presently located on the customer premises, the RGW stands between the home network and the access network. It imposes a considerable cost for the operator (acquisition and operation) and constitutes a single point of failure for all the services offered to the residential customers – such as Internet access, VoIP, IPTV and Video-on-Demand.

In this paper we propose an architecture for virtualized Residential Gateways (vRGWs) which physically removes the RGW from the customer premises, moving it into the operator data center as a virtualized entity. This solution potentially reduces deployment, maintenance and operation costs, whilst improving overall flexibility, reliability and manageability – both for the access network infrastructure and for provided services.

Keywords— Cloud Computing, Virtualization, Home Networks

I. INTRODUCTION

Looking into today's access network deployments, we see that Fiber-To-The-Premises (FTTx) network topologies are becoming predominant, with current DSL based topologies being gradually replaced by optical fiber. This opens an opportunity window for operators to rethink how some of their service offers are delivered, in order to reduce costs and improve flexibility and manageability.

Specifically, Residential Gateways (RGW), have remained mostly unchanged for some time. Standing on the customer premises, RGWs are feature-rich embedded systems that provide the interface between the home network and the operator's access network. In fact, RGWs handle local network services such as DNS, DHCP, NAT, routing, firewalling, and IEEE 802.11 [1] wireless connectivity, also providing direct support for added-value services such as IPTV (IGMP proxying [2] and VCI/VLAN management [3]) and VoIP (SIP [4] gateways and/or analog terminal adapters). However, in spite of its importance, RGWs represent a significant burden for the operator:

- The RGW device is relatively expensive. Even if this cost is subsidized by the operator or transferred to the customer, it represents a relevant share of the initial deployment costs.
- RGWs are relatively complex and prone to hardware failures and/or to misconfiguration, constituting a critical single point of failure often requiring on-site maintenance – a burden for the operator, due to the involved logistics, making it difficult to amortize its cost.
- Operator time to market is often dependent on the unit manufacturer to introduce new services to their subscribers, which is a serious penalty to pay, aggravated by the subsequent need to remotely upgrade thousands or millions of devices (an error-prone operation).
- It is difficult for the operator to keep a homogeneous set of RGWs, which affects manageability. Even if the operator adopts a single model from a single vendor, minor firmware and hardware revisions gradually compromise uniformity. In extreme cases, operators are even forced to ponder massive replacement of RGWs to support new services.

Considering this scenario, it would be attractive to consider alternative approaches able to overcome or at least mitigate these problems. It is obviously impossible to completely remove the RGW physical device from the customer premises: it will always be necessary to bridge the local network devices (computers, set-top-boxes, telephones, etc.) with the access network. Still, a considerable part of the functions currently hosted by the physical RGW device can be moved closer to the operator's infrastructure, thanks to advances in virtualization and access network technologies.

In this paper, we explore this idea of virtualizing RGWs, moving most of its functions to the operator infrastructure (using virtualization and private clouds) whilst keeping a drastically simplified device at the customer premises to support the remaining functions that cannot be moved.

The rest of this paper is organized as follows. Section 2 discusses the motivation for virtualized residential gateways. The proposed approach is presented in Section 3 (network architecture) and Section 4 (virtualization of the RGW). Section 5 presents the management architecture for vRGWs, while Section 6 addresses validation. Section 7 discusses related work and Section 8 concludes the paper.

II. THE CASE FOR VIRTUALIZED RGWS

The notion of Virtual Residential Gateways (vRGW), as proposed in this paper, is a natural extension of the trend towards cloud-based services that is gradually broadening its reach towards the operator's network infrastructure.

So far, the physical access network infrastructure has

978-3-901882-50-0

remained relatively excluded from this trend, since many of its components strongly depend on location. Nevertheless, there is space for improvement, namely by decoupling hardware-dependent functionalities (which can not be moved) from software-based functionalities, moving the latter to the data center to improve availability, flexibility and reduce costs. Furthermore, there are some cases where the location constraints are simply related with the "logical location" of the components, not its "physical location". In this case it might be possible to redesign the logical network in order to extend its reach up to the operator data center – thus supporting the virtualization of such components.

For the specific case of the vRGW, what we propose is a mix of both methods: to decouple hardware-based and software-based functionalities as much as possible, and also to extend the logical reach of the customer's "home network" to the data center so that it can include the virtualized RGW. This implies that, as already mentioned, there is a remaining device left at the customer's premises, mainly for simple bridging purposes. Fig. 1 illustrates this approach.

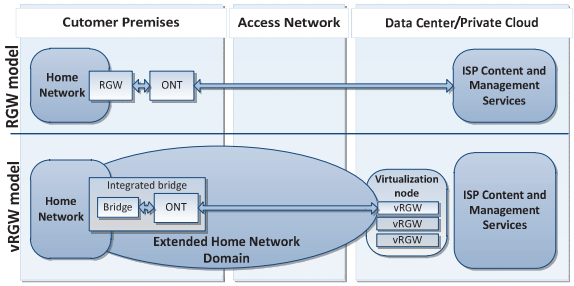

Fig. 1. Classic RGW vs. Virtualized RGW.

It should be mentioned that the proposed approach is not a mere transfer of a handful of services from the (physical) RGW to the operator infrastructure:

- First, in order to effectively overcome the limitations discussed in Section I, the remaining bridging device needs to be drastically simplified, in order to effectively reduce the associated capital expenditure (CAPEX) and operational expenditure (OPEX).
- Second, the access network infrastructure (and the data center) needs to accommodate thousands or millions of logical networks, in order to link the home network of each customer with its vRGW.
- Third, the virtualization technologies at the data center need to efficiently support a large number of vRGWs. Whilst commonplace virtualization platforms can be used, it should be noted that the vRGW is different from the typical virtualized server, demanding less computing resources but more network performance.

The potential benefits of virtualizing the RGW, according to the proposed approach, are manifold:

- The operator would be able to lower CAPEX, since the cost of the remaining device would be much lower, when compared to a full-fledged RGW. Eventually, this device could be merged with the ONT (Optical Network Terminal), further reducing its costs.
- OPEX would also be improved, since the need for on-site maintenance would be reduced (less hardware and misconfiguration problems). Also, running some of the RGW functions in the operator data center can simplify service creation and support, improving the main issues of its high cost, when compared to the low margin of RGWs.
- The operator becomes less dependent from hardware manufacturers (the virtual machine is, by nature, hardware independent). Also, the heterogeneity of RGWs is no longer an issue. The operator can keep a unified image for all virtual machines, using both VM templates and composition for increased flexibility – this also eases and accelerates the introduction of new services.
- The resources consumed by vRGWs can be efficiently managed at the data center (e.g. suspending unused vRGW instances and dynamically adjusting the hosting hardware to effectively necessary resources), resulting in substantial savings of energy and hardware resources.
- vRGW updates and replacement of defective instances (e.g. due to software and misconfiguration problems) are simple VM management operations at the operator data center, reducing downtime for the customer.

Next, a framework for support of virtual RGWs is proposed (Section 3), followed by an analysis of RGW virtualization at the data center (Section 4). Finally, the vRGW management architecture is presented (Section 5).

III. Proposed Network Architecture

Logically extending each customer's home network to the data center is a considerable challenge, due to the inherent scalability and manageability requirements. To the best of our knowledge, there are no proposals specifically addressing such a virtualization context. However, this proposal builds upon two reference frameworks from the Broadband Forum [5] for Ethernet-based broadband aggregation scenarios (even though they were not specifically developed with this objective in mind), namely TR-101 [6], which targets DSL technologies and TR-156 [3] for GPON scenarios.

A. Proposed Architecture for Support of vRGW

Fig. 2 presents the proposed vRGW-enabling architecture.

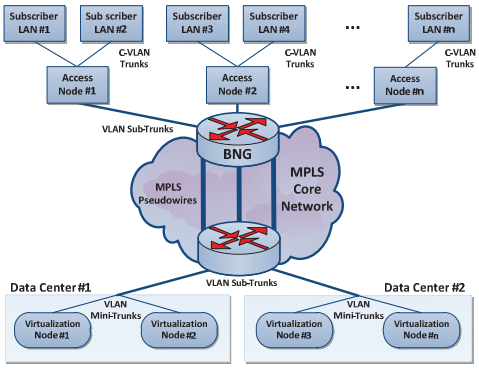

Fig. 2. vRGW-enabling network architecture.

According to this architecture, vRGW instances are hosted by virtualization nodes located at the operator's data centers

(DC#1 and DC#2, in the case of Fig. 2). Each virtualization node is responsible for a set of vRGW instances. Each virtualization node is connected to the network by a mini VLAN trunk representing the I/O network traffic of every individual vRGW instance running in that specific virtualization node (i.e., the corresponding set of Customer-VLANs, or C-VLANs).

Each mini VLAN trunk is aggregated into a larger sub-trunk representing all the C-VLANs on a specific data center. This sub-trunk is then encapsulated into MPLS Pseudowires (PW [7]) at one of the MPLS edge routers of the operator core network. Those PW are then transported to other edge routers, also known as Broadband Network Gateways (BNGs).

At the exiting BNGs, each PW is converted again into VLAN sub-trunks, each of them connected to the corresponding network access node. At each individual access node, each VLAN sub-trunk is divided into smaller C-VLAN trunks carrying the traffic of each individual subscriber (separated by one VLAN for each service being delivered).

At the customer premises, on the ONT, each VLAN is untagged and mapped into a port in one unmanaged bridge with a wireless access point (the ONT and this bridge are not represented in Fig. 2, for sake of simplicity). At this point all the equipment inside the subscriber LAN can connect to the stripped-down bridge and obtain a valid IP address from the DHCP daemon running on the vRGW.

This network architecture allows extending the logical reach of the subscriber LAN to the vRGW hosted at the operator's data center using the technologies already in place. Moreover, the joint use of VLAN stacking, VLAN trunks and MPLS Pseudowires make it possible to handle, aggregate and encapsulate in a scalable manner the high number of VLANs required by thousands or millions of subscribers.

Considering the aforementioned Broadband Forum VLAN Aggregation mechanisms [8], the proposed architecture may fit both 1:1 Customer VLAN (1 VLAN per service, using single tagged or stacked VLANs [9][10]) and hybrid Customer VLAN with Multicast VLAN topologies [11].

IV. Virtualization of the RGW

RGWs typically consist of an embedded device based on specialized derivations of Unix, including services such as NAT routing, DHCP, firewalling, DNS, content filtering, QoS management, VoIP services and application-level gateways. Physical interfaces typically include Ethernet ports (home network, access network), a wireless access point, analog telephony adaptor (ATA) and, in some cases, USB ports.

As already mentioned, those physical interfaces cannot be completely removed, making it necessary to keep a bridging device at the customer premises. This device can be much simpler than a full-fledged RGW, even if providing advanced interfaces such as an 802.11 wireless access point: the complex services and the most demanding network functionality – the interface between the home network and the Internet – are virtualized and moved to the data center. CAPEX and OPEX costs of such a device are expected to be lower, especially if it becomes integrated on the existing ONT.

From a virtualization environment point-of-view, the RGW has humble computing requirements (modest processors and a few hundred Mbytes of RAM), with the possible exception of the network interfaces between the access network and the home network. RGW consolidation by means of virtualization enables to further reduce their computing requirements. Overall, this means that virtualization of RGWs can be performed using off-the-shelf virtualization platforms. Potential compatibility issues (due to the embedded hardware of RGWs) are also minimized by the fact that RGW firmware is usually based on Unix-derived stacks ported to several hardware platforms, including the Intel x86 family.

A. Architectural Optimizations

In the specific scope of this paper, for sake of simplicity, it is assumed that RGW virtualization is achieved by migrating it to the Virtualization Nodes, without extensive rearrangements of its architecture. Each vRGW remains isolated from the other vRGWs sharing the same private cloud.

Nevertheless, the concept of vRGW also opens the possibility of more advanced solutions, where some services previously handled by the RGW can become consolidated outside the boundary of each individual RGW, on a context of co-location or functional distribution (see Fig. 3).

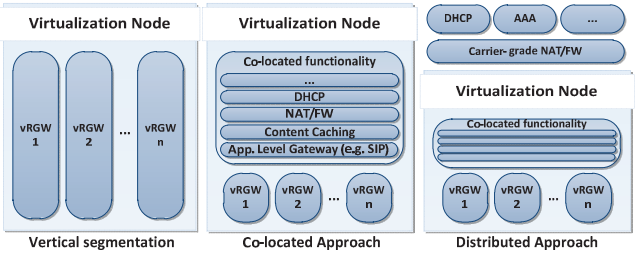

Fig. 3. Architectural Optimizations for vRGWs.

Application-level gateways such as SIP gateways, DNS cache servers and content caching mechanisms, for instance, could benefit from this approach to improve their manageability and functionality, as well as to reduce vRGW footprint. Also, heavy-duty network functionality such as NAT and firewalling could be moved outside the vRGW, being co-located or relocated to carrier-grade mechanisms. Also, DHCP can be moved outside the vRGW, being replaced by a relay agent (implementing DHCP option 82 [12]), located in the access node (i.e., in the OLT) or the vRGW.

Several RGW management functionalities could also be simplified if co-located at the virtualization node level (similarly to the vRGW management architecture proposed in Section V) or moved outside, to the operator infrastructure. In this perspective, the vRGW can eventually be trimmed down to a bare instance with minimal networking and management capabilities, an execution environment and a basic service set, retained for customer privacy or functional reasons.

B. User-driven Customization

The configuration of physical RGWs is typically shared

between the customer (that uses a local web interface to define user-accessible parameters) and the operator (that uses CWMP [13] to remotely configure the other parameters). A similar approach might be followed for vRGWs, allowing the customer to access an operator-provided web interface to define user-accessible parameters of the vRGW (e.g. DHCP pools and 802.11 credentials). The configuration for each customer might be added to its profile and activated upon instantiation of the corresponding vRGW.

C. vRGW Migration

vRGW migration is a valuable feature not only for providing continuous operation, but also to ensure geographic data center proximity when a subscriber moves its location to a different household (specific operator data centers could be chosen accordingly to geographical proximity criteria).

Live or cold migration of vRGWs inside the same data center is very similar to conventional VM migration in LAN environments, assuming the VLAN mini-trunks are mirrored across the virtualization nodes. However, migration across distinct data-centers is a different matter, involving remapping of C-VLANs – an issue out of scope for this paper.

Cold migration across data centers can be context-based (spawning or resuming a pooled vRGW instance on the destination data center) or image-based (stopping and moving the VM instance across data centers, later resuming its operation). Live vRGW migration across data centers without disruption is a complex operation, demanding close cooperation between the network and VM layers. Coordination between the MPLS BNG and edge routers must ensure the correct C-VLAN is encapsulated within a MPLS PW connected to the data center that will host the vRGW.

D. vRGW Pool Management

From the operators' perspective, the possibility of over-provisioning vRGW pools is a logical solution for resource optimization. By using a template-based mechanism with image composition capabilities, the system is able to easily assemble, customize and start pools of stateless vRGW instances, suspending the ones not immediately needed - this ensures a rapid starting time. Pooled vRGW reconfiguration is provided by a specific management component.

V. Management Architecture

Although vRGWs could be managed by common cloud computing management frameworks like OpenStack [14] or Open Nebula [15], the specifics of a network element such as the RGW probably require a different approach. As such, the management mechanisms for the proposed vRGW architecture are based on Broadband Forum's CPE WAN Management Protocol (CWMP [13]), the *de facto* standard for management of devices and services, enabling total management transparency to the operator on both physical and virtual devices, using existing Operations Support Systems (OSS).

A. Proposed Approach

The proposed management architecture is shown on Fig. 4.

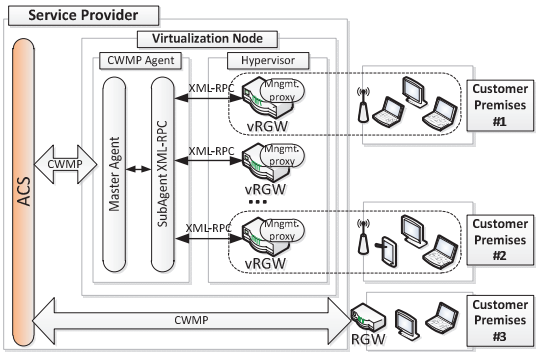

Fig. 4. vRGW management architecture

In order to reduce the vRGW instance overhead as much as possible, the proposed solution departs from the vertical segmentation model (one CWMP management agent per device) used for physical device management, going instead for a solution that remains compatible with the CWMP management paradigm. Each vRGW runs a basic proxy management component that communicates with a full-fledged CWMP management agent hosted at the hypervisor level through a simple XML-RPC [16] interface, thus decoupling the CWMP logic (hosted on the hypervisor) from specific vRGW management mechanisms.

The CWMP agent on the virtualization node is structured on two layers: the first implements CWMP protocol interfaces with the management server (or ACS – Auto Configuration Server, in CWMP terminology), while the second one deals with XML-RPC interfacing with the vRGWs. Both layers use an in-house CWMP modular agent framework [17], designed to decouple specific data model and agent functionality to specific-purpose subagents – a capability implicitly supported by CWMP [18], which allows for embedding the data model of proxied devices within the data model of another device.

VI. Validation

This section discusses a proof-of-concept implementation of the proposed solution, the experimental validation process and obtained results for vRGW performance.

A. Reference Testbed

In order to validate the proposed vRGW approach, we built the proof-of-concept testbed illustrated in Fig. 5, which emulates the data center environment and the home networks.

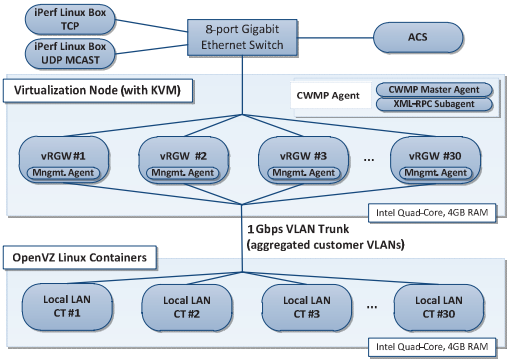

Fig. 5. vRGW virtualization testbed.

The virtualization node was based on an Intel server with 4GB of RAM, running CentOS Linux 6 x86 64 [19] with KVM (Kernel-based Virtual Machine [20]) as the virtualization hypervisor and *libvirt* [21] as management framework. The management server is a CentOS 6 Linux 6 x86 64 system with a java-based CWMP ACS. The KVM virtualization node also hosts the java-based CWMP Agent, incorporating the master agent and the XML-RPC layer for communication with the vRGW proxy management agents.

The interconnection between the virtualization node and the home networks was based on a 1Gbps Full Duplex Ethernet link, supporting one point-to-point VLAN trunk between the virtualization node and another CentOS 6 Linux x86 64 server running up to 30 OpenVZ Linux Containers [22] to emulate up to 30 independent subscribers (one per vRGW instance on the virtualization node). This VLAN Trunk aggregates the multiple C-VLANs involved.

The virtualization node was connected to a Gigabit switch together with the ACS and two other Linux machines (needed for vRGW performance tests, being used as traffic generators) running *iperf* [23], one of them configured as a UDP and UDP Multicast server source (to emulate VoIP and IPTV traffic, respectively) and the other as TCP generator (to emulate Internet connectivity and other general TCP traffic).

B. vRGW Implementation

The presented proof-of-concept prototype uses OpenWrt Backfire x86 RGW instances [24], virtualized using the KVM hypervisor and the *libvirt* management framework. Experimental measurements were collected using *virt-top* [25] for showing stats of virtualized domains.

As for the proxy management agent, the proof-of-concept implementation was done in Python [26], in the form of a XML-RPC server listening to system call requests coming from the CWMP agent on the virtualization node.

C. Evaluation of vRGW Performance

The reference model for the vRGW performance tests is based on the Quality of Experience Requirements for IPTV Services proposed by ITU for "Average Users" and "High Users" [27] (see Table I), which were used to configure the traffic generators from the testbed. Those requirements were used to model *iperf*-generated UDP traffic (IPTV and VoIP services). The "TCP Internet" traffic generator was not capped, using the remaining available bandwidth.

It should be noted that, despite the usage of UDP multicast for IPTV, a worst-case scenario was created by using Customer VLAN (1:1) mapping between the virtualization node and the home networks. This is roughly representative of Customer VLAN with Multicast VLAN scenarios where each customer consumes distinct TV channels.

TABLE I. DOWNSTREAM BITRATES FOR DIGITAL TRIPLE PLAY.

Service	Average User (Mbps)		High User (Mbps)	
Internet		5.0		10.0
Telephony		0.1		0.1
SDTV (MPEG-4)	2 channels	3.0	2 channels	3.0
HDTV (MPEG-4)	1 channel	8.0	2 channels	16.0
Total		**16.1**		**29.1**

To assess the testbed capabilities, measurements were conducted varying the number of vRGWs and considering ITU-defined requirements for "Average Users" and "High Users" except for telephony, in which a requirement of 3 phone calls using the G.711a codec (exceeding the 0.1 Mb/s ITU recommendation) was established, for both profiles.

Fig. 6 presents the average throughput measured for each home network, with 1, 10, 15, 20, 25 and 30 vRGWs hosted by the virtualization node and the "Average User" scenario. Results show the ITU reference requirements to be easily surpassed, even for 30 vRGWs hosted on modest hardware.

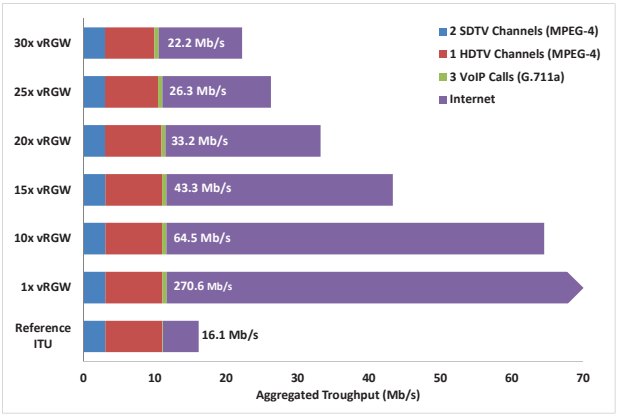

Fig. 6. Measured Throughput per vRGW ("Average User").

The measured average throughput per home network achieved for the "High User" profile is presented in Fig. 7. According to these results, our prototype should be able to support between 20 and 25 "High Users".

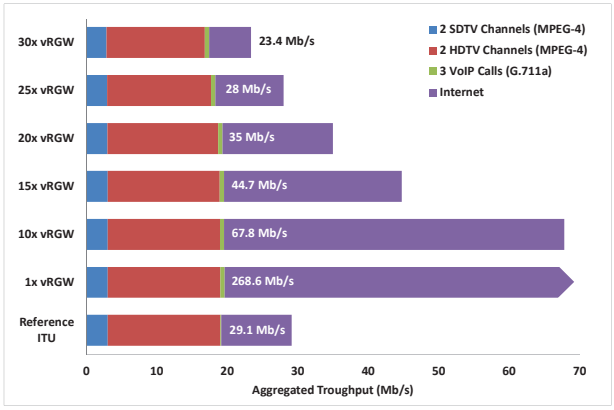

Fig. 7. Measured Throughput per vRGW ("High User").

We noticed a small but noticeable packet loss for the UDP traffic in the most demanding scenarios (many vRGWs). The three worst cases were: 4.30% packet loss (30 vRGW and "High User" profile), 2.35% (25 vRGW, "High User") and 1.57% packet loss (30 vRGW, "Average User"). This can be justified by the fact that we used no traffic prioritization, an unlikely scenario in production environments. Furthermore, such packet loss ratios can be handled by the forward error correction (FEC) mechanisms employed by IPTV platforms without any noticeable degradation of quality [28-29].

During the experimental tests we also monitored the virtualization node, in terms of CPU load and memory usage.

Fig. 8 shows the total CPU load used by the hypervisor of the virtualization node (represented as a percentage of the maximum CPU capacity), for the "High User" scenario. Apart from the single vRGW case, the average CPU usage for 10, 15 and 30 vRGWs under load fits between 75% and 85%. This threshold could be explained by the fact that, besides running the virtualization hypervisor, the virtualization node also needs to handle the management of multiple VLANs with high load, running network I/O traffic on its VLAN trunk.

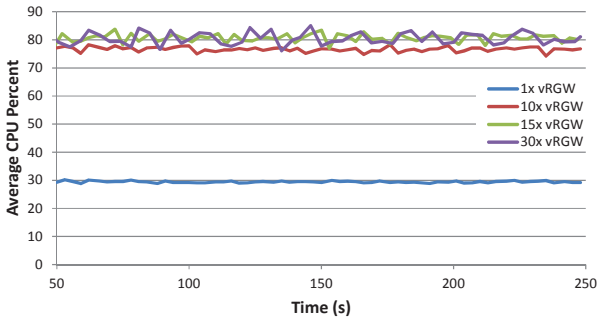

Fig. 8. CPU used by the Hypervisor ("High User").

Fig. 9 presents the average CPU load occupied by each individual vRGW (for the same "High User" scenario), represented as a percentage of the maximum CPU capacity of the virtualization node. Results are consistent with the hypervisor measurements of Fig. 8, showing no signs of remarkable overhead at the level of the hypervisor.

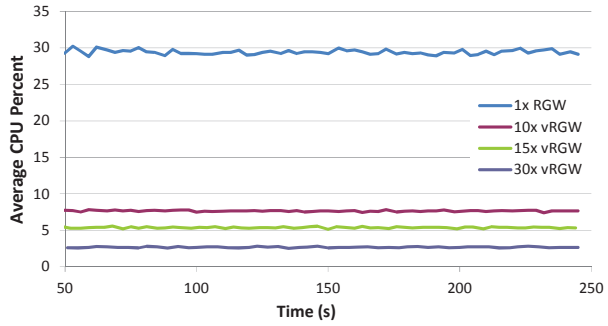

Fig. 9. Average CPU used per vRGW ("High User").

In terms of memory allocation, each vRGW OpenWrt instance was configured with 32MB of RAM, totaling up to 960MB (30x32MB) of allocated memory on the hypervisor. During the tests each vRGW consistently used around 38% of available memory (12MB aproximately), showing that RAM was not a key bottleneck in the specific system we used.

Finally, it should be stressed that the vRGW capacity per virtualization node, as hinted by presented measurements, is constrained by modest hardware (outdated Intel Q8400 CPU, 4GB of RAM, entry-level network interfaces). Production deployments may achieve important capacity improvement by tuning hardware components and using resource consolidation tools provided by carrier-grade virtualization platforms.

VII. Related Work

Regarding router virtualization, [32] and [33] discuss a solution using the Xen Hypervisor [34], focusing on the fair allocation of resources in scenarios with multiple users sharing the same virtualization host. Also, [35] studied virtual router migration scenarios on Xen hosts, with [36] and [37] discussing the virtualization-induced network performance penalty for virtualized network appliances. Finally, [38] proposes a solution for implementing distributed firewalls using virtualized security appliances.

Specifically for mass virtualization of RGWs in broadband access network environments, the Eurescom Project P2055 [39] provides a high-level discussion of three scenarios to remove the physical RGW from the customer's premises:

- Pushing the RGW functionalities to the access nodes. This approach requires massive hardware upgrades of access nodes and fragments computing resources across the operators network – increasing complexity and costs.
- Integrating RGW functionalities on the BNG, keeping the network design unchanged (unlike the first one). However, current BNGs are not expected to support massive RGW virtualization, thus leading again to the need of hardware upgrade and fragmentation of computing resources.
- As a stand-alone network element (NE) located somewhere in the operator's metro network. This option has the advantage of not interfering with already deployed network elements but introduces a new hardware component on the network, with inherent costs and maintenance requirements.

Also, [40] discusses several alternatives to physical RGW replacement, embedding transport capabilities on the access node (OLT) and decoupling AAA, DHCP and NAT functionality. When compared to these alternatives, our proposal stands out for a number of reasons:

- RGW virtualization uses mature virtualization technologies commonly used on data centers and private clouds.
- Moving the vRGW to the data center reduces computing resource fragmentation across the operator's network.
- It implies no changes in existing network setups and can be gradually deployed, resulting in an easier migration path, with coexistence of physical and virtual RGWs.

VIII. Conclusions

In this paper we proposed a framework for virtualization of RGWs, to reduce the operator's CAPEX and OPEX by taking advantage of the recent increase of available bandwidth and advances in consolidation/virtualization technologies.

This framework entails a network architecture that uses already existing technologies to logically extend each customer's home network to the data center. This allows decoupling the functionalities currently supported by physical RGWs, keeping a reduced set at the customer's premises (using a small bridge with minimum requirements) and moving most of them to the operator's data center. In a first approach, those functionalities may be maintained using isolated virtual RGW instances (vertical segmentation). Over time, part of them may be relocated outside the vRGW, being shared among several instances, improving functionality, efficiency and manageability.

ACKNOWLEDGEMENTS

This work was partially funded by PT Inovação (Project Virtuoso) and by Project QREN ICIS (Intelligent Computing in the Internet of Services – CENTRO-07-0224-FEDER-002003).

REFERENCES

[1] IEEE 802.11 Working Group, "IEEE 802.11: Wireless LAN Medium Access Control and Physical Layer (PHY) Specifications", 2007.
[2] B. Cain et al., "Internet Group Management Protocol, Version 3", IETF RFC 3376, October 2002
[3] Broadband Forum, "Using GPON Access in the context of TR-101, TR-156", September 2010.
[4] J. Rosemberg et al., "SIP: Session Initiation Protocol", IETF RFC 3261, June 2002.
[5] Broadband Forum, http://www.broadband-forum.org.
[6] Broadband Forum, "Migration to Ethernet-Based Broadband Aggregation, TR-101 Issue 2", July 2011.
[7] S. Bryant, P. Pate, "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", IETF RFC 3985, March 2005.
[8] G. Young, "Broadband Forum Overview with Focus on Next Generation Access", http://www.uknof.org.uk/uknof14/Young-BroadbandForum.pdf, September 2009.
[9] IEEE 802.1 Working Group, "IEEE Std. 802.1Q-2011, Media Access Control Bridges and Virtual Bridged Local Area Networks", 2011.
[10] IEEE 802.1 Working Group, "IEEE Std. 802.1ad-2005, Virtual Bridged Local Area Networks Amendment 4: Provider Bridges", March 2006.
[11] V. Joseph, S. Mulugu, "Deploying Next Generation Multicast-Enabled Applications: Label Switched Multicast for MPLS VPNs, VPLS, and Wholesale Ethernet", Morgan-Kaufmann, ISBN 0123849233, July 2011.
[12] M. Patrick, "DHCP Relay Agent Information Option", IETF RFC 3046, January 2001
[13] Broadband Forum, "TR-069 - CPE WAN Management Protocol specification v1.1, Amendment 4", July 2011.
[14] OpenStack project, http://www.openstack.org.
[15] Open Nebula project, http://opennebula.org.
[16] Apache XML-RPC Java implementation, http://ws.apache.org/xmlrpc.
[17] T. Cruz et al., "CWMP Extensions for Enhanced Management of Domestic Network Services", Proc. of LCN'2010 (The 35th IEEE Conf. on Local Computer Networks), Denver, USA, September 2010
[18] Broadband Forum, "Data Model Template for TR-069 Enabled Device, TR-106 Amendment 4", February 2010.
[19] CentOS project, http://www.centos.org.
[20] A. Kivity, "KVM, One Year On", Presented at KVM Forum 2007, Tucson, USA August 2007.
[21] Libvirt project, http://libvirt.org.
[22] K. Kolyshin, "Virtualization in Linux", http://download.openvz.org/doc/openvz-intro.pdf, September 2006.
[23] Iperf, http://iperf.sourceforge.net.
[24] OpenWrt project, https://openwrt.org.
[25] Virt-top, http://people.redhat.com/ rjones/virt-top.
[26] Python Programming Language, http://www.python.org.
[27] ITU, "Quality of Experience Requirements for IPTV Services, FG IPTV-DOC-0184", December 2007.
[28] M. Ellis, D. Pezaros, C. Perkins, "Performance Analysis of AL-FEC for RTP-based Streaming Video Traffic to Residential Users", in Proc. of the 19th Int. Packet Video Workshop (PV), Munich, Germany, 2012.
[29] B. Nagel, "Demonstration of TVoIP services in a multimedia broadband enabled access network", MUSE Project Presentation at Broadband Europe 2007, Antwerp, December 2007.
[30] Unix top project, http://www.unixtop.org.
[31] Tcpdump project, http://www.tcpdump.org.
[32] N. Egi et al., "Evaluating Xen for Router Virtualization" In Proc. of the International Workshop on Performance Modeling and Evaluation in Computer and Telecommunication Networks (PMECT) 2007, 2007.
[33] N. Egi et al., "Designing a Platform for Flexible and Performant Virtual Routers on Commodity Hardware", in Proc. of the 16th GI/ITG Workshop on Overlay and Network Virtualization (NVWS), 2009.
[34] P. Barham, B. Dragovic et al., "Xen and the Art of Virtualization", in Proc. of the 19th ACM Symposium on Operating Systems Principles (SOSP) 2003, New York, USA, October 2003.
[35] P. Pisa et al., "Migrating Xen Virtual Routers with No Packet Loss", in Proc. of the First Workshop on Network Virtualizaton and Intelligence For Future Internet - WNetVirt'10, Búzios, Brazil, April 2010.
[36] A. Bazzi, Y. Onozato, "Feasibility Study of Security Virtual Appliances for Personal Computing", in IPSJ Journal of Information Processing, Vol 19, No 0, July 2011.
[37] S. Zeng, Q. Hao, "Network I/O Path Analysis in the Kernel-based Virtual Machine Environment through Tracing", in Proc. of 1st International Conference on Information Science and Engineering (ICISE) 2009, Nanjing, China, December 2009.
[38] D. Basak et al., "Virtualizing networking and security in the cloud" in ACM SIGOPS Operating Systems Review, Vol. 4, Issue 4, 2010.
[39] D. Abgrall, "Virtual Home Gateway: How can Home Gateway virtualization be achieved?", EURESCOM P2055 D1, September 2011.
[40] Da Silva et al., "Home routing gateway virtualization: An overview on the architecture alternatives", Future Network & Mobile Summit, 2011.