An architecture for dynamic QoS management at Layer 2 for DOCSIS access networks using OpenFlow

Alaitz Mendiola*, Victor Fuentes, Jon Matias, Jasone Astorga, Nerea Toledo, Eduardo Jacob, Maider Huarte

Department of Communications Engineering, University of the Basque Country UPV/EHU, Alameda de Urquijo s/n, 48013 Bilbao, Spain

Abstract

Over the last few years, Software-Defined Networking (SDN) has emerged as one of the most disruptive and profitable novelties in networking. SDN was originally conceived to improve performance and reduce costs in Ethernet-based networks and it has been widely adopted in data center and campus networks. Similarly, thanks to the introduction of SDN concepts, access networks will benefit from the higher control, the lower maintenance costs and the better remote access to devices of SDN. However, its application to access networks is not straightforward and imposes great challenges to vendors and network operators, since current SDN technologies are not prepared to handle the provisioning of user equipment, specific port management or QoS requirements of common access networks. Most recent trends dealing with the SDN-ization of access networks advocate for the use of simple devices at the customer premises and the virtualization of the networking functionalities, requiring the provisioning of Layer 2 services in many cases. In such a scenario, this paper presents an architecture that brings SDN to

*Corresponding author
Email address: `alaitz.mendiola@ehu.eus` (Alaitz Mendiola)

common access networks using legacy equipment. In a nutshell, the architecture is based on the abstraction of the access network as a wide area OpenFlow switch where QoS-enabled pipes are dynamically created leveraging the high granularity of the OpenFlow protocol for packet classification. Furthermore, the OpenFlow protocol itself has been extended in order to support the advanced QoS requirements that are common to most access networks. The architecture has been implemented for DOCSIS access networks and it has been validated and evaluated using a real testbed deployed at our laboratory. The obtained results show that the architecture remains compliant with the ITU-T QoS recommendations and that the cost of introducing the elements required by the architecture in terms of service performance is negligible.
Keywords: Software-Defined Networking, OpenFlow, DOCSIS, virtual Customer Premises Equipment.

---

1. Introduction

After the breakthrough of Software-Defined Networking (SDN) [1] and the OpenFlow [2] protocol for its application in campus and data center networks, researchers, innovators and Standard Development Organizations (SDO) have started to work on the application of SDN concepts to access networks. Nevertheless, the adaptation of common access networks to fit under the SDN umbrella is not a trivial question. To begin with, network devices used in current access networks behave like closed boxes, without available interfaces to communicate with them using a SDN protocol. Thus, the hardware is not ready to be controlled by an external entity and does not meet the programmability requirements of SDN. In addition, there are three well known problems when dealing with the SDN-ization of access networks.

The first issue is the complexity inherent to the access networks, with special purpose interfaces for the management and configuration of the physical parameters. As mentioned before, the OpenFlow protocol was initially designed for its application in Ethernet devices, which do not have this kind of interfaces. Thus, the OpenFlow protocol is not prepared to deal with the abstractions and interfaces present at common access networks.

Another problem is related to the Quality of Service (QoS), which plays a key role in the access networks and allows to share the physical resources among the users [3]. In general, the OpenFlow protocol has a very limited QoS support. In fact, not even the most recent version of the OpenFlow protocol is ready to support the provisioning of QoS requirements common to most access networks. For instance, with OpenFlow, it would not be possible to control services in terms of the *maximum burst size* or the *service class*.

Finally, current legacy equipment requires a provisioning phase to connect the different elements that comprise the access network. This phase usually involves the network configuration of the devices located at the user premises and the downloading of a configuration file that determines the behavior of these devices. This is an issue, since no traffic can be exchanged between the access network elements until the provisioning ends, including the corresponding SDN control traffic. Thus, during the provisioning phase, no connectivity services would be provided to the end users, and the control of the network elements located at the user premises would be impossible.

As a consequence, the advancements in the introduction of SDN concepts are starting to occur at a very slow and gradual pace. In order to tackle these questions, many of the initial proposals are based on novel and disruptive architectures where the devices located at the user premises are kept simpler and functionalities are transferred to the operator's network [4]. This is also the approach followed by the Network Enhanced Residential Gateway (NERG) architecture [5] proposed by the Broadband Forum or the Domain 2.0 architecture designed by AT&T [6], where the use of a *simplified and OpenFlow-enabled Customer Premises Equipment (CPE)* is envisioned. Even major manufacturers like Juniper have opted for an architectural approach [7] based on the virtualization of as many network components as possible to improve openness, scalability, modularity and automation within the access networks. Their solutions rely on the concept of a virtual-Customer Premises Equipment (vCPE), a component of a network element that provides a service or network functionality to the end-user, such as IP address assignment or Network Address Translation (NAT). This network functionalities are usually stored in data center and downloaded to the network elements in an automated fashion. As outlined not only by Juniper, but also by some network operators [8], this type of solutions require Layer 2 (L2) visibility to allow the operator to introduce new services without extra hardware requirements.

Some researchers and innovators are also dealing with the introduction of SDN concepts to legacy access networks in an attempt to accelerate the SDN-ization of this type of networks. However, most of the proposals are on a very early stage and are not compliant with the requirements for the introduction of vCPEs. In such a scenario, this paper presents a novel architecture for the dynamic provisioning of L2 services with QoS support in common access networks using OpenFlow. The solution represents an outstanding contribution in the SDN-ization of the access networks, as it is the first attempt to add external control and programmability to the access network with legacy equipment. Furthermore, it facilitates the adoption of the aforementioned vCPE paradigm, which has the approval of the SDOs and the industry.

In a nutshell, the basis of the solution proposed here is to enable and disable QoS-enabled pipes in a dynamic fashion while the entire access network is abstracted as a wide area OpenFlow switch. This abstraction is achieved using a proxy that makes possible to control the entire network using a single OpenFlow controller. The architecture has been designed to be technology agnostic and it can be used to bring SDN concepts to common access networks such as GPON, EPON or DOCSIS [9]. In this paper we propose as an experimental use case deployment of the architecture over a DOCSIS access network.

This paper is structured as follows. Section 2 reviews recent innovations and research proposals in the areas of QoS with OpenFlow and SDN applied to DOCSIS access networks. Next, Section 3 describes the proposed architecture and the elements that are used. There is also an introduction to the novelties regarding QoS in Section 4 and the operational details of pre-provisioning the DOCSIS access network, enabling a new QoS-enabled pipe and disabling an existing one in Section 5. Section 6 presents the performance evaluation, including descriptions about the testbed that has been deployed and the obtained results. Finally, Section 7 summarizes the conclusions.

2. Related Work

As mentioned in Section 1, one of the foundations of future access networks is going to be the virtualization of as many network functionalities as possible, which could be simplified by the adoption of new networking paradigms and the provisioning of L2 services to end-users [8]. Nonetheless, most current access networks are not ready for these new requirements, as they are not prepared for the dynamic provisioning of L2 services. In order to validate and evaluate the performance of the architecture presented in this paper, a DOCSIS access network has been used. As a consequence, the present section reviews the related work about two key research topics: the *QoS in OpenFlow* networks and the *SDN-ization of the DOCSIS access networks*.

Attracted by the benefits of having a logically centralized control plane and the high level of flow granularity achieved in OpenFlow networks, QoS in SDN has become an active research topic that has gained a lot of attention [10–13]. Most solutions are based on custom controllers or applications able to select the best possible path under various circumstances, that is, based on Path Computation Elements (PCE) that leverage the complete knowledge of the network state [14–16]. They rely on the use of queues at the forwarding layer to ensure the QoS, as they allow to prioritize the traffic that pertains to a certain service class at the time of processing it. However, this kind of proposals are not prepared to deal with the special QoS requirements of access networks. For instance, the selection of an optimal path does not make sense on an access network, where all the traffic is forced to pass through the network element that performs the aggregation of the users' traffic.

Regarding implementing QoS in OpenFlow, the protocol provides a minimum set of capabilities, namely the *enqueue* action in all versions of the protocol and the *metering messages* in the most recent ones. These capabilities simply allow the assignment of flows to previously created queues attached to the output ports and the application of rate limiters at the input ports respectively. In many cases, these functionalities are not enough to provide the required QoS. In this particular case, they are not enough to control the QoS at the DOCSIS access network, as they do not provide the necessary tools to guarantee, for example, a minimum sustained rate. This is a well known limitation of the OpenFlow protocol and several solutions propose the extension of the protocol in order to solve it [17]. For instance, Ishimori et al. [18] have extended the OpenFlow protocol to enhance QoS by supporting multiple packet scheduling functions. However, none of the proposed solutions based on the extension of the OpenFlow protocol support the QoS requirements of common access networks, as they are focused on the improvement of different features.

The solution presented in the current paper follows this approach but stands out from the remaining solutions as it extends the OpenFlow protocol making use of the *Experimenter* messages. In our architecture, the experimenter message has been extended to directly create queues at the wide area virtual switch that is exposed to the OpenFlow controller. This is achieved by creating QoS classes at the OpenFlow devices, which unlike typical queues, are not only associated to the outport, but also to the inport, making possible the abstraction of the DOCSIS access network as a virtual switch to the OpenFlow controller.

On the other hand, when talking about the *SDN-ization of the DOCSIS access network*, three different strategies have been widely adopted. The first one is based on SDN enabled Cable Modem Termination Systems (CMTS), and it is the most popular approach among CMTS manufacturers. For instance, Arris [19], which is one of the most important CMTS manufacturers, is working on SDN enabled CMTSs due to the benefits that SDN brings to network operators and service providers. They also consider SDN as the enabler to use standard Ethernet optics and switching from the data center to the node [20]. Nevertheless, OpenFlow-enabled CMTSs are not available yet, they remain as proposals pending to be implemented by the manufacturers. Thus, this approach is not ready to be used due to the lack of commercial products, as most proposals are on a very early stage [21].

The remaining strategies are based on the utilization of legacy CMTSs. One of the possibilities is to have a controller which is aware of the particularities of the CMTS and the DOCSIS Provisioning System (DPS), further explained in Section 3. This is the approach followed by [22], where the Open Daylight Project is extended with a PacketCable MultiMedia (PCMM)[23] plugin able to control the DOCSIS access network by dynamically adding and removing service flows. Still, PCMM only supports the provisioning of L3 service flows. This means that the DOCSIS capabilities, which allow the creation of service flows based on a richer combination of L1 to L4 fields, are underutilized. Furthermore, it does not allow to provide L2 services, making impossible to adopt revolutionary concepts such as the vCPE concept.

In summary, both QoS and SDN in DOCSIS access networks are active research topics. Still, none of the proposals that have been studied so far support the dynamic provisioning of L2 services, either because they are based on PCMM and only support L3 services or because they are theoretical proposals based on SDN-enabled CMTSs that are not implemented yet. In addition, most strategies for the SDN-ization of the access networks require controllers aware of the particularities of the underlying technology, which imposes the adaptation of the controller to be applied with different access network technologies. Besides, regarding the QoS, the OpenFlow protocol provides a basic set of operations as enqueueing and rate limiting that are not sufficient for the QoS requirements of the DOCSIS access networks.

For all these reasons, we propose a novel and disruptive approach that solves all the aforementioned issues. The approach is based on the abstraction of the whole DOCSIS access network as a wide area OpenFlow virtual switch through a proxy connected to an OpenFlow controller. It allows to get the most out of the DOCSIS access network, specially regarding QoS control. For instance, it allows to broaden the packet classification spectra thanks to the high granularity provided by the outer OpenFlow devices. On the other hand, it enables the deployment of services based on L2 parameters. Moreover, it allows to manage the entire network as a single node with any controller able to speak the OpenFlow protocol.

At the time of writing this paper, we are aware of some efforts in this direction. The most relevant one is the architecture proposed by Oliver Solutions [24], which is based on a Hardware Adaptation Layer and the accessFlowNE technology to abstract the access network as an OpenFLow switch. However, being a proprietary solution, there is no information about its design, neither public evidence about its operation and performance.

3. Description of the proposed architecture

To the best of our knowledge, the architecture described here is the first open solution that makes possible the abstraction of common access networks as an OpenFlow virtual switch, making use of open standards and interfaces that allow to control all the OpenFlow and non-OpenFlow devices of the network through the same elements. Furthermore, it defines a new type of QoS-enabled pipe associated to an inport, which can be dynamically enabled or disabled.

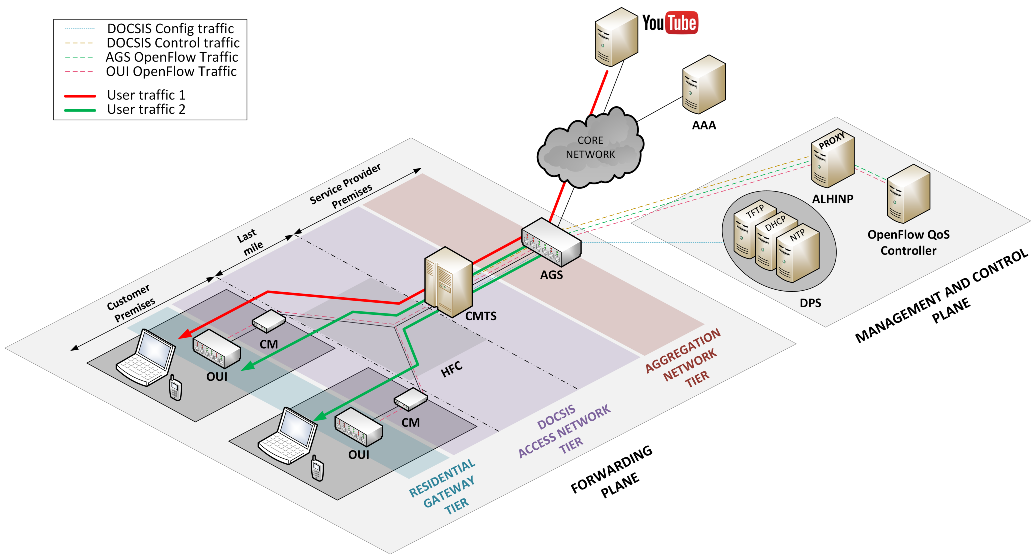


Figure 1: Architecture design

As mentioned before, the architecture can be applied to most access networks, but it has been evaluated using a DOCSIS access network. Thus, in this particular case, the architecture makes possible the dynamic provisioning of services with a certain QoS to the users of a DOCSIS access network after a successful Authentication, Authorization and Accounting (AAA) process. It enhances the dynamic service provisioning of PCMM, which is limited to Layer 3 (L3) services, by making possible the definition of services based on the information of the Layers 1 (L1) to 4 (L4) thanks to the OpenFlow granularity.

Besides the security resources, as shown in Figure 1, two different planes are differentiated within the architecture: the *forwarding plane* and the *management and control plane*.

3.1. Forwarding plane: the three layer approach.

The forwarding plane is where both the OpenFlow switches and the access network elements reside, and it is organized in three different tiers, namely the *Residential Gateway Tier*, the *DOCSIS Access Network Tier* and the *Aggregation Network Tier*, which are further described in the following subsections.

3.1.1. The Residential Gateway Tier

As it happens in most legacy access network technologies, in DOCSIS access networks the traffic can only be characterized by a static subset of L1 to L4 parameters. Nonetheless, OpenFlow is capable of distinguishing flows by means of any combination of packet header fields, ranging from the physical to the application layer of the TCP/IP protocol stack. For instance, OpenFlow supports traffic characterization by means of ARP or MPLS fields, which is not supported in DOCSIS. In such a scenario, the Residential Gateway Layer enhances the traffic characterization capabilities available at DOCSIS access networks because it allows to leverage the granularity of OpenFlow.

In the Residential Gateway Tier, outgoing traffic belonging to a specific QoS-enabled pipe is tagged with a Q-VLAN, which is later used at the DOCSIS Access Network Layer to direct the traffic to a service flow. The Q-VLAN is the first of the two different VLAN tags that are used in our solution. The Q-VLAN univocally identifies the QoS-enabled pipe the packet belongs to, whereas the second VLAN tag (CM-VLAN) identifies the Cable Modem (CM) inside the DOCSIS access network, later explained in Section 4.

The Residential Gateway Tier consists of OpenFlow User Instances (OUI) that extend the OpenFlow capabilities near the end-user. Each OUI is a software element, which can reside in a separate physical device or be embedded into the devices already placed at the user premises. This later method is consistent with the vCPE approach [8], according to which elementary CPEs are used to provide basic L2 connectivity through the access network to end-users. The Residential Gateway Tier is connected to the DOCSIS Access Network Tier through two interfaces: one used for the transmission of OpenFlow control traffic and the other one used for the transmission of user data; additionally, the Residential Gateway Tier is also connected to the actual end-user equipment which acts as source or destination of user data.

3.1.2. The DOCSIS Access Network Tier

According to [25], DOCSIS access networks consist of CPEs, CMs, a CMTS, a Hybrid Fiber Coaxial (HFC) network, a DOCSIS Provisioning System (DPS) and a Network Management System (NMS).

Firstly, the CPEs can be embedded in the CMs or be separate standalone devices connected to the CMs' LAN interfaces. Home routers, set-top devices or personal computers are typical examples of CPEs. Secondly, the CMs adapt the traffic to and from CPEs for its transmission through the HFC network. Meanwhile, the CMTS connects the back office of the operator and the core network with the HFC network. The main function of the CMTS is the forwarding of packets between the two domains that it interconnects, that is, the HFC and the core network.

Additionally, a set of applications that reside in the back office of the service provider make possible the configuration and support of DOCSIS access network elements, which in this architecture are considered part of the Management and Control Plane. On the one hand, the DPS provides the necessary configuration to the CM. It is usually a software suite consisting of a DHCP server for the networking configuration, a NTP server to provide time and date information and a TFTP server where the configuration files are stored and served. On the other hand, the NMS provides all the necessary tools for the management of the network elements and usually includes a SNMP server and a syslog server.

All the elements included in the DOCSIS Access Network Tier are necessary in DOCSIS access networks and are part of the operational processes. For the purpose of this paper, the DOCSIS Access Network Tier will be considered provisioned, which is a slow process that occurs very few times, as the evaluation of the architecture is going to be focused on the operational aspects.

3.1.3. The Aggregation Network Tier

The aim of the Aggregation Network Tier is to gather user traffic coming from the DOCSIS access network and to switch it as needed. This is necessary because in order to work with L2 services, the CMTS works in bridge mode and it is not capable of switching the traffic to and from the HFC as it usually does when it works as a router.

The AGgregation Switch (AGS) is an OpenFlow enabled switch that resides in the Aggregation Network Tier. On the one hand it is charge of switching the traffic to and from the core network, the Internet traffic being the most representative example. On the other hand, it also switches directly the traffic between two CPEs inside the same DOCSIS access network. In this later situation, it does so without involving the core network. Thus, the Aggregation Network Tier is essential to allow the interconnection between users that belong to the same access network as well as between users that belong to different DOCSIS access networks.

3.2. Management and control plane

For the management and control of the forwarding elements two components are used: the Alien Hardware Integration Proxy (ALHINP) and the OpenFlow Controller. Both elements are custom designs and support the extensions to the OpenFlow protocol specifically created for the provisioning of services with QoS requirements, which are later explained in Section 4. Their design and functionalities are described in the following subsections.

3.2.1. ALIEN Hardware Integration Proxy

The ALHINP is the key element of the proposed architecture, since it is the one that enables the management and control of the access network by hiding and making transparent to the OpenFlow controller all the complexity of the non-OpenFlow infrastructure. It is based on the Hardware Abstraction Layer (HAL) defined in the ALIEN FP7 Project [26], which is an abstraction layer that allows the integration of different types of ALIEN devices[1] into an OpenFlow-controlled network.

One of the most interesting features of the ALHINP is that it broadens the functionalities provided by the HAL in order to support the QoS extensions of the OpenFlow protocol for the creation and utilization of QoS-enabled pipes at the abstracted wide area OpenFlow virtual switch. This involves the control of the QoS-enabled pipes at the OpenFlow devices and the creation of the corresponding service flows between the elements of the DOCSIS access network. In summary, the ALHINP provides the necessary mechanisms so that the QoS requirements specified at the OpenFlow controller level are actually implemented in the access

[1]Legacy networking devices without support for the OpenFlow protocol

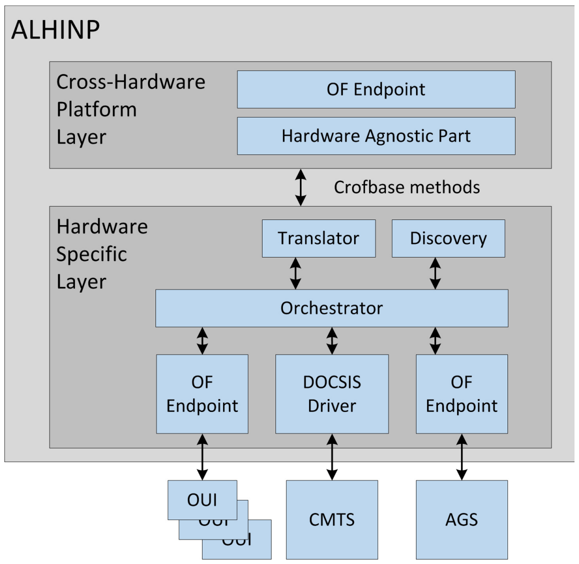


Figure 2: ALHINP High Level Design

network infrastructure to forward the traffic of the end-users with the specified QoS constraints. The ALHINP design is graphically depicted in Figure 2.

It is worth mentioning that the ALHINP appears as an OpenFlow enabled device that abstracts the whole access network to the OpenFlow controller, which is known as the *Big Switch Abstraction*. This is achieved thanks to the *Cross-Hardware Platform Layer (CHPL)*, which is common to all types of ALIEN devices. This layer is the one responsible for providing a virtualized view of the underlying physical devices to the OpenFlow Controller through its *OpenFlow endpoint*. It also contains a *Hardware Agnostic Part (HAP)*, in charge of the communication with the *Hardware Specific Layer (HSL)* through an abstract API. The HAP would remain unchanged even if the ALHINP were applied to other access technologies.

Of uttermost importance is the previously mentioned HSL. This is the layer

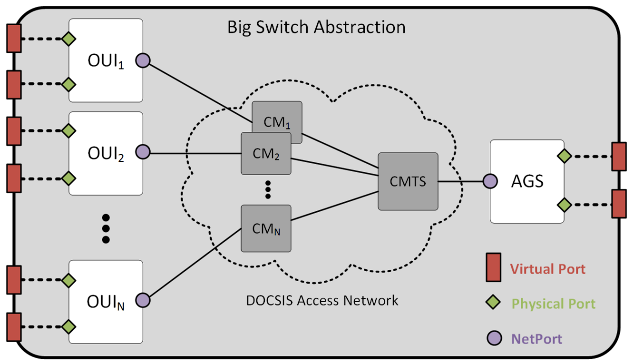


Figure 3: Big Switch Abstraction

that varies depending on the underlying physical devices that are used. In this case, the HSL has been specifically designed for its application with DOCSIS access networks. Thus, it implements the technology and vendor specific features to allow the discovery and configuration of the CMs and the CMTS. The modules that comprise the HSL are introduced below.

In order to abstract the whole access network as an OpenFlow enabled device, it is necessary to abstract the port numbers of the OUIs and the AGS. This is performed at the *Translator* module, which is also in charge of hiding the events that are generated at the internal ports of these very same elements. One of the main functionalities provided by the Translator module is the translation of the physical ports into the virtual ones. In Figure 3 the Big Switch Abstraction is depicted. As it can be seen, the *Physical ports* of the OUIs are either considered *Virtual Ports* if they are edge ports or *Netports* if they are inner ports facing a CM.

It is also necessary to detect all the CMs inside the access network and control the CM-VLAN assignment to each of them. As mentioned before, our solution is based on the utilization of two different VLAN tags, the Q-VLAN and the CM-VLAN. In this case, the CM-VLAN is necessary to identify the CM that each end-user is attached to when working in bridge mode, as it will be further explained in Section 4. Furthermore, the ALHINP must also be aware of the OUIs and be able to virtualize them, that is, to establish the correspondence between the virtual ports and the physical ports. These functionalities are provided by the *Discovery* module.

Both the Discovery and the Translator modules communicate with the *Orchestrator* module, which is the key element inside the HSL (see Figure 2). Furthermore, it is also in charge of the Packet Out message generation, and above all, it is the module that dispatches, splits and translates all the incoming messages. Besides, it is in charge of the retrieval of statistical information from the underlying network devices.

The Orchestrator module is aware of the different forwarding elements that it manages and controls through the *OpenFlow Interfaces* and the *DOCSIS Driver*. These elements are also referred to as the *Device Drivers*. One of the OpenFlow Interfaces acts as the OpenFlow controller of the AGS and inserts the flow entries in it as specified by the *Orchestrator*. Similarly, the OpenFlow Interface in charge of the control of the OUIs does the same process with these forwarding elements, whereas the DOCSIS Driver translates the OpenFlow related information into DOCSIS aware configuration.

3.2.2. OpenFlow QoS Controller

The OpenFlow QoS Controller used in this architecture uses an extended version of the OpenFlow 1.2 protocol with support for QoS management to control the Big Switch Abstraction. These QoS extensions are meant to enable and disable QoS-enabled pipes at the OpenFlow switches, which in conjunction with the existing service flows at the DOCSIS access network conform the QoS-enabled pipes at the Big Switch Abstraction level. Furthermore, when a new service has to be established in the OpenFlow devices, the OpenFlow QoS Controller obtains the QoS-enabled pipes that are already configured in the switches and assigns the service to one of them. If there is no QoS-enabled pipe available, a new one is created and associated with the service.

Even if the last version of the protocol supports some metering and enqueueing functionalities, it would need to be extended in the same way in order to support this novel concept of QoS-enabled pipes. The extensions to the OpenFlow protocol have been implemented to be modular and based on standard mechanisms provided by OpenFlow for the extension of the protocol. As a consequence, the QoS extensions can be used and integrated with existing OpenFlow controllers such as NOX, FloodLight or Open Daylight.

It is worth mentioning that the QoS extensions are mandatory in order to operate the network taking into consideration the QoS constraints. Nevertheless, the Big Switch Abstraction is OpenFlow controller independent, and it can be used with any OpenFlow controller, extended or not with the QoS extensions.

4. Novel QoS functionalities

When referring to DOCSIS access networks it must be taken into account that they rely on QoS criteria to optimize the physical resources. In DOCSIS access networks, *service flows* are unidirectional[2] transport mechanisms between a CM and the CMTS that are characterized by a set of *packet classifiers* to which a certain QoS is applied. Packets are assigned to the service flows if they match the

[2]In DOCSIS, upstream is from CM to CMTS and downstream is from CMTS to CM.

packet classifiers, which can be a combination of fields from L1 to L4 [27].

In common access networks, the bandwidth plays a key role. For that reason, we have selected the bandwidth as the QoS parameter to be managed by the OpenFlow QoS Controller. As a consequence, among the different QoS parameters that can be associated to service flows in DOCSIS access networks, we have decided to expose the *Minimum Reserved Traffic Rate* and the *Maximum Sustained Traffic Rate* to the controller. Nevertheless, the utilization of other QoS parameters that are usually taken into account in access networks such as the *Maximum Burst Rate* could also be exposed to the QoS controller.

Therefore, in order to be able to control the DOCSIS access network through an OpenFlow controller that sees the entire network as a wide area OpenFlow virtual switch, it is necessary to control these QoS features. This involves the mapping of the tree topology of the DOCSIS access network into the Big Switch Abstraction and the control of the service flows and their QoS parameters. Furthermore, thanks to the proposed integration of DOCSIS and OpenFlow, it is not only possible to make use of the QoS aspects of the DOCSIS network using an OpenFlow controller, but it is also possible to enhance the QoS features by providing more granularity at the time of filtering and classifying packets.

This section describes the novelties introduced to make possible the configuration of the DOCSIS QoS capabilities from the OpenFlow controller. First, the concept of *QoS-enabled pipes* at the Big Switch Abstraction has been defined. Second, the OpenFlow protocol itself has been extended using the *Experimenter* messages to support the creation of QoS-enabled pipes at the OpenFlow switches. Third, a new SET_NEW_QOS_PIPE action has been implemented for the FLOW_MOD messages and finally, a QoS resource management algorithm has been implemented in the ALHINP.

4.1. QoS-enabled pipes and their properties

In OpenFlow, QoS is supported by queues that are associated to outports that establish a processing priority. However, in the proposed architecture, QoS is supported using QoS-enabled pipes that are not only associated to an outport but also to an inport. This fact makes possible to abstract the DOCSIS access network as a wide area virtual switch. The ALHINP stores information about these QoS-enabled pipes, and it also stores information related to the QoS parameters that are later used in the service flows of the DOCSIS access network. The structure of the QoS-enabled pipes has been defined extending the existing queue structure with the additional properties listed below.

- *OFPQT_MIN_RATE:* represents the minimum rate configured for a DOCSIS service flow expressed in KB/s. This property has been reused from the existing OpenFlow queue structure.

- *OFPQT_MAX_RATE:* represents the maximum sustained rate configured for a DOCSIS service flow expressed in KB/s. This property has also been reused from the existing OpenFlow queue structure.

- *OFPQT_SRC_PORT:* represents the incoming port from where the QoS can be provided. This property is entirely new.

Besides, both in upstream and downstream, the packets belonging to a QoS-enabled pipe must be mapped into a service flow when they reach the DOCSIS access network. As a consequence, a relation between the OpenFlow QoS-enabled pipe identifier and a way to classify traffic in the DOCSIS access network must

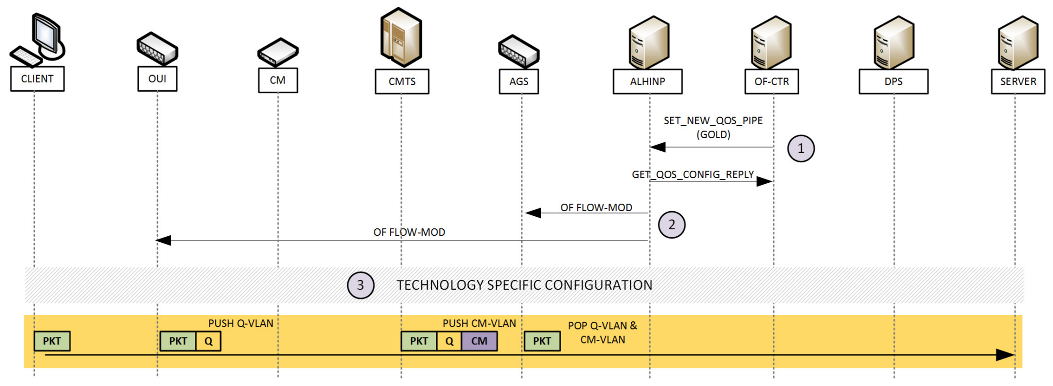


Figure 4: Creation of a new QoS class and VLAN PUSH/POP operations

exist. In this case, it is going to be achieved based on the Q-VLAN tags. That is, the packets that are processed by the QoS-enabled pipe XY are tagged with the VLAN_VID XY so that the CM can transmit them using the corresponding service flow. This feature enhances the packet classification properties of the DOCSIS access network, as packets are going to be filtered in the OpenFlow devices taking advantage of all the granularity of the OpenFlow technology.

4.2. New OpenFlow message: SET_NEW_QOS_PIPE

As mentioned before, the creation of a QoS-enabled pipe on the Big Switch Abstraction implies the creation of a service flow in the DOCSIS access network and the programming of the OpenFlow devices to tag and untag the packets with their corresponding Q-VLAN tag. In such a scenario, the OpenFlow protocol has been extended with the SET_NEW_QOS_PIPE experimenter message, which allows to set a new QoS schema. The complete message exchange for the creation of a new QoS-enabled pipe involving control over the OpenFlow and non-OpenFlow devices is depicted in Figure 4, so as the VLAN PUSH/POP operations.

The SET_NEW_QOS_PIPE message is an asynchronous message sent by the controller and processed by the ALHINP, which in return responds with a message of the type GET_QOS_CONFIG_REPLY message that contains the information and status about the QoS or with a OFPET_QUEUE_OP_FAILED in case of error (1). Once the ALHINP receives a message of this type, the OUIs and the AGS must be programmed to map each flow to a certain service flow in the DOCSIS access network. This is achieved by tagging the packets with a certain Q-VLAN at the OpenFlow devices. These Q-VLANs are stored in the ALHINP so that it maps the different QoS-enabled pipes and their capabilities to the ports at the OUIs and the AGS (2). Regarding the access network devices, the ALHINP informs the DPS about the new set of service flows, packet classifiers and QoS parameters with a new configuration file, which is sent to the DPS to be downloaded by the corresponding CM. It must be taken into account that there is always a pair of default upstream and downstream service flows between each CM and the CMTS, as stated by the standard (3). This default configuration is always present in the configuration file stored at the ALHINP, so as the default behavior to be installed at the OpenFlow devices. The custom QoS behavior is defined by the controller.

4.3. New OpenFlow action: SET_QOS_PIPE

Once that the different QoS-enabled pipes are configured in the ALHINP, when a FLOW_MOD message is sent from the controller an additional action type is used to associate the flow to a certain QoS-enabled pipe. The new SET_QOS_PIPE action type holds the classifier of the QoS class that will be used for that type of traffic and it is sent together with the remaining actions in the *actions field* of the FLOW_MOD message. The QoS-enabled pipe action is interpreted at the ALHINP as the associated Q-VLAN tag that is later used as a SET_VLAN_VID in the FLOW_MOD messages that are actually sent to the ingress/egress elements:

the OUIs and the AGS.

Thanks to the classification enhancement, this feature allows to take advantage of all the granularity provided by OpenFlow in the DOCSIS access network. In addition, it can be used with most Ethernet networking devices as it only requires to support VLAN push and pop actions.

4.4. QoS resource management process

As previously mentioned, the QoS controller is aware of two QoS parameters, namely the *Minimum Reserved Traffic Rate* and the *Maximum Sustaining Traffic Rate*. In order to keep the QoS management simple to the network administrators, the remaining QoS parameters that can be controlled in the DOCSIS access network are handled transparently by the ALHINP without being exposed to the OpenFlow QoS Controller. This is achieved by holding the remaining QoS configuration information in a datastore. Depending on the values in the SET_NEW_QOS_PIPE message and the SET_QOS_PIPE action field, the ALHINP constructs the new configuration file to be sent to the DPS. The generated file is a combination of a fixed part and a variable part. On the one hand, the fixed part is generated including basic information such as the maximum number of CPEs allowed, the SNMP MIB Object that configures the CM in bridge mode and the primary and OpenFlow service flows (later explained in Section 5). On the other hand, the variable part of the configuration file includes information related to the service flows of the QoS enabled pipes and their packet classifiers, which in the current implementation are the *Traffic Priority*, the *Maximum Traffic Burst* and the *Maximum Concatenated Burst*. Furthermore, in the case of the upstream service flows, the Class of Service (CoS) is specified with the Service Flow Scheduling Type and additional parameters such as the *Nominal Polling Interval*

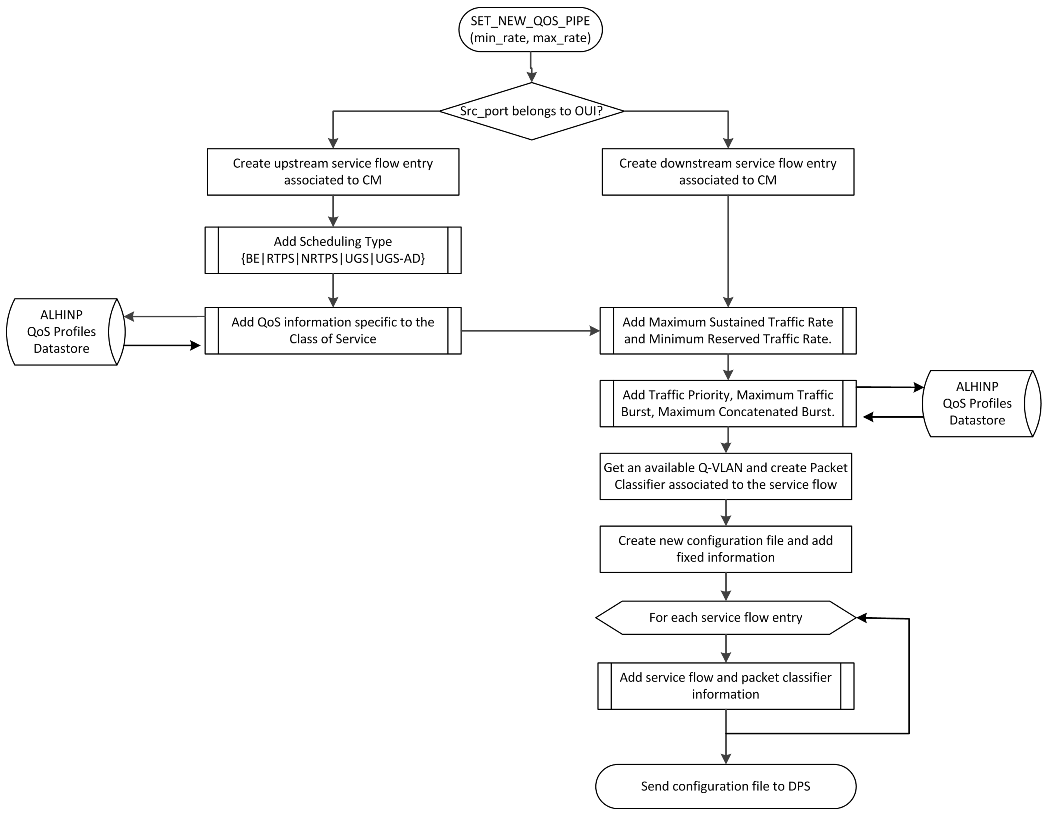

SET_NEW_QOS_PIPE (min_rate, max_rate)
Src_port belongs to OUI?
Create upstream service flow entry associated to CM
Create downstream service flow entry associated to CM
Add Scheduling Type {BE|RTPS|NRTPS|UGS|UGS-AD}
ALHINP QoS Profiles Datastore
Add QoS information specific to the Class of Service
Add Maximum Sustained Traffic Rate and Minimum Reserved Traffic Rate.
Add Traffic Priority, Maximum Traffic Burst, Maximum Concatenated Burst.
ALHINP QoS Profiles Datastore
Get an available Q-VLAN and create Packet Classifier associated to the service flow
Create new configuration file and add fixed information
For each service flow entry
Add service flow and packet classifier information
Send configuration file to DPS

Figure 5: QoS resource management algorithm

in the case of Real Time Polling Service CoS, or the *Unsolicited Grant Size* in the case of an Unsolicited Grant Service CoS are included. It is worth mentioning that the values specified for these parameters are fixed. The complete QoS resource management algorithm is shown in Figure 5.

5. Operational details

This section introduces the operational details of three key stages of the network operation. First, the pre-provisioning of the DOCSIS access network is described. Then, it presents the operational details of requesting a service with a new QoS-enabled pipe. Finally, it also introduces the operational details when a QoS-enabled pipe is no longer used.

5.1. Pre-provisioning

In DOCSIS access networks, each CM must go through a provisioning process to obtain its configuration file. It is assumed that a basic connectivity exists between the CM and the CMTS, which corresponds to the default pair of upstream and downstream service flows (1). Besides, the configuration file loaded in the CM contains 3 pairs of bidirectional secondary services flows (2), as depicted in Figure 6, which are specific to our use case. The first pair of service flows (a) is referred to the OpenFlow service flow, as it is used for the in-band OpenFlow traffic of the OUIs that transports the traffic to and from the IP address of the ALHINP. The second pair of service flows (b) is for the Silver service, which has been configured for TCP, UDP and ICMP traffic with 500 KB/s in both directions. Finally, a third pair of service flows (c) for the Gold service exists, configured with a higher priority for UDP traffic with 20 Mbps for the downstream and 2.5 Mbps for the upstream. In addition, the ALHINP stores all the relevant information about the two QoS classes, Silver and Gold, for future operations.

Further, the OpenFlow devices are also configured in the boot up process (3). In a nutshell, the OUIs' and the AGS' dataplane contains two flow tables. One of them is automatically configured to strip the VLAN tags and perform several actions related to the abstraction of the network. The other flow table is programmed when new services and new QoS-enabled pipes are enabled or disabled. In conclusion, the pre-provisioning involves all the necessary steps to get the DOCSIS access network ready to provide services with the required QoS-enabled pipes.

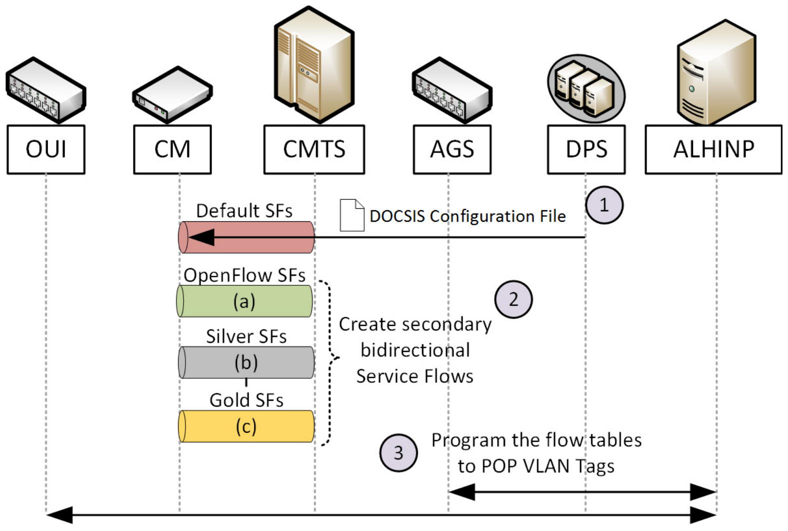

OUI
CM
CMTS
AGS
DPS
ALHINP
1
Default SFs
DOCSIS Configuration File
OpenFlow SFs
(a)
2
Create secondary bidirectional Service Flows
Silver SFs
(b)
Gold SFs
(c)
3
Program the flow tables to POP VLAN Tags

Figure 6: Pre-provisioning of the DOCSIS access network and the OpenFlow devices

5.2. *Service activation*

In this architecture, service activation only happens after an enhanced AAA process univocally associated to the service (1), which is further explain in [28]. Taking into consideration that the services are already provisioned in the DOCSIS access network, when a new service activation is requested (2) the ALHINP informs the OpenFlow QoS Controller that the QoS-enabled pipe already exists (3). Then, the OpenFlow QoS Controller proceeds with the following step (4), which involves sending the FLOW_MOD message with the SET_QOS_PIPE action, as seen in Figure 7.

When the FLOW_MOD message is processed by the ALHINP it generates the corresponding FLOW_MOD messages to be sent to the OUI and the AGS. In the case of an upstream flow, the Q-VLAN tag is pushed into the packet at the OUI, which involves the SET_QOS_PIPE action to be translated into common PUSH_VLAN and SET_VLAN_VID OpenFlow actions. Additionally, the

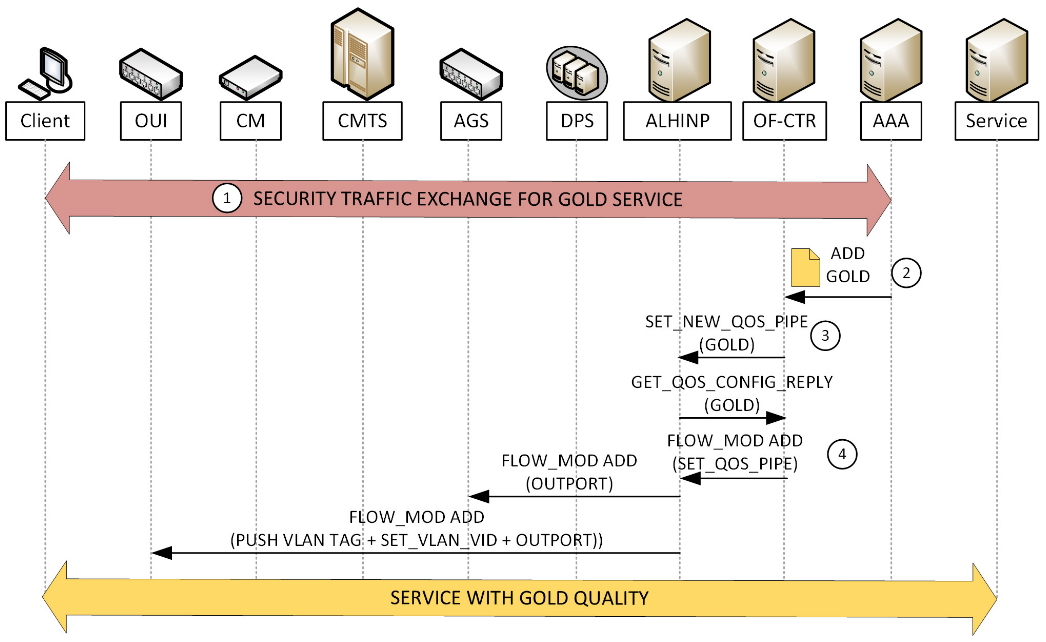

Client
OUI
CM
CMTS
AGS
DPS
ALHINP
OF-CTR
AAA
Service
1 SECURITY TRAFFIC EXCHANGE FOR GOLD SERVICE
ADD GOLD
2
SET_NEW_QOS_PIPE (GOLD)
3
GET_QOS_CONFIG_REPLY (GOLD)
FLOW_MOD ADD (SET_QOS_PIPE)
4
FLOW_MOD ADD (OUTPORT)
FLOW_MOD ADD (PUSH VLAN TAG + SET_VLAN_VID + OUTPORT))
SERVICE WITH GOLD QUALITY

Figure 7: Service activation

Q-VLAN tag used by DOCSIS equipment to map the traffic into the corresponding service flow is removed from the packet at the AGS. On the contrary, in the case of a downstream flow the Q-VLAN tag is pushed into the packet at the AGS, whereas the OUI removes it. In summary, in both cases the traffic is tagged with the Q-VLAN for its transmission through the DOCSIS access network and the Q-VLAN tag is removed at the edge devices.

At this point, once the service activation process ends, the user is able to obtain a service defined by a certain subset of L1 to L4 classifiers with a guaranteed QoS characterized by a minimum and a maximum traffic rate.

5.3. Service deactivation

Similarly, when a service is requested for deactivation, a FLOW_MOD message is sent from the OpenFlow controller to the ALHINP specifying the match

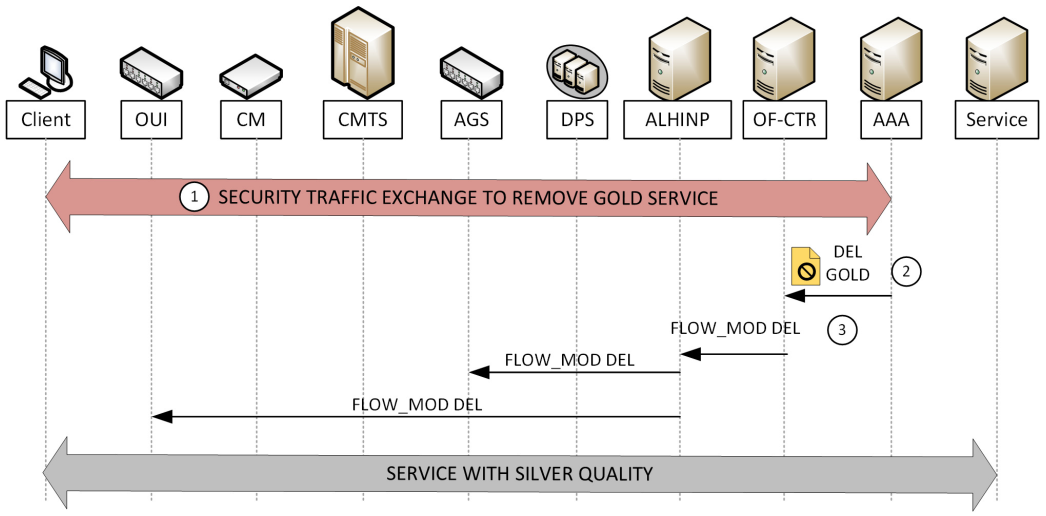


Figure 8: Service deactivation

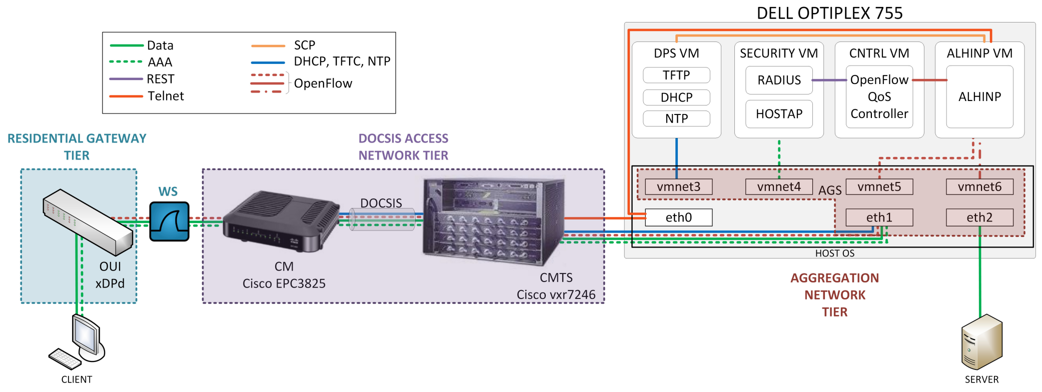


Figure 9: Testbed deployed at the I2T Research Group laboratory

fields of the flow to be removed. Then, the ALHINP translates the FLOW_MOD to resend it to the OUI and the AGS, as it is depicted in Figure 8.

6. Performance evaluation

In order to evaluate that the QoS requirements are fulfilled in the proposed architecture, two different experiments have been conducted using a testbed deployed at the EHU OpenFlow Enabled Facility (EHU-OEF) [29] available at the I2T Research Group laboratory [30], which is depicted in Figure 9. First, the times introduced by the authentication and deauthentication processes used to enable and disable a QoS-enabled pipe have been measured. These metrics are going to be referred to as *authentication time (T_A)* and *deauthentication time (T_D)*. Second, the *latency* and the *throughput* of the proposed architecture have been measured in a normal operation. These metrics have been selected because we consider that they provide meaningful information to demonstrate that the QoS requirements specified for a given service are fulfilled. More specifically, the throughput measurements have been useful to probe that the packets are being classified properly and transmitted trough the corresponding service flows and that the bandwidth limitation is working properly. Besides, the tests have shown that the additional elements of the ALHINP architecture do not impose significant penalties in the network performance. This section presents the deployed testbed, the metrics that have been used and the obtained results.

6.1. Testbed description

The testbed that has been used consists of a client workstation with a Linux Mint 17 where a custom wpa_supplicant with support for service based authentication has been installed. It is connected to the OUI, implemented in a x86_64 machine with 4 GB of memory and running the xDPd 0.5 [31] with the ROFL libraries [32] on top of a Linux Mint 17. Please note that future commercial OUIs will be more lightweight embedded devices, even if a PC has been used as prototype. The OUI works with OpenFlow 1.2 and performs packet forwarding using two different flow tables. In the first flow table, two flow entries are proactively installed by the ALHINP; the first flow entry is used to direct the incoming traffic to the second flow table whereas the second flow entry is used to forward the traffic that comes from the DOCSIS access network back to the client without VLAN tags. A *dag* traffic capturer with Wireshark installed has been located behind the OUI for time measuring purposes.

Regarding the DOCSIS access network, it consists of a Cisco EPC3825 CM and a Cisco VXR7246 CMTS configured to run in bridge mode. This is achieved with a special firmware at the CM that disables the WIFI module and by selecting the bridge mode in the configuration file. The CMTS also needs a special configuration to work in bridge mode, which is achieved by associating each CM to a VLAN (CM-VLAN). Both elements are connected through a coaxial network and the DOCSIS 3.0 communication standard.

Finally, a DELL OptiPlex 755 with 4 physical interfaces and a VMWare Workstation has been used to implement the AGS, the management and control plane and the security resources. As in the case of the OUI, the AGS has been implemented using the xDPd 0.5 to work with OpenFlow 1.2. It also contains two flow tables, where the first one is dedicated to the control traffic itself and the second one is meant to handle the data traffic. When a new QoS class is requested, the FLOW_MOD messages that the OpenFlow controller sends to the AGS update the second flow table.

Additionally, there are four VMs running on top of the VMWare hypervisor to host the ALHINP, the OpenFlow QoS Controller, the DPS and the security resources, which are described below:

- *DPS VM:* an 8 GB Linux Mint 17 VM with 512 MB of memory. It contains the Incognito Broadband Command Center to orchestrate the TFTP, DHCP and NTP services necessary to configure the CMs in the DOCSIS access network.

- *Security Resources VM:* an 8 GB Debian 6 VM with 512 MB of memory. It runs a modified HostAP able to authenticate specific services for a user and a Radius server. The Radius server uses a REST interface to communicate the service profile to the OpenFlow QoS Controller when a successful authentication and authorization process occurs.

- *OpenFlow QoS Controller VM:* an 8 GB Linux Mint 17 VM with 512 MB of memory and a modified NOX controller extended with the SET_NEW_QOS_PIPE message and the SET_QOS_PIPE action. It has an interface to communicate with the ALHINP using OpenFlow.

- *ALHINP VM:* an 8 GB Linux Mint 17 VM with 512 MB of memory with the ALHINP installed in it. The ALHINP is the element that receives the OpenFlow traffic generated by the AGS and the OUIs. It is able to distinguish the traffic of each device because it is connected to the AGS through two different interfaces. It is also connected to the DPS and uses the SCP protocol to send the new configuration files when a new QoS is to be applied at the DOCSIS access network. Further, it is also connected to the CMTS and uses Telnet to trigger the provisioning process of the CMs.

For the latency and the throughput measurement, the client and the server have been replaced by a SmartBit 600B device from SPIRENT, which is a traffic generator and high-accuracy network performance analysis system that implements the RFC 2544 [33]. As depicted in Figure 10, the SmartBit 600B acts both, as traffic generator and traffic consumer thanks to its two interfaces. In this setup,

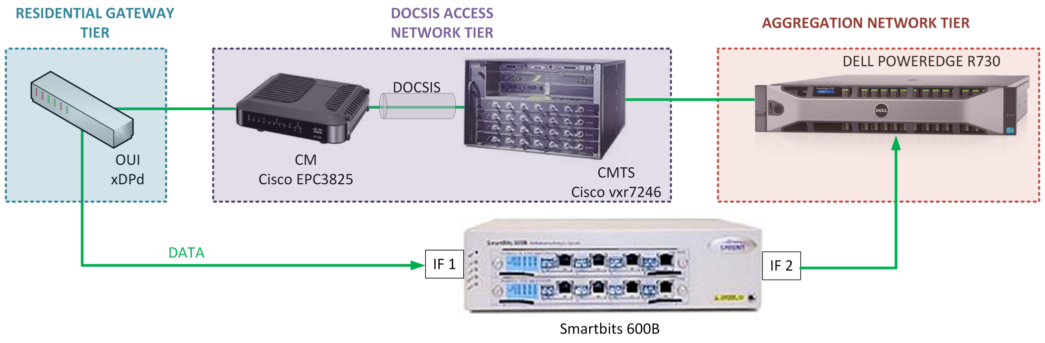


Figure 10: Testbed with SmartBit 600B for latency and throughput measurement

the AGS runs in a server with eight Gigabit Ethernet interfaces, an Intel Xeon 2.6 GHz processor with 8 cores and 32 GB of RAM.

6.2. *Selected metrics*

As mentioned before, for the performance evaluation of the proposed architecture two experiments have been conducted. They have provided the key metrics in order to asses the correct performance of the platform, which are described in the following subsections.

6.2.1. *Authentication and deauthentication times*

Regarding the evaluation of the overhead introduced by the AAA processes, two metrics have been used. The first one is the authentication time, which is the time necessary to enable a service associated to a new QoS-enabled pipe in the entire setup, whereas the second one measures the deauthentication time that takes to stop using that QoS-enabled pipe. In this case, the DOCSIS access network has been pre-provisioned with two different QoS-enabled pipes: Gold and Silver, as described in Section 5. For this configuration, T_A refers to the period of time from the moment that the EAPOL_Start message is sent by the wpa_supplicant installed

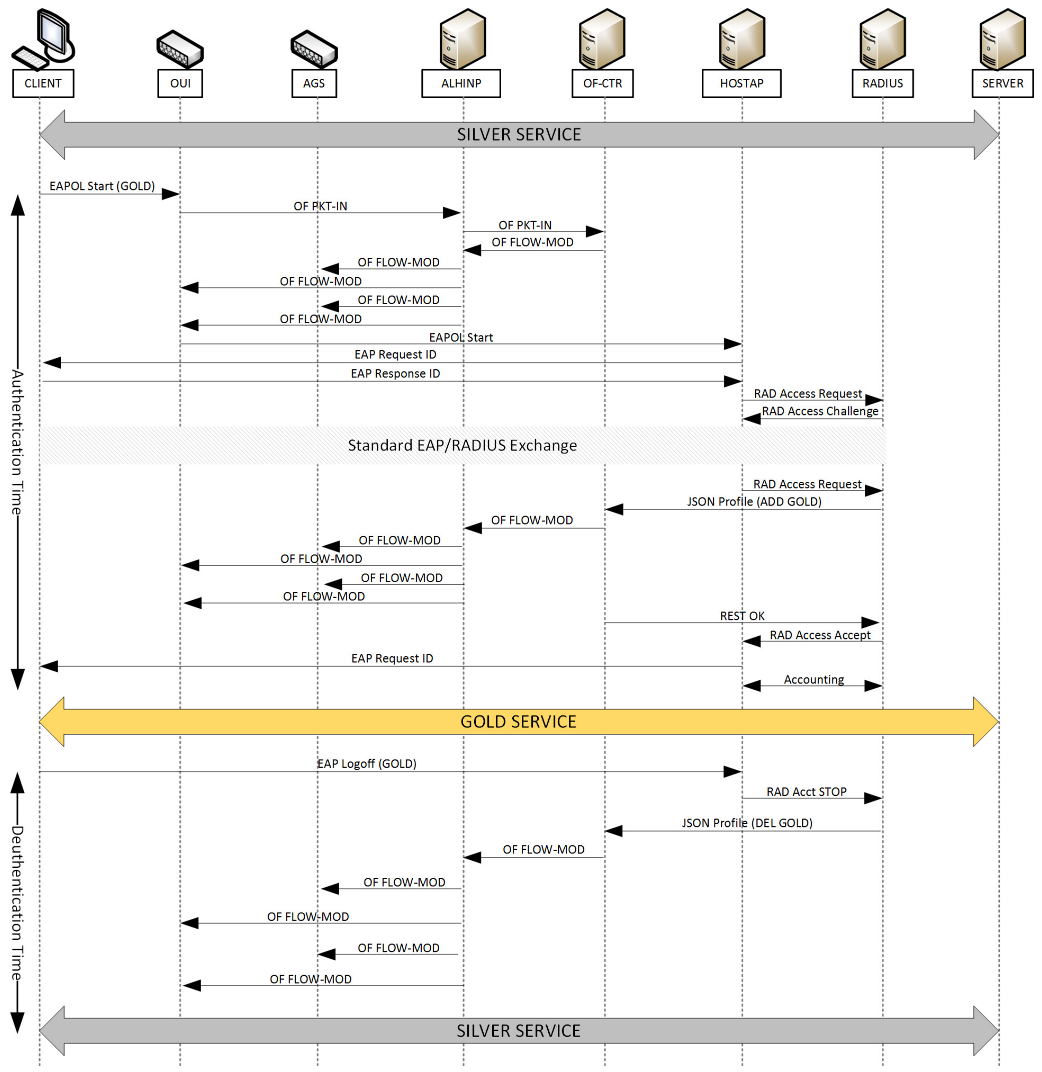
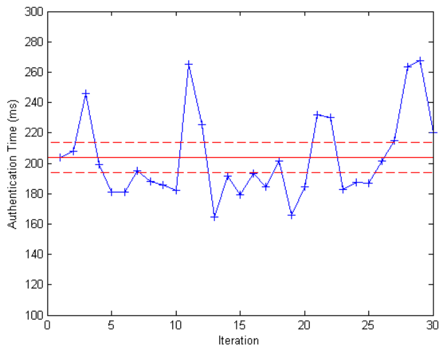


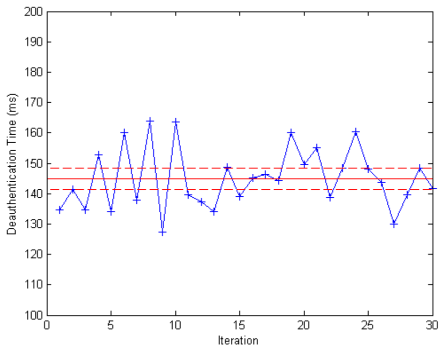
Figure 11: Message exchange during T_A and T_D

in the client to request the Gold QoS-enabled pipe until the service becomes available. The message exchange that takes place during T_A can be seen in Figure 11. It also shows the message exchange from the moment that the user requests to leave the Gold QoS-enabled pipe by sending the EAP_Logoff message from the wpa_supplicant until it starts using automatically the Silver QoS class, that is, T_D.

Both delays, T_A and T_D have been measured 30 times. As a consequence, according to the Central Limit Theorem, it can be considered that the metric follows a normal distribution.



(a) T_A



(b) T_D

Figure 12: Authentication and deauthentication measurement results

Metric	**Average**	**Standard Deviation**	**95% Confidence interval**	
			Lower limit	**Upper limit**
T_A	203.752 ms	28.339 ms	193.61 ms	213.893 ms
T_D	144.948 ms	9.966 ms	141.381 ms	148.514 ms

Table 1: Values of T_A and T_D

Figure 12a depicts the results obtained in each of the 30 measurements for T_A, which is 203.752 ms average, with a standard deviation of 28.339 ms. Similarly, the 30 values obtained for the T_D are shown in Figure 12b. In this case, the T_D is 144.948 ms with a standard deviation of 9.966 ms. Both figures also include the obtained average value and the lower and upper limits of the 95% confidence interval.

As it can be seen in Table 1, which summarizes the results obtained for both metrics, T_A is higher than T_D. This is a direct consequence of the amount of packets that must be processed during T_A and T_D, which is higher in the case of the authentication. Furthermore, due to the reactive way in which the ALHINP based architecture operates, during the authentication the ALHINP and the OpenFlow QoS Controller must learn about the service requested, which is not the case during the deauthentication phase. Despite the differences between the two metrics, both of them meet the performance requirements of the highly demanding real-time IP services indicated in the ITU-T Y.1541(2011).

6.2.2. Latency and throughput

The two other metrics used for the performance evaluation are the latency and the throughput, which have been obtained by means of a SmartBit 600B. The performance evaluation has been conducted in conformance with the RFC 2544 [33]. Thus, both metrics have been measured using L4 packets of different frame sizes: 64, 128, 256, 512, 1024, 1280 and 1518 bytes. We have decided to leave out of our performance evaluation the values obtained with 1518 bytes frames because they have been affected by the fragmentation and therefore, are not comparable to the rest of the measurements that have not been affected by it. Fragmentation for this frame size occurs at the AGS and it is the consequence of the two VLAN tags introduced by the solution, which makes the frame size to exceed the maximum Ethernet frame size. On the one hand, the throughput has been measured 10 times for each frame size over a 10 seconds interval, following the recommendation. On the other hand, the latency measurement has been repeated 20 times, using a maximum throughput flow over 120 seconds, as specified by the recommendation.

The RFC 1242 [34] defines the metrics in use as follows:

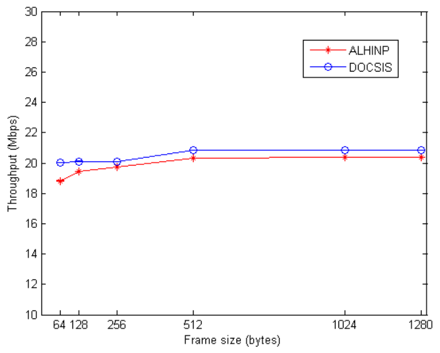
- Throughput: maximum rate at which none of the offered frames are dropped by the device.

- Latency: time interval starting when the end of the first bit of the input frame reaches the input port and ending when the start of the first bit of the output frame is seen on the output port

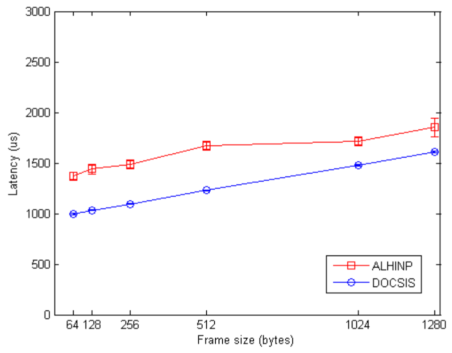
Taking into consideration that the same SmartBit 600B has been used for traffic generation and consumption, no synchronization tools are required to obtain precise and reliable measurements. In the case of the latency measurement, the input port is the Interface IF1 of the SmartBit 600B and the output port is the Interface IF2 of the SmartBit 600B, as we are measuring the latency of the downstream channel.

As mentioned before, the latency and the throughput have been measured in conformance with the RFCs 2544 and 1242 using the SmartBit 600B. Figure 13a depicts the average throughput obtained, in bytes per second, for each one of the frame sizes in the case of a pure DOCSIS access network and with the ALHINP. As it can be seen, for frame sizes equal or higher than 256 bytes the values obtained are approximately a 2% lower, thus, the penalization of using the ALHINP proxy is negligible. For smaller frame sizes, the cost of introducing the ALHINP is minimal, between 3.23% and 6.08%. This behavior is the expected one, as the amount of packets processed per second decreases and the impact of having additional elements with their corresponding processing delays is minimized. Table 2 summarizes the results[3].

[3]A null standard deviation for the throughput is the result of the methodology followed by the SmartBit 600B, which uses pre-defined rates and selects the values that result in the same number



(a) Throughput



(b) Latency

Figure 13: Average throughput and latency for DOCSIS and ALHINP

Frame size (bytes)	**ALHINP**				**DOCSIS**			
	Average	**Std. Deviation**	**95% Confidence interval**		**Average**	**Std. Deviation**	**95% Confidence interval**	
			Lower Limit	**Upper Limit**			**Lower Limit**	**Upper Limit**
64	18.82	0.03	18.80	18.84	20.04	0.03	20.02	20.06
128	19.46	0.02	19.44	19.47	20.11	0.03	20.09	20.12
256	19.75	0	19.75	19.75	20.08	0	20.08	20.08
512	20.33	0	20.33	20.33	20.83	0	20.83	20.83
1024	20.41	0	20.41	20.41	20.83	0	20.83	20.83
1280	20.40	0	20.40	20.40	20.84	0	20.84	20.84

Table 2: Throughput comparisson in Mbps

Frame size (bytes)	**ALHINP**				**DOCSIS**			
	Average	**Std. Deviation**	**95% Confidence interval**		**Average**	**Std. Deviation**	**95% Confidence interval**	
			Lower Limit	**Upper Limit**			**Lower Limit**	**Upper Limit**
64	1368.9	113.9	1328.1	1409.6	996	13.6	990.1	1002
128	1439.5	130	1393	1486	1032.5	13.1	1026.8	1038.3
256	1488	130.7	1441.2	1534.8	1096.4	13.2	1090.6	1102.2
512	1670	116.4	1628.3	1711.6	1229.5	11.1	1224.6	1234.3
1024	1714.2	113.8	1673.5	1755	1480.6	17.7	1472.8	1488.3
1280	1851.7	1927.8	2035.7	2000.4	1609.3	12.6	1603.7	1614.8

Table 3: Latency comparisson in μs

of received packets for a given rate

Since we are using UDP traffic in the downstrean channel, this test is also valid to demonstrate that the QoS requirements are being fulfilled. The configuration file used for this test is the same one explained in Section 5, with a higher priority downstream service flow for the UDP traffic limited to 20 Mbps. As shown in Table 2, the throughput obtained for all frame sizes is approximately 20 Mbps. Therefore, we can conclude that the packet classification in the DOCSIS access network is working properly and that the traffic is being handled as specified by the QoS parameters, in this case the Maximum Sustained Traffic Rate.

Finally, Figure 13b depicts the latency in the case of a pure DOCSIS access network and when the ALHINP architecture is used. It is worth noting that in both cases the latency increases with the frame size. This is the normal behavior, as larger frames pose higher processing times at the forwarding devices. As it can be seen in Table 3, even with the extra delay introduced by the ALHINP, which varies between the 10% and 17%, the latency remains below 2 ms. According to the ITU-T G.114 recommendation [35], if delays are kept below 150 ms, most applications are not significantly affected. Thus, the ALHINP architecture can be considered compliant with the ITU-T G.114 and therefore, the cost introduced by the additional elements is negligible, being a valid architecture to provide any kind of services, even applications such as VoIP or online gaming, that make use of small frame sizes.

7. Conclusions

To the best of our knowledge, the architecture presented in this paper represents the first working solution based on open standards able to dynamically provide L2 services with QoS requirements over a DOCSIS access network via

OpenFlow. On the one hand, it allows to control a DOCSIS access network using an OpenFlow controller without changing the underlying DOCSIS infrastructure. It differs from the other approaches for the SDN-ization of the DOCSIS access network presented by major network providers and manufacturers in that it can be applied over legacy network devices. Furthermore, as it is based on the abstraction of the DOCSIS access network as a wide area OpenFlow virtual switch, the OpenFlow controller remains agnostic to the DOCSIS particularities. In fact, this feature makes the ALHINP-based architecture a suitable candidate for its adoption by other access network technologies such as GPON.

Another important feature provided by this new architecture is that it makes possible the provisioning of L2 services in a dynamic fashion and after a successful AAA process. As a consequence, it provides the means to adopt the vCPE approach envisioned by SDOs and companies. This way, the CM provides basic L2 connectivity to the core network of the provider, whereas more advance network functions such as NAT or firewalling are performed by the OUIs.

Besides, a custom controller has been designed with advanced support for QoS control. In order to provide L2 services with QoS across the entire DOCSIS access network it has been extended with custom OpenFlow messages that allow the creation of QoS-enabled pipes in the OpenFlow devices. These QoS-enabled pipes in conjunction with the existing service flows between the CMs and the CMTS behave like QoS-enabled pipes at the Big Switch Abstraction level.

The cost introduced by the ALHINP architecture compared to a pure DOCSIS access network is considered negligible. The throughput remains practically unchanged for higher frame sizes whereas for small frame sizes it gets reduced up to 6%. The same happens with the latency, which in the worst case scenario, for small frame sizes, introduces an additional 40% of delay. In both cases, the metrics that have been used remain compliant with the recommendations of the ITU-T. As a consequence, we consider these results satisfactory taking into consideration the extra functionalities provided by the architecture.

In summary, our solution meets all the requirements for the adoption of the SDN concepts and the vCPE approach in legacy access networks. Even more, it remains compliant with the ITU-T G.114, which specifies the acceptable delays for voice applications and the Y.1541 that specifies the network performance objectives for IP-based services.

Acknowledgements

This research was partly funded by the Seventh Framework Programme of the European Commission, through the ALIEN (317880) project and by the Spanish Ministry of Economy and Competitiveness under the Secure deployment of services over SDN and NFV based networks project S&NSEC TEC2013-47960-C4-3-P. The authors want to thank Norberto Ojinaga and Octavio Alfageme from Euskaltel and Pedro Ortega from CISCO for their support.

References

[1] Open Networking Foundation, Software-Defined Networking: The new norm for networks, Tech. rep., white paper (2012).

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, OpenFlow: Enabling Innovation in Campus Networks, SIGCOMM Comput. Commun. Rev. 38 (2) (2008) 69–74.

[3] R. Mehmood, R. Alturki, S. Zeadally, Multimedia applications over metropolitan area networks (MANs), Journal of Network and Computer Applications 34 (5) (2011) 1518 – 1529, dependable Multimedia Communications: Systems, Services, and Applications. doi:http://dx.doi.org/10.1016/j.jnca.2010.08.002.

[4] CableLabs, An architecture for using SDN to support virtual cable modems and device configuration in DOCSIS based networks, Invention Disclosure (2014).
URL http://www.cablelabs.com/wp-content/uploads/2014/04/60653-publish-on-CL-website.pdf

[5] Broadband Forum, Work on Network Enhanced Residential Gateway (WT-317) and Virtual Business Gateway (WT-328), Tech. rep. (mar 2014).

[6] AT&T, AT&T Domain 2.0 Vision White Paper, Tech. rep., white Paper (jan 2015).
URL http://www.att.com/

[7] S. Hubbard, Implementing the Innovative Edge for Cloud-Based Services, Tech. rep., Juniper, white Paper (2012).

[8] D. R. Lopez, B. Iribarne, Business Cases for Virtual Customer Premises Equipment (vCPE) for end users, in: Terena Networking Conference 2013 (TNC2013), Terena, 2013, extended Abstract.

[9] R. Shaddad, A. Mohammad, S. Al-Gailani, A. Al-hetar, M. Elmagzoub, A survey on access technologies for broadband optical and wireless networks,

Journal of Network and Computer Applications 41 (0) (2014) 459 – 472. doi:http://dx.doi.org/10.1016/j.jnca.2014.01.004.

[10] H. Egilmez, S. Dane, K. Bagci, A. Tekalp, OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks, in: Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific, 2012, pp. 1–8.

[11] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, et al., B4: Experience with a globally-deployed software defined WAN, in: Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, ACM, 2013, pp. 3–14.

[12] K. Govindarajan, K. C. Meng, H. Ong, W. M. Tat, S. Sivanand, L. S. Leong, Realizing the Quality of service (QoS) in Software Defined Networking (SDN) based Cloud infrastructure, in: Information and Communication Technology (ICoICT), 2014 2nd International Conference on, 2014, pp. 505–510.

[13] W. Wendong, Q. Qinglei, G. Xiangyang, H. Yannan, Q. Xirong, Autonomic QoS management mechanism in Software Defined Network, Communications, China 11 (7) (2014) 13–23.

[14] M. Bari, S. Chowdhury, R. Ahmed, R. Boutaba, PolicyCop: An Autonomic QoS Policy Enforcement Framework for Software Defined Networks, in: Future Networks and Services (SDN4FNS), 2013 IEEE SDN for, 2013, pp. 1–7.

[15] S. Sen, D. Shue, S. Ihm, M. J. Freedman, Scalable, optimal flow routing in datacenters via local link balancing, in: Proceedings of the ninth ACM conference on Emerging networking experiments and technologies, ACM, 2013, pp. 151–162.

[16] A. Kassler, L. Skorin-Kapov, O. Dobrijevic, M. Matijasevic, P. Dely, Towards QoE-driven multimedia service negotiation and path optimization with Software Defined Nnetworking, in: Software, Telecommunications and Computer Networks (SoftCOM), 2012 20th International Conference on, IEEE, 2012, pp. 1–5.

[17] B. Sonkoly, A. Gulyas, F. Nemeth, J. Czentye, K. Kurucz, B. Novak, G. Vaszkun, On QoS Support to Ofelia and OpenFlow, in: Software Defined Networking (EWSDN), 2012 European Workshop on, 2012, pp. 109–113.

[18] A. Ishimori, F. Farias, E. Cerqueira, A. Abelem, Control of Multiple Packet Schedulers for Improving QoS on OpenFlow/SDN Networking, in: Software Defined Networks (EWSDN), 2013 Second European Workshop on, 2013, pp. 81–86.

[19] J. White, Can DOCSIS Networks Leverage SDN?, Tech. rep., ARRIS Inc. (2012).

[20] N. Nandiraju, S. Ozer, Applying the Software Defined Networking Paradigm to MSO Commercial networks, Tech. rep., ARRIS Inc. (2013).

[21] CableLabs, DOCSIS Service Flow provisioning via Openflow, Invention Disclosure (2012).

URL http://www.cablelabs.com/wp-content/uploads/2014/04/60421-publish.pdf

[22] Open DayLight Project, Packet Cable PCMM Project (jan 2015).
URL https://wiki.opendaylight.org/view/Project_Proposals:PacketCablePCMM

[23] CableLabs, PacketCable 2.0 Quality of Service Specification PKT-SP-QOS, Tech. rep. (oct 2014).

[24] Oliver Solutions, accessFlowNE (jun 2014).
URL http://oliver-solutions.com/accessflowne/

[25] CableLabs, DOCSIS 3.0 Physical Layer Specification CM-SP-PHYv3.0, Tech. rep. (aug 2013).

[26] ALIEN FP7 project (feb 2015).
URL http://www.fp7-alien.eu/

[27] CableLabs, DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification CM-SP-MULPIv3.0, Tech. rep. (oct 2014).

[28] J. Matias, E. Jacob, Y. Demchenko, C. de Laat, L. Gommans, Extending AAA Operational Model for Profile-Based Access Control in Ethernet-Based Neutral Access Networks, in: Evolving Internet (INTERNET), 2010 Second International Conference on, 2010, pp. 168–173. doi:10.1109/INTERNET.2010.37.

[29] J. Matias, A. Mendiola, N. Toledo, B. Tornero, E. Jacob, The EHU-OEF:

An OpenFlow-based Layer-2 experimental facility, Computer Networks 63 (2014) 101–127. `doi:10.1016/j.bjp.2013.11.013`.

[30] I2T Research Group (jan 2015).
URL `http://http://i2t.ehu.eus/`

[31] BISDN, extensible DataPath daemon (jan 2015).
URL `http://www.bisdn.de/wordpress/`

[32] BISDN, Revised OpenFlow Library (jan 2015).
URL `https://www.codebasin.net/redmine/projects/rofl-core/wiki/Tutorial`

[33] S. Bradner, J. McQuaid, Benchmarking Methodology for Network Interconnect Devices, Tech. rep., RFC 2544 (1999).

[34] S. Bradner, Benchmarking Terminology for Network Interconnection Devices, Tech. rep., RFC 1242 (1991).

[35] ITU-T, G.114 One-way Transmission Time, Tech. rep. (may 2003).