Optical Switching and Networking 42 (2021) 100631

Contents lists available at ScienceDirect

Optical Switching and Networking

journal homepage: www.elsevier.com/locate/osn

Experimental validation of an SDN residential network management proposal over a GPON testbed

Noemi Merayo [*], David de Pintos, Juan C. Aguado, Ignacio de Miguel, Ramón J. Durán, Patricia Fernández, Rubén M. Lorenzo, Evaristo J. Abril

Optical Communications Group of the Department of Signal Theory, Communications and Telematic Engineering, E.T.S.I. Telecomunicación, Universidad de Valladolid (Spain), Campus Miguel Delibes, Paseo de Belén 15, 47011, Valladolid, Spain

ARTICLE INFO

Keywords:
Software defined networking
Gigabit passive optical networks
Quality of service
Residential network management
Experimental validation

ABSTRACT

We propose, and experimentally demonstrate, an SDN (Software Defined Networking) new management solution for legacy GPONs (Gigabit Passive Optical Networks), which allows users to dynamically control their residential networks by means of a management application. In this way, users can customize the allocation of resources (and set constraints, if desired) to connected devices in their residential network, fast and efficiently. This real-time customization enables new business models for network operators and service providers. As a proof of concept and to validate the management solution, we demonstrate, in a testbed environment, the operation of a dynamic network scenario where an operator has a business model in which users have a contracted basic bandwidth, but they are allowed to increase it temporarily when using highly demanding services.

1. Introduction

The emergence of services with high bandwidth demands and extreme QoS requirements has produced a dramatic increase in network traffic. Consequently, there exists an important pressure on network control and bandwidth in optical networks, especially in the access segment. Under this scenario it is necessary to provide networks with new ways to communicate as well as greater intelligence and security [1]. In this way, Software Defined Networks (SDN) have appeared to respond to these new needs in deployed networks controlling devices by means of a common external software using a set of protocols. Indeed, SDN defines how a centralized SDN controller can communicate with the forwarding plane of virtual and physical network devices (routers or switches) using specific protocols (like OpenFlow [2], NETCONF [3] or RESTCONF [4]) so that the controller can manage them and can configure the path of packets across the network devices. Thus, SDN abstraction and separation of control and data planes improve the efficiency of networks, leading to better control of network traffic, enabling management automation, improving scalability, and providing a reduction of operating and hardware costs [1,5]. One important advantage of SDN is that it can complement legacy networks instead of replacing them, as far as possible.

On the other hand, Passive Optical Network (PON) technologies are becoming the *de facto* high-speed solution for wired access in many countries, and a high market penetration evolution in the short and medium term is expected. According to the September 2018 report by the FTTH Council in Europe [6], 160 million households in the European Union are connected by FTTH/B technology, representing an average of 37.4% of the total households. But the figures soar even more when the FTTH Council gives forecasts for the 2020–2025 period [7]. Forecasts predict the number of connected households will go from 189 million in 2020 (45.2%) to 263 million in 2025 (65%). Moreover, PON solutions will tend to be the predominant FTTH architecture in the coming years, going from 57% in September 2020 to a forecast of 73% in 2025.

Despite the great acceptance of SDN solutions in different network architectures, the integration of SDN in PONs is still in its early stages and its integration will have a great impact since, as we have just mentioned, it is the most widely deployed access technology worldwide. Nevertheless, it the integration is gaining momentum, as its implementation in residential homes, moving the service management from the home gateway to the cloud, can provide many advantages. Some of these advantages are related to the fact that subscribers connect many gadgets in their residential networks, such as computers, smartphones, tablets, smart TVs and IoT (Internet of Things) devices, but they are not able to manage the real bandwidth consumption or the network behaviour of each device, or at least are unable to do it in an easy way. In

---

[*] Corresponding author.
E-mail address: noemer@tel.uva.es (N. Merayo).

https://doi.org/10.1016/j.osn.2021.100631
Received 4 September 2020; Received in revised form 31 March 2021; Accepted 24 May 2021
Available online 29 May 2021

addition, home router devices from different vendors provide different interfaces to configure the residential network, and this configuration is lost when the router is replaced by another one, causing troubles to regular subscribers who do not have technological skills. Finally, Internet Service Providers (ISPs) are not interested in developing interfaces for network subscribers to deal with their bandwidth resources, since the residential market provides low profits to ISPs [8]. However, due to the explosion of connected devices and user applications, residential networks are becoming increasingly complex, and the development of user-friendly interfaces that permit residential users to deal with the desired behaviour of their services and applications, easily and transparently, should be a key point to allow a global and efficient network management. As a consequence, SDN can address these challenges, as it allows vendor-agnostic device management, and it permits that third-party applications can easily interact with the SDN controller, since SDN provides a common interface and a unified touch point for policy, control, and management [1]. This unified control allows a complete control of the network, as well as a more efficient supervision and updating of network devices without the need to change the hardware. Moreover, the centralized overview of all devices in the network permits a high degree of dynamism in the reconfiguration and control of services and client applications of residential users, even when they require them in real time. Even more, as network control and management are both programmable, it allows innovation of agile services [5]. In fact, SDN permits the creation and modification of services in a faster way, contrary to the methods implemented in traditional access networks, which require that PON equipment has to be synchronized every time that any modification in the GPON configuration is done. This behaviour of SDN positively impacts on the PON network performance in terms of latency. On the other hand, SDN allows the coexistence of multiple PON infrastructures from different vendors since the SDN protocols can interact in a transparent way with different equipment. Then, network operators/ISPs can easily control different PON devices using a common programmable interface, in contrast to traditional methods in which each manufacturer provides a specific control interface to interact with their specific devices in different manners. Finally, SDN makes it easier to apply centralized and automatized responses to network threats and to detect and block malicious traffic without affecting the normal network operation. As a consequence, many proposals are focusing in controlling residential networks using SDN techniques [8,14].

In this paper, we propose a novel SDN solution for residential networks over legacy PON architectures, so that users can manage the bandwidth associated with their home devices (smartphones, tablets, laptops), as well as control and impose restrictions on them, both dynamically and individually. In order to test our proposal, we have implemented that SDN solution (based on OpenFlow) over a legacy GPON testbed and, as a proof-of-concept, we have also programmed an easy-to-use web application that permits users to contract services dynamically, in real-time, even for a short period of time, as well as to interact with its connected devices. To the best of our knowledge this is the first time that a residential network management solution over PON infrastructures using SDN techniques is experimentally demonstrated.

The paper is organized as follows. Section 2 describes the state of the art regarding the integration of SDN in PON architectures. Section 3 explains the design and implementation of the proposed SDN residential network management solution over a GPON testbed. Section 4 presents the application of that solution to a legacy GPON testbed and discuss the experimental results obtained for a proof-of-concept business model. Furthermore, Section 5 describes an improvement of the initial SDN solution to manage residential networks over GPONs by using NAT. Finally, in Section 6, the most relevant conclusions obtained in this experimental study are shown.

2. State of the art

Nowadays, more and more home users are becoming concerned with their Internet usage and consumption, thereby demanding control and autonomy in the management of their own residential networks. Thus, several research works have approached the management of residential networks using SDN strategies [9,14,16,18–20], although none of those solutions are integrated with PON infrastructures. In particular, Kim et al. [9] have designed a system that permits subscribers to be aware of their Internet usage by means of OpenFlow, so that subscribers can visualize statistics and set bandwidth consumption policies for specific devices and applications. Likewise, Chetty et al. [10] have implemented a software tool (uCap) for residential users that, together with an OpenFlow virtual switch, exchanges statistics and control messages with an SDN controller that permits users to configure Internet usage policies for devices. Yiakoumis et al. [11] have developed a web-based management prototype so that home subscribers can prioritize some traffic or applications in terms of latency or bandwidth. In particular, they build a small user-ISP infrastructure on which residential subscribers can set priority levels to their applications. Gharakheili et al. [12,13] have identified use cases where subscribers can benefit from the control of their residential network, by providing Quality of Experience (QoE), bandwidth usage and parental control. Then, they have also implemented an SDN prototype where subscribers are able to restrict and manage the available bandwidth in their residential homes using a web interface. Mortier et al. [14] have designed a home router to monitor and control network traffic flows, using NOX [15] and OpenFlow to permit per-flow control, and a custom DHCP implementation to enable traffic isolation. Amiri et al. [16] have designed an SDN scheme to share the bandwidth among different applications in residential networks, giving priority to applications such as online games to provide QoE. This solution was implemented using Mininet as a network emulator [17]. Other authors have proposed slicing home networks so that multiple service providers can share a common infrastructure [18–20]. This type of architecture is able to provide bandwidth and traffic isolation between slices, as well as independent control of slices.

Apart from the SDN management of residential networks, other proposals focus on integrating SDN in PONs with different objectives. Indeed, since SDN-based OLTs (Optical Line Terminals) and ONTs (Optical Network Units) are not available yet, most of studies implement an abstraction layer in order to turn legacy devices into SDN-controllable devices, for example using OpenFlow. In this line, Lee et al. [21] have developed an OpenFlow agent inside an OLT so that it can simultaneously interact with the OLT in its native language and with an external SDN controller. Moreover, authors in Ref. [22] developed a GPON-based SDN-enabled virtual switch so that the GPON network becomes a single Open-Flow switch (using the OpenFlow protocol). In addition, they also designed virtual Ethernet switches on top of an OpenFlow network so each subscriber is provided with one switch to simplify the control and management of their residential devices. Similarly, Clegg et al. [23] have described an architecture that integrates OpenFlow on many access technologies, testing the software on a Gigabit Ethernet Passive Optical Network (GEPON). Authors in Refs. [24,25] have implemented modules to map OpenFlow messages to native PON commands that can be legible to existing OLTs. In this research context, Parol and Pawlowski [26] have proposed an extension of the OpenFlow protocol for GPONs (OpenFlowPLUs) defining new OpenFlow messages to interact with the Optical Management and Control Interface (OMCI) channel of the GPON standard. Other different approaches focus on implementing specific bandwidth allocation or service configuration strategies inside the SDN controller, moving these issues out of the OLT [27,28]. Other studies propose the virtualization of PONs [29,30], that is, virtual PONs that can operate with different protocols or bandwidth allocation strategies using SDN. On the other hand, other research works allow energy savings in PONs using SDN techniques [31,32]. Besides, SDN is also applied into Long-Reach PONs

(LR-PON) to allow strong protection mechanisms since due to its long reach and large split ratio any cut in the single feeder optical fiber may interrupt services for several thousand users [33]. Finally, the Virtual OLT Hardware Abstraction (VOLTHA) open source project, proposed by the CORD initiative [34], abstracts the PON network as a programmable Ethernet switch to be controlled by an SDN controller. It communicates with PON hardware devices using vendor-specific protocols through OLT and ONU adapters in order to provide control of the access network and its configuration. On the south side, VOLTHA communicates with the vendor protocols using the OpenOLT agent that is running on white box OLTs. This agent is used by VOLTHA through an adapter (OpenOLT adapter) [35]. However, the OpenOLT agent uses Broadcom's BAL (Broadband Adaptation Layer) software for interfacing with chipsets in OLTs and part of this software is not open source and some proprietary source code is required to build the drivers. Moreover, the CORD initiative is focused on the disaggregation of functions, and VOLTHA is more specifically focused on the disaggregation of PON architectures to functional modules and on the creation of an abstraction layer to which every vendor will be able to map. In this way, SEBA (SDN Enabled Broadband Access) [36], which also belongs to the CORD initiative, seeks to generalize access technologies. In fact, SEBA supports several virtualized access technologies (PON, G.Fast, DOCSIS) as well as residential access and wireless backhaul and it includes NEMs (Network Edge Mediators) to provide mediation to different operators' backend management systems. Finally, the Open Source Access Manager (OSAM) proposal [37] intents to integrate with ONAP (Open Network Automation Platform) for orchestration. In this way, ONAP is a platform for orchestration, management, and automation of network and edge computing services for network operators, cloud providers and so on [38]. In this global scenario, each SEBA device is considered a "black-box" and OSAM defines the flows and scenarios, such as adding subscribers or activation of OLT, in a SEBA device.

In summary, some previous works have provided solutions for the SDN-based control of residential networks, but those solutions do not provide joint control of the access networks supporting those residential networks. Other works have exclusively focused on the application of the SDN paradigm to passive optical networks, thus transforming GPON devices in SDN-controllable devices. However, none of the past proposals have taken into consideration both issues simultaneously: SDN-based control of a residential network coupled with SDN-control of a legacy PON infrastructure. Some work in that line has been done by Flores Moyano et al. [39,40], who have integrated Network Function Virtualization (NFV) to ensure QoS for SDN passive optical access networks and residential networks. However, that work is mainly focused on the virtualization of the residential gateway rather than on testing residential network management over PON infrastructures.

In this paper, which extends our initial work in Ref. [41], we implement and experimentally validate an OpenFlow end-to-end SDN residential network management proposal over GPONs, to provide efficient and full control of the residential network, ensuring fulfilment of QoS requirements to subscribers, but also demonstrating full functionality without modifying legacy GPONs. In particular, we propose an experimental solution in which residential users can perform several actions, like managing their traffic flows by defining and controlling their network resources, or even contracting temporarily dynamic services or extra-bandwidth, using a developed management web application.

3. Design and implementation of an SDN solution to manage residential networks over GPONs

A. Description of the SDN approach implemented over a GPON

We propose to integrate an SDN approach over legacy GPONs which permits to configure the access network by means of OpenFlow using an external SDN controller and several OpenFlow switches. The architectural solution is shown in Fig. 1. To turn legacy OLTs and ONTs into OpenFlow controllable devices, an SDN hardware abstraction layer must be implemented. Therefore, we propose to use OpenFlow Virtual Switches (OVS) [42] connected to the OLT and to ONTs, and an OpenDayLight (ODL) controller (Fig. 1) [43]. However, other SDN controllers, such as ONOS [44], could also be used. In this network scenario, the SDN controller will be able to dynamically modify services according to real network traffic or user requirements, allowing a flexible control of the GPON capabilities. Therefore, the SDN controller belongs to ISPs/Network Operators that provide Internet connections and services to their customers. Moreover, as GPONs operate in two channels with different wavelengths, the downstream channel (from the OLT to ONTs) and the upstream channel (from ONTs to the OLT), the SDN controller has to deal with the traffic in both channels, so that the contracted services of the network subscribers comply with the corresponding QoS requirements (e.g., guaranteed bandwidth).

Ideally, the OpenFlow switches (OVS) should be integrated inside the OLT and the ONTs (SDN based OLT/ONTs). However, as SDN-based OLTs and ONTs are not yet available, the OVSs can be embedded in an external hardware like Raspberry Pi, Banana Pi, mini-computer or a computer. In particular, one OVS should be implemented in a computer logically co-located with the OLT (Central OVS, COVS) to emulate the SDN layer of the OLT, and thus to control the downstream traffic of the entire GPON (Fig. 1). On the other hand, at the users' side, an OVS (Remote OVS, ROVS) lies beside each ONT to control the upstream traffic requirements of each subscriber. These devices will handle a lower computation load than the COVS, so cheaper devices than computers can be used for ROVS, like Raspberry Pi, as shown in ONT_1 and ONT_2 in Fig. 1.

To set an SDN configuration, the ODL controller sends OpenFlow messages to every ROVS through the conventional GPON channels and to the COVS through a direct connection. First, the SDN controller sends flow tables to the virtual switches (COVS, ROVS) to configure the services (Internet, HDTV, VoIP) and their QoS requirements. In fact, the OpenFlow tables are programmed by the SDN controller (that belongs to ISPs/Network Operators) and their entries are modified in real-time each time residential users demand new services or any modification in their QoS requirements. Then, two flows are created for each service as the GPON operates in two channels: one for managing the downstream QoS requirements and another for the upstream QoS requirements. The former flow is created in the COVS and the later in the ROVS. The match instructions and fields of each flow differ depending on the channel and on the developed functionalities, as it will be explained in the next sections. For example, if the maximum bandwidth is going to be controlled by means of SDN, the bandwidth rate assigned to each service (at both channels) is measured with OpenFlow meters [45]. A meter measures the rate of packets assigned to it and enables controlling the rate of those packets. Meters are attached directly to flow entries. In our proposal each meter entry consists of a meter identifier and a meter band. The meter band specifies the maximum rate associated with that flow (band rate) and the way to process the packets of the flow (band type). Besides, some band types have optional arguments (*drop*, *dscp remark*) called type specific arguments and we use the *drop* option. Then, the data rate of each meter band is continuously measured, and if the rate is larger than the value defined in the band rate *drop*, packets are discarded, so this choice can be used to define a rate limiter band. Therefore, to control the maximum bandwidth of one specific service by means of meters, it is necessary to attach 1 m to each flow, one for the maximum downstream bandwidth and another for the maximum upstream bandwidth of the service. Then, the band rate defines the maximum bandwidth associated with the service at both channels (upstream, downstream). Finally, it should be noted that since residential users are not expected to know the concept of flows and meters, this low-level process is totally abstracted into higher-level commands that can be understand by users, such as guaranteed bandwidths.

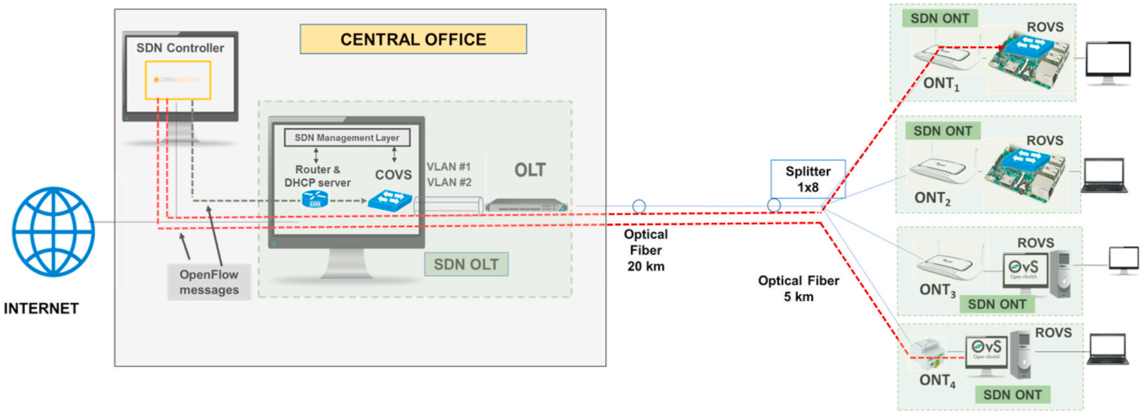


Fig. 1. SDN residential network management proposal over a GPON testbed.

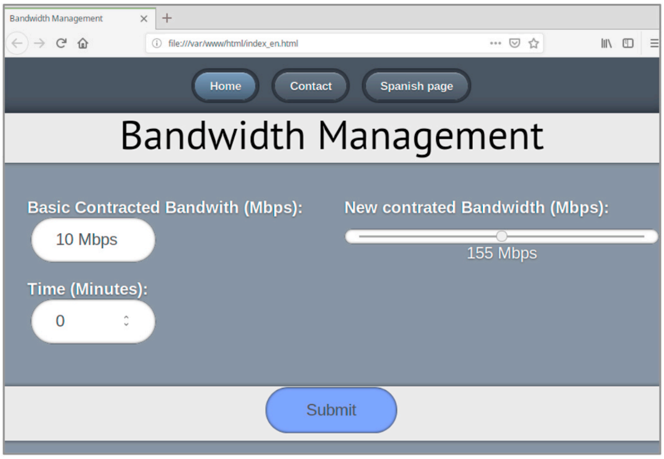
On the client side, GPONs can employ level 3 (L3)-model ONTs, which integrate router functionalities, and level 2 (L2)-model ONTs, without routing functionalities. In real PON deployments, residential users should use vendor-provided devices due to security restrictions imposed by the PON standards when the user authentication process is performed. In this way, ONTs must be compatible with the specific OLT to which they are connected in the Central Office and identification keys must be exchanged to be registered in the PON. Furthermore, L2 ONTs act as switches and they are transparent for network devices, so that it is necessary to add an L3 element to route the packets towards the residential network and to provide unique IP address to the different devices connected to the ONT. For the implementation of the proposed SDN environment over GPONs, L2 ONTs are preferred due to flexibility, as they are not limited by the L3 vendor ONTs routing options. Thus, we deploy OVS and configure flows differentiated by IP addresses, acting similarly to a router although they are L2. Then, every ROVS needs a local DHCP server to provide IP addresses to the devices connected to its associated L2 ONT. First, the global DHCP server of the service providers or network operators assigns a range of IP addresses to each local DHCP server in the ROVSs, and then each ROVS assigns IP addresses to each connected device. It is worth noting that the IP addresses are no longer local to the residential network. Thus, every device will have a unique IP address in the GPON.

Regarding the OLT configuration, in legacy GPONs, to allow variations in the subscribers' services and traffic, a minimum guaranteed bandwidth is given to every client, so that the Dynamic Bandwidth Allocation (DBA) algorithm in the GPON layer is in charge of ensuring the fulfillment of that guarantee [46,47]. Apart from this guaranteed bandwidth, every client may receive a non-guaranteed bandwidth, which is also handled by the DBA. However, in our SDN proposal over GPON the bandwidth will be also controlled by the SDN layer to permit a tighter control of the bandwidth allocation process. The SDN layer limits the maximum available bandwidth for each subscriber/service (according to the user contract) and ensures that the total maximum bandwidth of the GPON is not exceeded. Then, the DBA is configured to treat subscribers equally, but allocating bandwidth taking into account real-time traffic demands, as the differentiation and the fulfilment of bandwidth guarantees is provided by the SDN layer (dropping those packets exceeding the maximum assigned bandwidth). Consequently, this approach avoids modifying the legacy GPON layer, provides the full functionality of an SDN solution, and the excess-bandwidth, which is administered by the DBA algorithm in the GPON layer (layer 2), is now also controlled by the SDN layer.

B. Development of a web service application to control the residential network

As a proof of concept, and to validate the SDN solution, we consider the operation of a dynamic network scenario. The operator/service provider of such a network has a business model in which users have a contracted basic bandwidth, but are allowed to increase it temporarily when using highly demanding services, such as online games or Ultra High Definition television (paying an extra fee for this temporal use of additional bandwidth) by means of SDN techniques. Moreover, users are able to control the bandwidth consumption of each connected device. However, since residential users are not able to deal with low-level SDN concepts, this process is abstracted into high-level commands implemented in a transparent way in a web service application. In fact, we have designed an easy-to-use web application, which integrates all these functionalities and that the provider could offer to customers to use this service and manage their residential networks. The web application is installed on the same device as the ROVS (i.e., at the customer premises), so that only a machine directly connected to the ROVS can access to the application.

As shown in Fig. 2, the web application has a simple interface. It informs the customer of the current contracted bandwidth ("Basic Contracted Bandwidth"), but it also allows the customer to increase the contracted bandwidth ("New Contracted Bandwidth") for a period of time ("Time") by paying an additional fee. Then, the residential user can send a request through the web application (Fig. 2) with the new contracted bandwidth. Then, this application sends an OpenFlow message with these parameters to the SDN management layer of the ISP/Network Operator (at the OLT side) that configures the GPON network with OpenFlow. Once the request for bandwidth increase arrives at the service provider system, it verifies that there is enough bandwidth for the request to be granted, and then it sends an affirmative or negative



Bandwidth Management

Basic Contracted Bandwith (Mbps): 10 Mbps

New contrated Bandwidth (Mbps): 155 Mbps

Time (Minutes): 0

Fig. 2. Web application interface to hire a time-limited higher bandwidth.

response to the subscriber by means of the SDN controller. In the affirmative case, a new temporal flow is created using the "hard_timeout" field to indicate the lifetime of the flow in seconds, so that, when the service expires, this flow is eliminated and the basic service contracted ("Basic Contracted Bandwidth") will prevail. Moreover, the new flow contains the new contracted bandwidth restriction (associated with a new created meter) and the match instructions with the Ethernet address (MAC address) of the ONT (residential user).

On the other hand, the application also enables to control services and bandwidth requirements per individual device (tablets, smartphones, laptops, TVs, etc.) through OpenFlow. Thus, users can manage the resources within their residential networks, in such a way that they can restrict the maximum bandwidth associated with each connected device. To support this functionality, every device must be univocally identified at GPON network level, rather than at the residential network level. For this reason, L2 ONTs are highly preferred (as previously mentioned), as they allow us to implement specific router functionalities to achieve an efficient SDN control. As it is shown in Figs. 3 and 4, the web interface shows every connected device and the permitted actions on them, i.e., setting bandwidth consumption restrictions. These restrictions can be imposed in two different ways:

- o Ad-hoc maximum bandwidth restriction: the subscriber can limit the assigned bandwidth to a particular device, always lower than the total contracted bandwidth (Fig. 3).
- o Bandwidth management by category: the client can assign different categories ("High", "Medium" or "Low") to each connected device (Fig. 4).

The steps of the process are represented in Fig. 5. When a network subscriber requests setting a restriction for one of his/her connected devices, the web application communicates with the management system of the service provider. In fact, it communicates with the SDN management layer that configures the GPON network through OpenFlow (Fig. 5). The web application of the residential user sends an OpenFlow message indicating the type of service, the MAC address of the device to be restricted and the bandwidth restriction or the selected category. Once the message has been received by the SDN management layer at the OLT, it checks that the constraint does not exceed the global maximum bandwidth restriction, and gives an order to the SDN controller to create a new OpenFlow flow with the desired bandwidth restriction (by setting the associated meter as explained in Section III. A), adding a match for the Ethernet address (MAC address) of the device and the range of IP addresses assigned to the associated ONT. This double match is necessary since one restricted device (with a specific MAC address) could eventually move to another residential network, which is managed by the same SDN controller (e.g., when the owner of a phone visits a neighbor's house and connects it to her network), so the previous imposed restriction should disappear. It is worth noting that we have a global flow with the total basic bandwidth contracted by the subscriber, but at the same time we can have several flows restricting the bandwidth of certain devices, for example, a tablet, a smartphone, a laptop, computer, etc. In this way, the packets retransmitted by the OVS must be filtered by both flows, one for the total basic service and another for each specific device if it shows any imposed restriction.

4. Validation analysis and results

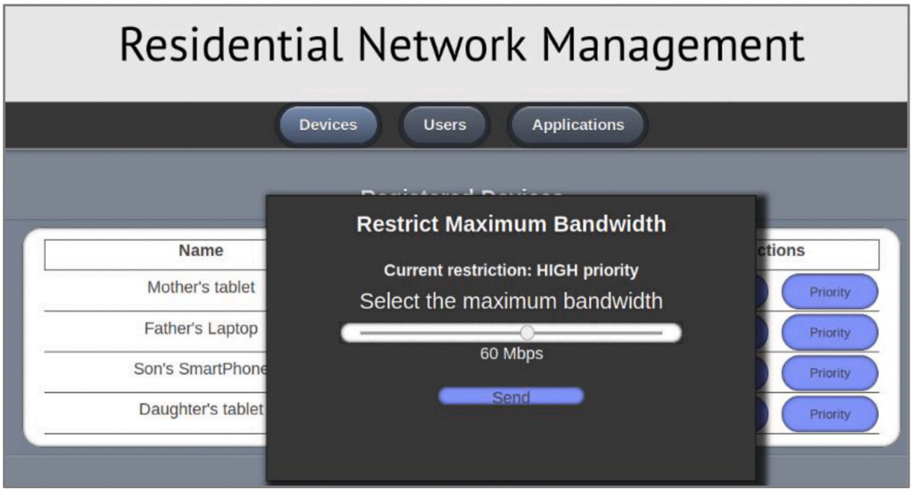
We consider a new business model in which users can directly configure their residential network and customize the services contracted by the network operator using an easy-to-use web interface. We first describe the GPON testbed scenario where we have implemented the solution and then analyze two use cases, the first one related to the contract of temporarily higher bandwidth and the second related to the internal control of the residential network devices.

A. GPON testbed scenario

In order to check our proposal and to demonstrate its performance and viability we have implemented the SDN residential network management proposal over a GPON testbed (Fig. 6) using equipment from the Telnet-RI vendor [48]. It includes an OLT SmartOLT 350, which implements a full-duplex GPON interface of 2.488 Gbps (downstream) and 1.244 Gbps (upstream) following the ITU-T G.984.x specifications. The connection between the OLT and the optical splitter includes three spools of Standard Single Mode Fiber (SSMF) of different lengths. The testbed is equipped with optical splitters connected to the ONTs by distribution fibers. The length of each link can be individually configured, using a connection panel, from 100 m up to 5 km, so that the testbed can emulate realistic scenarios (different distances from the clients to the Central Office). On the client side, we have used two Level-2 ONTs (Wave Access 512 model [48]), configuring the testbed so that they were located approximately 20 km away from the Central Office. Moreover, we have attached computers to work as routers with flexible capabilities by means of OVSs.

B. Use Case 1: Hiring temporarily higher bandwidth

In this use case, we consider that subscribers can temporarily request to increase their guaranteed bandwidth when using highly demanding


Residential Network Management

Restrict Maximum Bandwidth

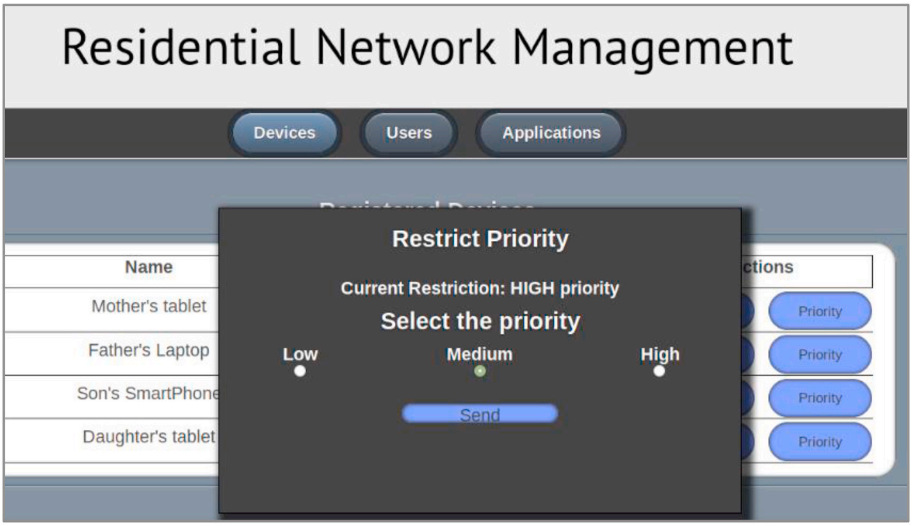
Current restriction: HIGH priority

Select the maximum bandwidth

60 Mbps

Name	ctions
Mother's tablet	Priority
Father's Laptop	Priority
Son's SmartPhone	Priority
Daughter's tablet	Priority

Fig. 3. Web application interface to set ad-hoc maximum bandwidth restrictions over devices.



Residential Network Management

Name	ctions
Mother's tablet	Priority
Father's Laptop	Priority
Son's SmartPhone	Priority
Daughter's tablet	Priority

Restrict Priority

Current Restriction: HIGH priority

Select the priority

○ Low ● Medium ○ High

Fig. 4. Web application interface to set bandwidth management restrictions by categories over devices.

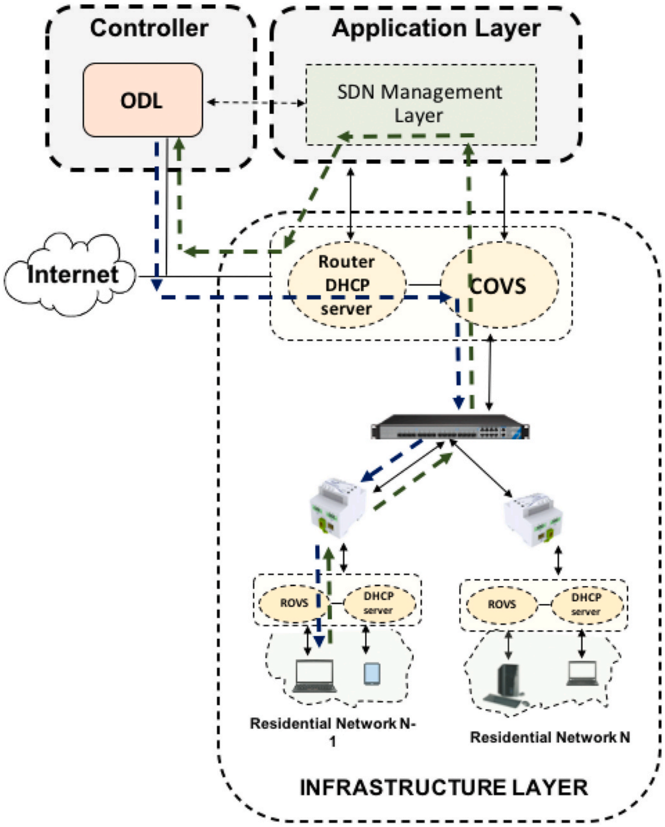


Fig. 5. Block diagram of the end-to-end SDN approach over a GPON to manage residential networks.

services by paying a higher fee, in our particular case an Internet service. One advantage of applying SDN for this kind of functionality is that it provides an open interface on the controller to allow for automated control of the entire GPON network and its services in a centralized way. Furthermore, as the GPON management becomes programmable, SDN streamlines and speeds up the modification of real-time QoS requirements in services, in contrast to implemented methods in legacy PONs that require a synchronization process of the PON equipment (OLTs and ONTs) each time that services are created or modified. This behaviour positively impacts on the GPON network performance in terms of latency or throughput. Therefore, as this service management is done using SDN techniques, the ODL controller builds the flow entries to update the service configuration of that ONT and sends the corresponding OpenFlow messages to the COVS and ROVS with the new meter configuration, as shown in Fig. 7, which displays real time network data captured with Wireshark [49]. In this example, for ONT$_1$ we consider that the user has contracted an Internet service of 50 Mbps, so an initial meter with the band rate to 50 Mbps, which corresponds with the basic contracted bandwidth must be enabled.

To test the dynamicity in bandwidth request and allocation, the subscriber, by means of the management application, requests at t = 180 s, to increase the guaranteed bandwidth to 150 Mbps during 3 min. Once that period expires, i.e., at t = 360 s, the network returns to the initial configuration. Therefore, Fig. 7 shows the OpenFlow messages regarding the upstream channel when the subscriber of ONT$_1$ (IP address 192.168.0.2) asks for a temporary increase on its bandwidth. It can be noticed that the ODL controller sends an OFTP_FLOW_MOD message at t = 180 s, that is, when the subscriber asks for increasing its bandwidth to 150 Mbps (meter ID = 18363338), and a destination match field (OFPXMT_OFB_ETH_DST) with the MAC address of the computer that runs the OVS towards the upstream channel (MAC address 90:e2:ba:e2:42:b2). As it can be observed, this information is contained in an OFMP_METER_CONFIG message, where the rate field contains the meter rate of 150 Mbps (rate: 153600 bps).

Moreover, Fig. 8 shows the real time throughput for the upstream services of ONT$_1$, measured using Wireshark. It can be seen that, at t = 180 s, the ONT increases its bandwidth to the requested value of 150 Mbps, and the transition to the new levels is very fast (around 2 s), accurate and stable. After that, at t = 360 s (when the 3 min period expires), the ONT returns to its initial bandwidth configuration, that is, 50 Mbps. Fig. 8 also shows a slightly higher transmission rate than that expected. This is due to the fact that Wireshark includes the packet headers when computing the bandwidth, but OVS does not. Although we only represent the upstream channel results, the results for the downstream link are similar.

C. Use Case 2: Residential Network Management

In this second use case, the developed web interface allows network subscribers to directly interact with the control of their connected devices in the residential network. The network automation provided by SDN makes it easier to ensure security in residential homes, since SDN is able to automatically detect and cut off resources (such as bandwidth) when malicious attacks happen, in contrast to many manual reviews that

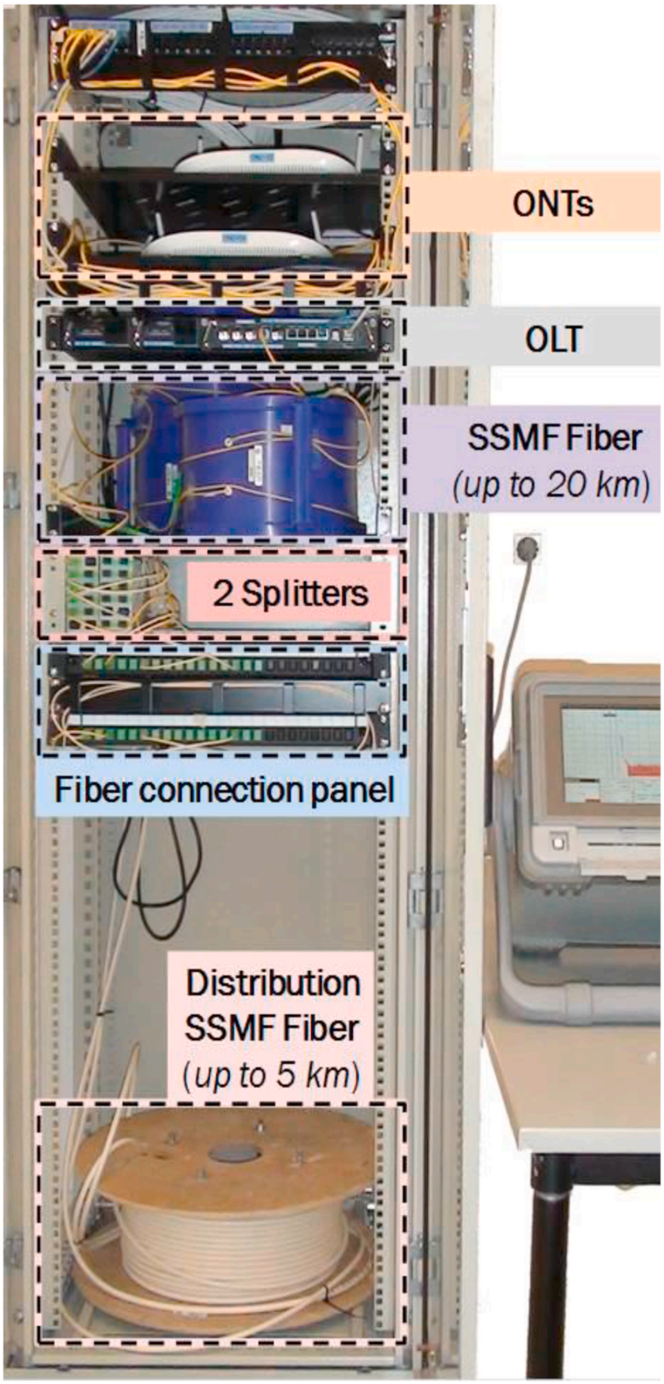


Fig. 6. GPON testbed with its PON devices.

administrators have to make in currently deployed networks. Even more, SDN can automatically adapt the security to specific threats in the immediate environment [50], for example for one particular device/application, without affecting the network operation of the remaining devices/applications in the same residential network. Then, this case study focuses on the SDN management of the residential network using two Level-2 ONTs, so that users and devices (TV, smartphones, laptops, etc.) are visible for the central computer, which emulates a central system of the service provider. This is achieved by means of an OVS and a DHCP server, which make it possible for all user devices to belong to the same subnet but also to distinguish which ONT (residential network) they belong to.

By being able to differentiate between end devices, the possibility of restricting bandwidth requirements per device is added through OpenFlow. Therefore, it was implemented a functionality in the web application so that subscribers can manage the resources within their residential networks, in such a way that they can restrict the bandwidth associated with each connected device or give a priority to each device. Thus, we have developed a method that automatically registers those connected devices in a local database hosted in the same place as the web application, storing the name and the MAC address of the device. As an example, Fig. 9 shows all the devices registered in the web application of the residential network of ONT_4. In the next subsection we analyze the performance of the two policies mentioned in Section III.B (ad-hoc bandwidth restriction, and bandwidth management by categories), applying them to the devices "*Son's smartphone*" and "*Daughter's tablet*" (Fig. 9). The experimental analysis has been carried out at both channels, but only the results of the downstream channel are presented due to lack of space. However, the performance of the upstream channel is similar.

1. AD-HOC Bandwidth Restriction Management

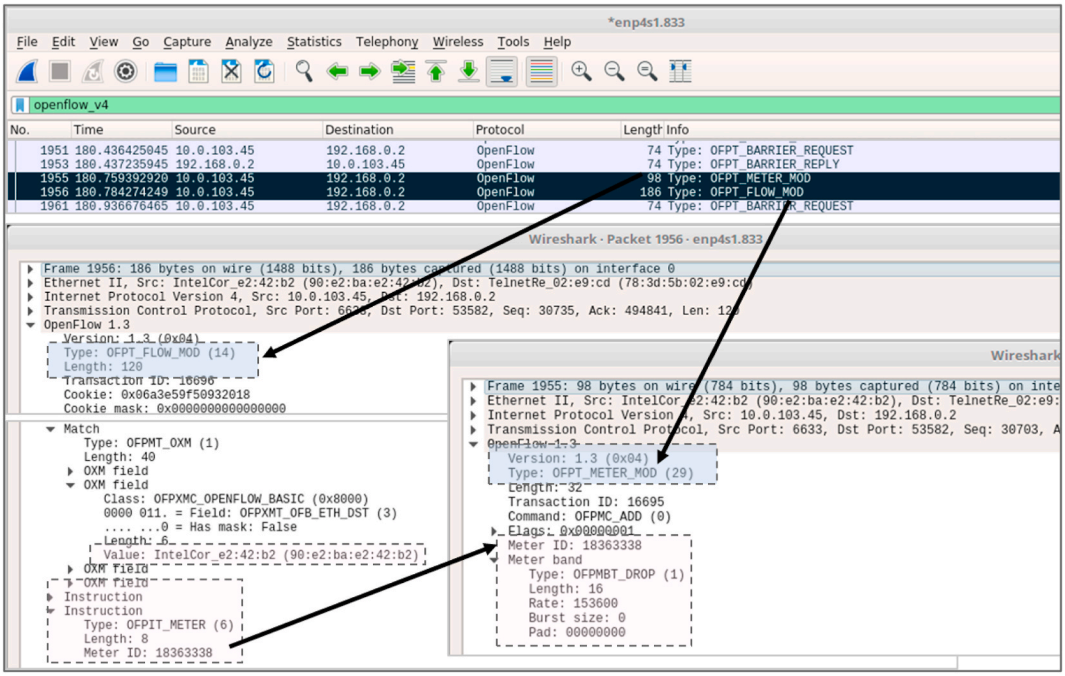
In this example we consider a network subscriber with a basic contracted bandwidth of 100 Mbps and several devices composing the residential network, as it can be observed in Fig. 9. However, this study only considers that devices "Son's smartphone" and "Daughter's tablet" are transmitting (Table I). Then, the user would like to apply parental control over some devices ("Son's smartphone" and "Daughter's tablet") in order to restrict their bandwidth consumption. Initially, none of the devices shows any restriction (so that they can transmit up to 100 Mbps), but after 3 min (180 s) the subscriber imposes restrictions for these devices, setting 20 Mbps and 40 Mbps for the "Son's smartphone" and for the "Daughter's tablet" respectively, as shown in Table 1. As an example of why this type of functionality may be interesting, it could be, for example, the COVID-19 pandemic, and the need to reserve a person's bandwidth during specific moments during teleworking (important meetings by videoconference, teaching by videoconference, and so on) limiting the bandwidth of the rest of the family (eg. leisure) at those times.

Fig. 10 shows a Wireshark capture of OpenFlow messages sent to the OVS that controls the "Daughter's tablet" device (MAC address d4:5d:df:0b:02:1b). It can be noticed that the ODL controller sends an OFTP_FLOW_MOD message with the instruction to restrict its bandwidth to 40 Mbps (meter ID = 197147) at minute 3 (around the second 180). As it can be observed, this information is contained in an OFMP_METER_CONFIG message, where the rate field contains the meter band of 40 Mbps expressed in bits per second, that is, 40960 bps.

Moreover, Figs. 11 and 12 show the real time evolution of the assigned bandwidth for both devices ("Son's smartphone" and "Daughter's tablet") using Wireshark. Thus, during the first 180 s, since the subscriber has not imposed any restriction, the ROVS shares the available bandwidth equally between both devices, thereby leading to a throughput of 50 Mbps for each device. Then, in the next 3 min, when the subscriber has already set bandwidth restrictions to both devices, it is observed that each device receives the maximum rate specified in the restriction, that is, 20 Mbps for the "Son's smartphone" (Fig. 11) and 40 Mbps for the "Daughter's tablet" (Fig. 12). Moreover, it can be observed in both graphs that these restrictions are applied very fast (around 2 s) to the assigned bandwidth of each device. If the sum of every restricted bandwidth was higher than the total contracted bandwidth, the ROVS would not allow exceeding the limit, so it would share the bandwidth between devices proportionally to the imposed restrictions.

2. Bandwidth Management by category

In this use case, we analyze the restrictions imposed on devices based on three categories. The high category has no restrictions, while the medium category is limited to half the basic contracted bandwidth, and



```
No.   Time            Source         Destination    Protocol   Length Info
 1951 180.436425045 10.0.103.45    192.168.0.2     OpenFlow        74 Type: OFPT_BARRIER_REQUEST
 1953 180.437235945 192.168.0.2    10.0.103.45     OpenFlow        74 Type: OFPT_BARRIER_REPLY
 1955 180.759392920 10.0.103.45    192.168.0.2     OpenFlow        98 Type: OFPT_METER_MOD
 1956 180.784274249 10.0.103.45    192.168.0.2     OpenFlow       186 Type: OFPT_FLOW_MOD
 1961 180.936676465 10.0.103.45    192.168.0.2     OpenFlow        74 Type: OFPT_BARRIER_REQUEST
```

```
▸ Frame 1956: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits) on interface 0
▸ Ethernet II, Src: IntelCor_e2:42:b2 (90:e2:ba:e2:42:b2), Dst: TelnetRe_02:e9:cd (78:3d:5b:02:e9:cd)
▸ Internet Protocol Version 4, Src: 10.0.103.45, Dst: 192.168.0.2
▸ Transmission Control Protocol, Src Port: 6633, Dst Port: 53582, Seq: 30735, Ack: 494841, Len: 120
▾ OpenFlow 1.3
    Version: 1.3 (0x04)
    Type: OFPT_FLOW_MOD (14)
    Length: 120
    Transaction ID: 16696
    Cookie: 0x06a3e59f50932018
    Cookie mask: 0x0000000000000000
  ▾ Match
      Type: OFPMT_OXM (1)
      Length: 40
    ▸ OXM field
    ▾ OXM field
        Class: OFPXMC_OPENFLOW_BASIC (0x8000)
        0000 011. = Field: OFPXMT_OFB_ETH_DST (3)
        .... ...0 = Has mask: False
        Length: 6
        Value: IntelCor_e2:42:b2 (90:e2:ba:e2:42:b2)
    ▸ OXM field
    ▸ OXM field
  ▸ Instruction
  ▾ Instruction
      Type: OFPIT_METER (6)
      Length: 8
      Meter ID: 18363338
```

```
▸ Frame 1955: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on inte
▸ Ethernet II, Src: IntelCor_e2:42:b2 (90:e2:ba:e2:42:b2), Dst: TelnetRe_02:e9:
▸ Internet Protocol Version 4, Src: 10.0.103.45, Dst: 192.168.0.2
▸ Transmission Control Protocol, Src Port: 6633, Dst Port: 53582, Seq: 30703, A
▾ OpenFlow 1.3
    Version: 1.3 (0x04)
    Type: OFPT_METER_MOD (29)
    Length: 32
    Transaction ID: 16695
    Command: OFPMC_ADD (0)
  ▸ Flags: 0x00000001
    Meter ID: 18363338
  ▾ Meter band
      Type: OFPMBT_DROP (1)
      Length: 16
      Rate: 153600
      Burst size: 0
      Pad: 00000000
```

Fig. 7. OpenFlow messages sent by the ODL controller to the ONT_1 with the new allocated bandwidth.

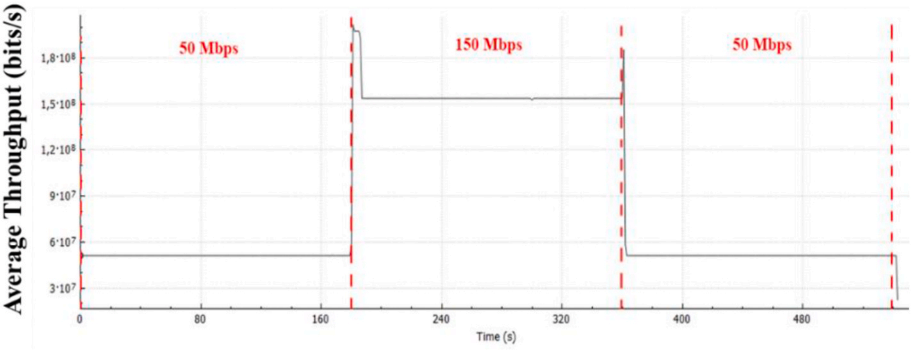
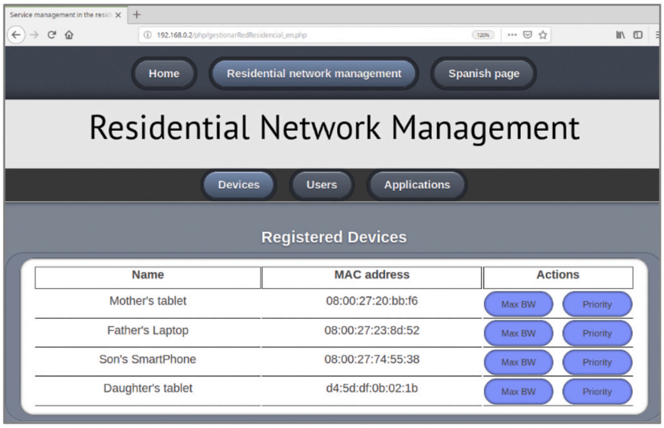


Fig. 8. Real time throughput of ONT_1 captured with Wireshark.



Registered Devices

Name	MAC address	Actions
Mother's tablet	08:00:27:20:bb:f6	Max BW Priority
Father's Laptop	08:00:27:23:8d:52	Max BW Priority
Son's SmartPhone	08:00:27:74:55:38	Max BW Priority
Daughter's tablet	d4:5d:df:0b:02:1b	Max BW Priority

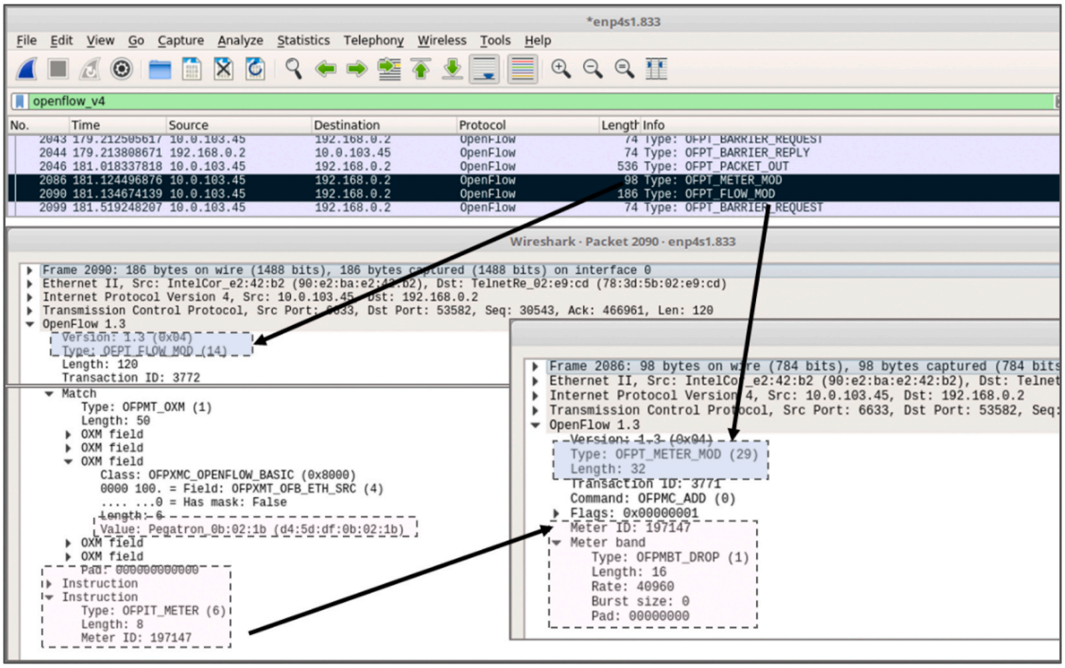
Fig. 9. Registered devices in the residential network of the ONT.

Table 1
Bandwidth Restrictions imposed on both devices.

Device name	Maximum Bandwidth Restriction	Basic Contracted Bandwidth
Son's Smartphone	20 Mbps (from second 180)	100 Mbps
Daughter's Tablet	40 Mbps (from second 180)	100 Mbps

the low category is limited to one third of the contracted bandwidth. Although our proof of concept scenario implements a simple policy, service providers could integrate more complex or different category policies. In this use case, we consider the same devices as before ("Son's smartphone" and "Daughter's tablet"), and a basic contracted bandwidth of 100 Mbps. Therefore, the high category (no restrictions) can provide up to 100 Mbps, the medium category is restricted to a maximum of 50 Mbps, and the low category is limited to a maximum of 33 Mbps. The test is divided into four intervals of time, as shown in Table 2.

At the beginning of this test, from second 0 to 180, both devices belong to the high category. During the second period, from second 180 to 360, the medium category is assigned to the "Daughter's tablet". During the third period, from second 360 to 450, the subscriber sets the low category for the "Son's smartphone" device. Finally, during the last interval, from second 540 to 720, the "Daughter's tablet" device is set



```
No.   Time           Source        Destination   Protocol  Length Info
 2043 179.212505617 10.0.103.45   192.168.0.2   OpenFlow      74 Type: OFPT_BARRIER_REQUEST
 2044 179.213808671 192.168.0.2   10.0.103.45   OpenFlow      74 Type: OFPT_BARRIER_REPLY
 2046 181.018337818 10.0.103.45   192.168.0.2   OpenFlow     536 Type: OFPT_PACKET_OUT
 2086 181.124496876 10.0.103.45   192.168.0.2   OpenFlow      98 Type: OFPT_METER_MOD
 2090 181.134674139 10.0.103.45   192.168.0.2   OpenFlow     186 Type: OFPT_FLOW_MOD
 2099 181.519248207 10.0.103.45   192.168.0.2   OpenFlow      74 Type: OFPT_BARRIER_REQUEST
```

```
▸ Frame 2090: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits) on interface 0
▸ Ethernet II, Src: IntelCor_e2:42:b2 (90:e2:ba:e2:42:b2), Dst: TelnetRe_02:e9:cd (78:3d:5b:02:e9:cd)
▸ Internet Protocol Version 4, Src: 10.0.103.45, Dst: 192.168.0.2
▸ Transmission Control Protocol, Src Port: 6633, Dst Port: 53582, Seq: 30543, Ack: 466961, Len: 120
▾ OpenFlow 1.3
    Version: 1.3 (0x04)
    Type: OFPT_FLOW_MOD (14)
    Length: 120
    Transaction ID: 3772
  ▾ Match
      Type: OFPMT_OXM (1)
      Length: 50
    ▸ OXM field
    ▸ OXM field
    ▾ OXM field
        Class: OFPXMC_OPENFLOW_BASIC (0x8000)
        0000 100. = Field: OFPXMT_OFB_ETH_SRC (4)
        .... ...0 = Has mask: False
        Length: 6
        Value: Pegatron_0b:02:1b (d4:5d:df:0b:02:1b)
    ▸ OXM field
    ▸ OXM field
      Pad: 000000000000
  ▸ Instruction
  ▾ Instruction
      Type: OFPIT_METER (6)
      Length: 8
      Meter ID: 197147
```

```
▸ Frame 2086: 98 bytes on wire (784 bits), 98 bytes captured (784 bits
▸ Ethernet II, Src: IntelCor_e2:42:b2 (90:e2:ba:e2:42:b2), Dst: Telnet
▸ Internet Protocol Version 4, Src: 10.0.103.45, Dst: 192.168.0.2
▸ Transmission Control Protocol, Src Port: 6633, Dst Port: 53582, Seq:
▾ OpenFlow 1.3
    Version: 1.3 (0x04)
    Type: OFPT_METER_MOD (29)
    Length: 32
    Transaction ID: 3771
    Command: OFPMC_ADD (0)
  ▸ Flags: 0x00000001
    Meter ID: 197147
  ▾ Meter band
      Type: OFPMBT_DROP (1)
      Length: 16
      Rate: 40960
      Burst size: 0
      Pad: 00000000
```

Fig. 10. Openflow messages of the upstream services sent to the "*Daughter's tablet*" device.

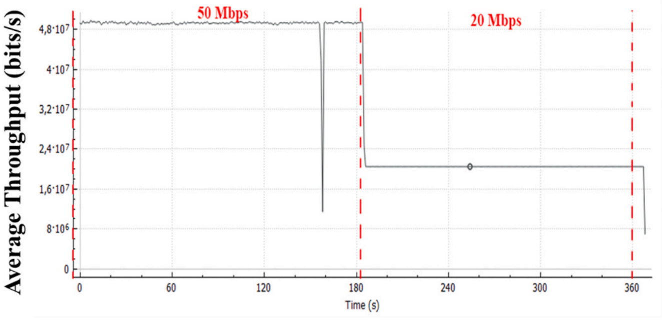


Fig. 11. Real time evolution of the allocated bandwidth to the "*Son's smartphone*" device.

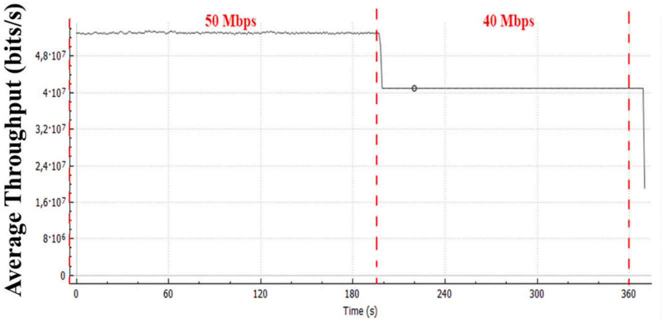


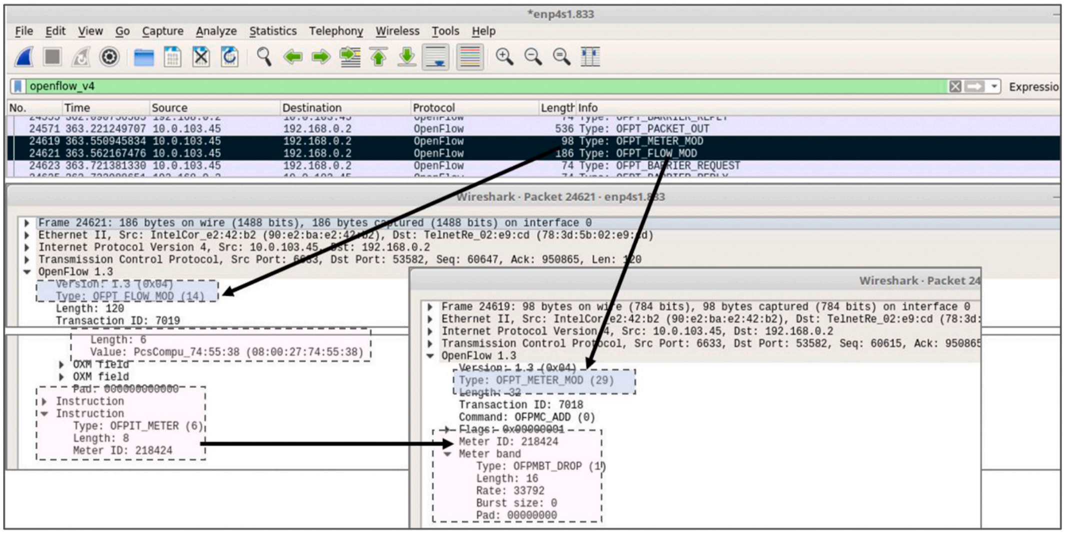
Fig. 12. Real time evolution of the allocated bandwidth to the "*Daughter's tablet*" device.

again to the high category. During the entire test both devices transmit at a peak rate of 100 Mbps in the downstream and upstream channels. Fig. 13 shows a Wireshark capture of the OpenFlow messages sent to the OVS that controls the "Son's smartphone" device when the subscriber sets the low category at the sixth minute (second 360). It can be observed that the ODL controller sends two messages to the home OVS. The first message, OFTP_FLOW_MOD, creates the meter (ID = 218424) and identifies the device to be restricted by MAC address (08:00:27:74:55:38). The second message, OFMP_METER_CONFIG, creates the 33 Mbps bandwidth restriction (rate of 33792 bps) for that meter.

Table 2
Restrictions imposed on both devices considering category levels.

Interval (sec)	Restriction level	Devices	
		Son's Smartphone	Daughter's Tablet
0–180	Category	High (no restrict.)	High (no restrict.)
	Assigned BW	50 Mbps	50 Mbps
180–360	Category	High (no restrict.)	Medium
	Assigned BW	50 Mbps	50 Mbps
360–540	Category	Low	Medium
	Assigned BW	33 Mbps	50 Mbps
540–720	Category	Low	High (no restrict.)
	Assigned BW	33 Mbps	67 Mbps

For the sake of clarity, Table 2 shows the maximum bandwidth levels that should be assigned at each interval to both devices according to their categories, whereas Figs. 14 and 15 represent the real time bandwidth evolution for both devices measured with Wireshark in the GPON testbed. In the first interval (second 0 to 180) the total bandwidth is equally shared by both (non-restricted) devices, so they both get a throughput of around 50 Mbps. In the following interval (second 180 to 360), the "Daughters tablet" is restricted to a maximum of 50 Mbps (as it now belongs to the medium category). However, since that value is not currently exceeded, the bandwidth is still approximately equally split between both devices. In the third interval (second 360 to 540) the "Son's smartphone" is set to the low category, thus getting a throughput of 33 Mbps. Since the "Daughter's tablet" still belongs to the medium category it cannot (and does not) exceed 50 Mbps (although there is still bandwidth available in the contracted rate, 100 Mbps). Finally, in the last interval (second 540 to 720), the "Son's smartphone" keep s the same category (low) while the "Daughter's tablet" is set to the high category. Since the latter device has no restrictions now, it can use all the remaining bandwidth, thereby getting a throughput of around 67 Mbps (vs. 33 Mbps for the "Son's smartphone"). As it can be observed in Figs. 14 and 15, the SDN policy is efficient as the bandwidth fluctuations



```
No.    Time           Source        Destination   Protocol   Length Info
24571 363.221249707 10.0.103.45   192.168.0.2   OpenFlow    536 Type: OFPT_PACKET_OUT
24619 363.550945834 10.0.103.45   192.168.0.2   OpenFlow     98 Type: OFPT_METER_MOD
24621 363.562167476 10.0.103.45   192.168.0.2   OpenFlow    186 Type: OFPT_FLOW_MOD
24623 363.721381330 10.0.103.45   192.168.0.2   OpenFlow     74 Type: OFPT_BARRIER_REQUEST
```

```
▸ Frame 24621: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits) on interface 0
▸ Ethernet II, Src: IntelCor_e2:42:b2 (90:e2:ba:e2:42:b2), Dst: TelnetRe_02:e9:cd (78:3d:5b:02:e9:cd)
▸ Internet Protocol Version 4, Src: 10.0.103.45, Dst: 192.168.0.2
▸ Transmission Control Protocol, Src Port: 6633, Dst Port: 53582, Seq: 60647, Ack: 950865, Len: 120
▾ OpenFlow 1.3
    Version: 1.3 (0x04)
    Type: OFPT_FLOW_MOD (14)
    Length: 120
    Transaction ID: 7019
      Length: 6
      Value: PcsCompu_74:55:38 (08:00:27:74:55:38)
  ▸ OXM field
  ▸ OXM field
    Pad: 000000000000
▸ Instruction
▾ Instruction
    Type: OFPIT_METER (6)
    Length: 8
    Meter ID: 218424
```

```
▸ Frame 24619: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▸ Ethernet II, Src: IntelCor_e2:42:b2 (90:e2:ba:e2:42:b2), Dst: TelnetRe_02:e9:cd (78:3d:
▸ Internet Protocol Version 4, Src: 10.0.103.45, Dst: 192.168.0.2
▸ Transmission Control Protocol, Src Port: 6633, Dst Port: 53582, Seq: 60615, Ack: 950865
▾ OpenFlow 1.3
    Version: 1.3 (0x04)
    Type: OFPT_METER_MOD (29)
    Length: 32
    Transaction ID: 7018
    Command: OFPMC_ADD (0)
  ▸ Flags: 0x00000001
    Meter ID: 218424
  ▾ Meter band
      Type: OFPMBT_DROP (1)
      Length: 16
      Rate: 33792
      Burst size: 0
      Pad: 00000000
```

Fig. 13. Openflow messages of the upstream services sent to the "*Son's smartphone device*".

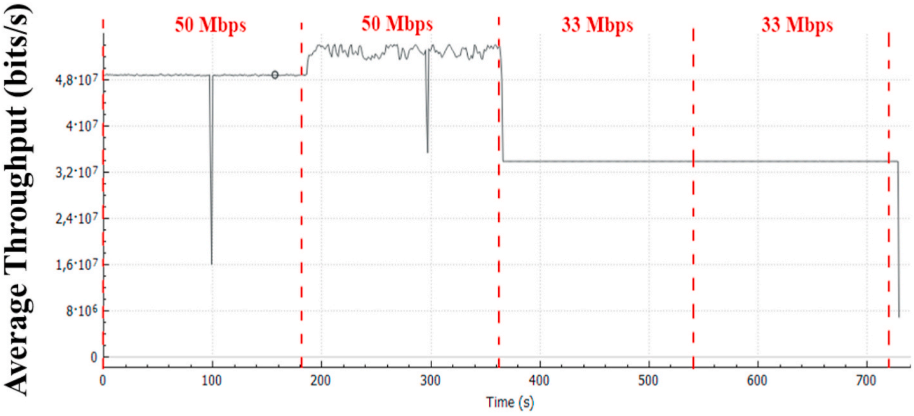


Fig. 14. Real time evolution of the allocated bandwidth of the "*Son's smartphone*" device.

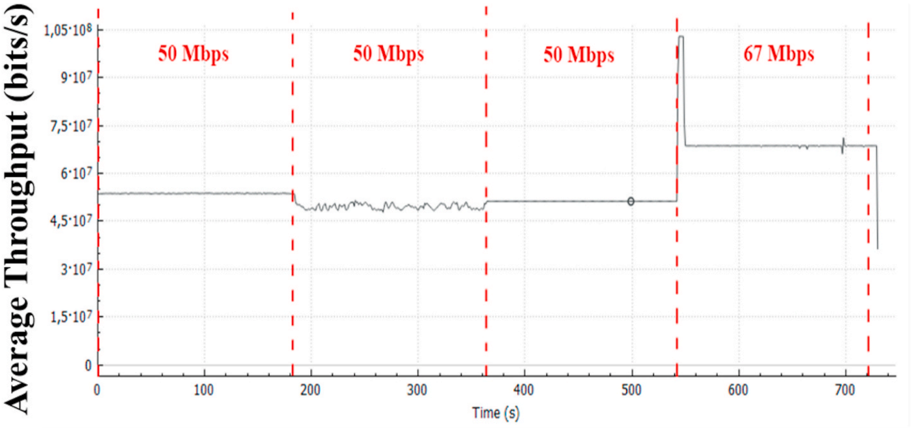


Fig. 15. Real time evolution of the allocated bandwidth of the "*Daughter's tablet*" device.

when categories are assigned are not very high and the transitions between them are fast (a very few seconds). However, it can be noticed that in some intervals, fluctuations are higher since they depend on the internal behavior the OVS when several filters are active at the same time on the flow tables depending on the imposed restrictions.

5. Improvement of the SDN solution to manage residential networks over GPONs by using NAT

A. Description of the SDN approach implemented over a GPON using NAT

In Sections 4 and 5, we have presented an approach which assumes public IP addresses assigned to each device. That approach may increase costs and security risks as well as restrict the user's freedom when choosing IP addresses for the residential devices. Therefore, in this section, we present a network scenario that avoids these issues by using SNAT (Source NAT) in the residential network of the subscriber. This approach eliminates the distributed DHCP server on the client side (ONT side), which was previously shown in Fig. 5. Then, the DHCP server located at the OLT will only provide an IP address to the OVS located at the ONT side (ROVS), which will be the gateway to the residential network. In this way, the residential network will remain private and the subscriber will be able to choose the IP addresses of their devices, as it can be observed in Fig. 16.

In the ROVS (ONT side), a SNAT translation is performed to go from the private residential network to the public network (towards the GPON). But in addition to modifying the IP address of devices, the source ports (in the case of upstream) or the destination ports (in the case of downstream) are also modified, assigning a range of associated ports to each device of the residential network. In this way, the traffic associated to every device within the residential network can be identified by means of the ports that are being used. Therefore, the ROVS will be able to filter the packets of each device through the range of ports assigned to it by SNAT. The configuration to translate the IP addresses is done through the *iptables* tool (the most widely used tool for implementing firewalls). Moreover, a script is created so that every time a new device is connected, it automatically assigns a unique port range associated with this device and saves this information in an internal database together with its MAC address and its private IP address (inside the residential network). Finally, the last step will be to create services for that device, which will be differentiated by ports. Therefore, a different flow needs to be created for each associated service for each port.

B. Validation analysis and results

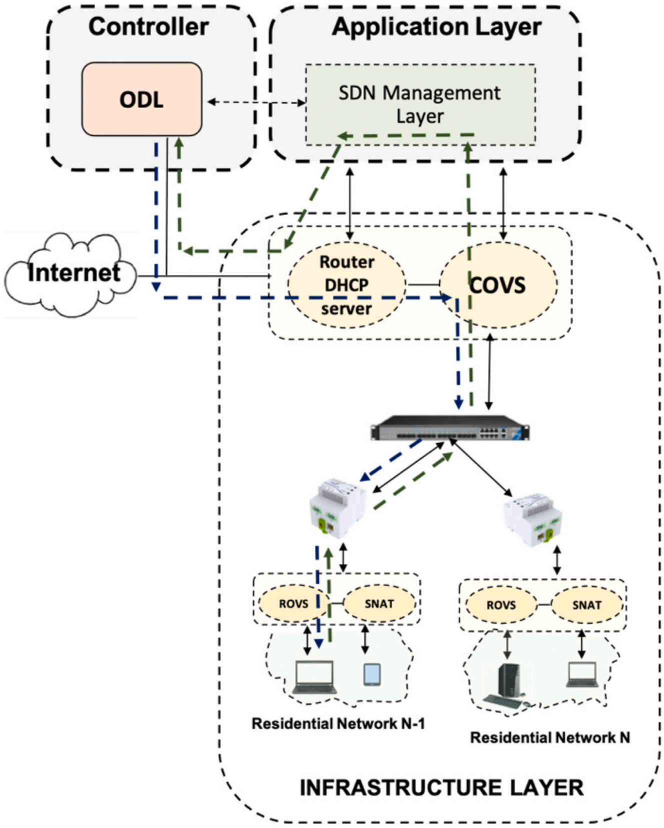


Fig. 16. SDN approach over legacy GPONs using SNAT.

To demonstrate this approach, we have set up an experiment on the GPON testbed. The initial network environment is modified, so that the users now remain within a private network using SNAT. The configuration is shown in Fig. 17. In this scenario, we consider the same devices as before, that is, "Son's smartphone" and "Daughter's tablet". All residential devices belong to the 172.16.10.0/24 subnet. For instance, the "Son's smartphone" is assigned 172.16.10.2 and the "Daughter's tablet" 172.16.10.3. Every device of the residential network communicates with the GPON by means of a single public IP address, which will be the IP address given to the ROVS (192.168.1.148).

Fig. 18 shows the configuration of the associated ROVS, in which the *iptables* tool redirects the traffic to the ROVS. Then, SNAT is configured through *iptables*, so that the residential network is connected to the ROVS. As can be seen in Fig. 18, the "Son's smartphone" device (internal IP 172.16.10.2) has the port range 5000–5019 when the destination is the upstream channel (traffic goes from the residential network to the GPON) and therefore the destination will be the ROVS (IP 192.168.1.148). The same performance can be noticed in Fig. 18 for the "Daughter's tablet" device (internal IP 172.16.10.3), but with a different range of ports, that is, 5020–5029.

With this configuration, we have carried out an experimental test in the GPON testbed with both devices ("Son's smartphone" and "Daughter's tablet") imposing bandwidth restrictions to control their bandwidth consumption. We have set a maximum bandwidth of 100 Mbps for the "Son's smartphone" device and 50 Mbps for the "Daughter's tablet" device. Then, we have created two new flows and two new meters, one for each device, as shown in Fig. 19. In particular, we have created a flow for the "Son's smartphone" device with a match that corresponds to

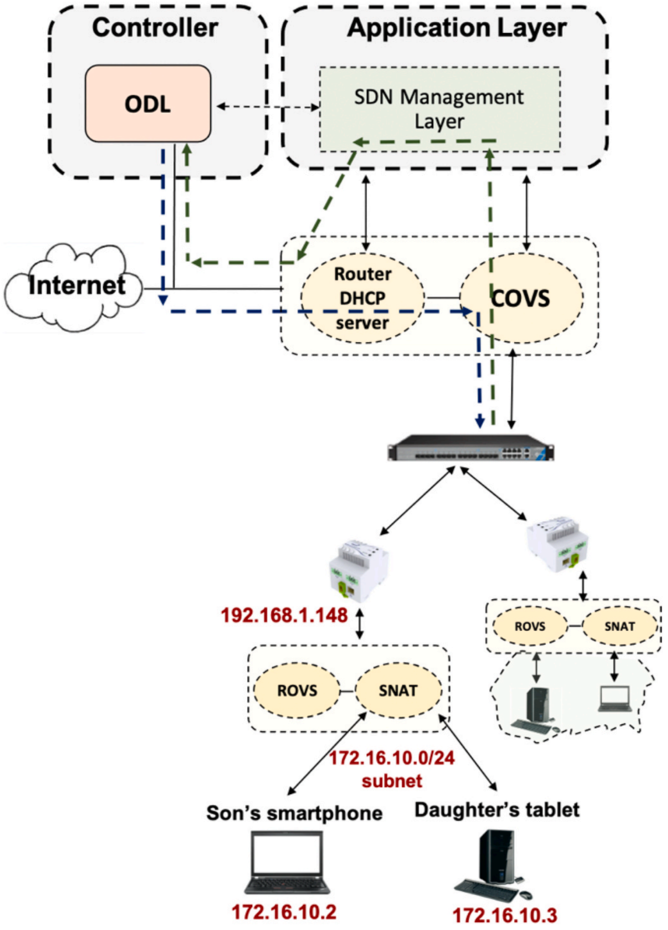
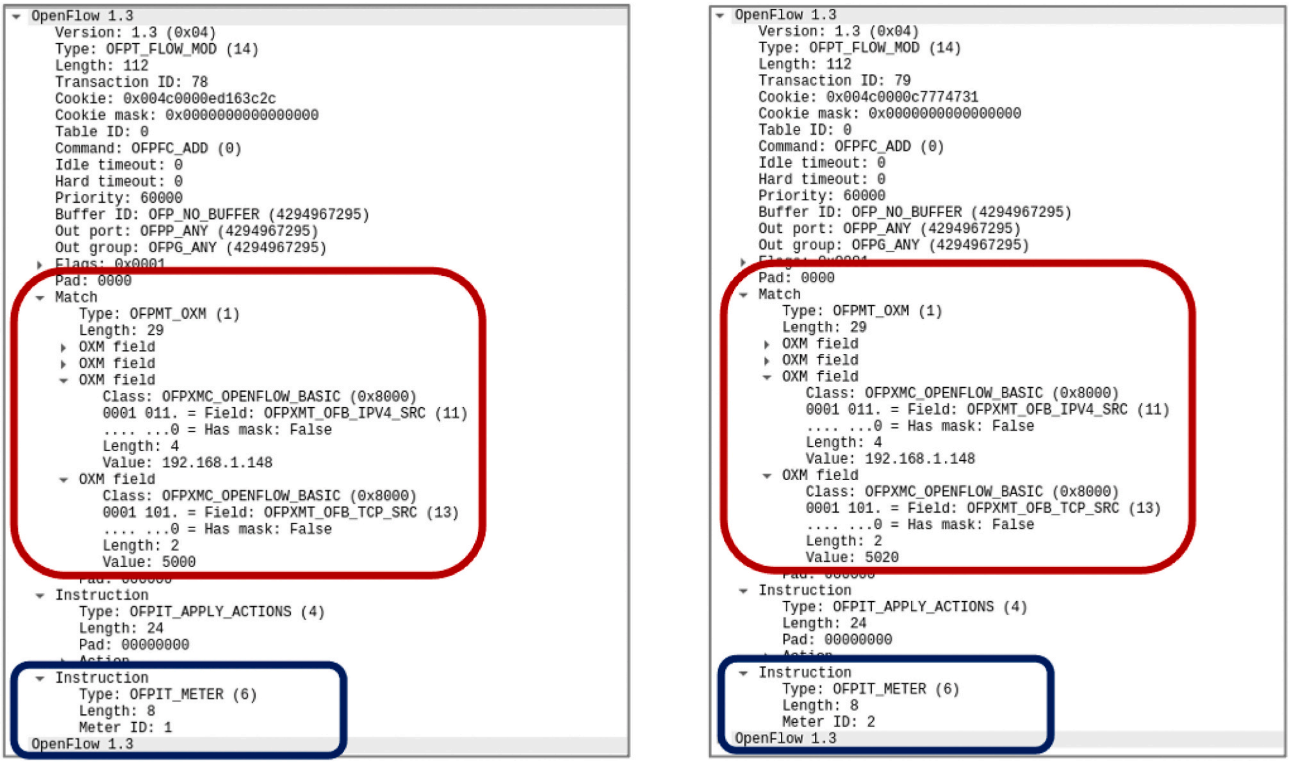


Fig. 17. Experiment with public and private IP addresses in the new SDN approach over legacy GPONs using SNAT.

```
Chain POSTROUTING (policy ACCEPT)
target     prot opt source          destination
SNAT       udp  --  172.16.10.3     anywhere              to:192.168.1.148:5020-5029
SNAT       tcp  --  172.16.10.3     anywhere              to:192.168.1.148:5020-5029
SNAT       udp  --  172.16.10.2     anywhere              to:192.168.1.148:5000-5019
SNAT       tcp  --  172.16.10.2     anywhere              to:192.168.1.148:5000-5019
MASQUERADE  all  --  172.16.10.0/24  anywhere
```

Fig. 18. Configuration made in the OVS with iptables to apply SNAT.



```
OpenFlow 1.3
  Version: 1.3 (0x04)
  Type: OFPT_FLOW_MOD (14)
  Length: 112
  Transaction ID: 78
  Cookie: 0x004c0000ed163c2c
  Cookie mask: 0x0000000000000000
  Table ID: 0
  Command: OFPFC_ADD (0)
  Idle timeout: 0
  Hard timeout: 0
  Priority: 60000
  Buffer ID: OFP_NO_BUFFER (4294967295)
  Out port: OFPP_ANY (4294967295)
  Out group: OFPG_ANY (4294967295)
  Flags: 0x0001
  Pad: 0000
  Match
    Type: OFPMT_OXM (1)
    Length: 29
    OXM field
    OXM field
    OXM field
      Class: OFPXMC_OPENFLOW_BASIC (0x8000)
      0001 011. = Field: OFPXMT_OFB_IPV4_SRC (11)
      .... ...0 = Has mask: False
      Length: 4
      Value: 192.168.1.148
    OXM field
      Class: OFPXMC_OPENFLOW_BASIC (0x8000)
      0001 101. = Field: OFPXMT_OFB_TCP_SRC (13)
      .... ...0 = Has mask: False
      Length: 2
      Value: 5000
    Pad: 000000
  Instruction
    Type: OFPIT_APPLY_ACTIONS (4)
    Length: 24
    Pad: 00000000
    Action
  Instruction
    Type: OFPIT_METER (6)
    Length: 8
    Meter ID: 1
OpenFlow 1.3
```

```
OpenFlow 1.3
  Version: 1.3 (0x04)
  Type: OFPT_FLOW_MOD (14)
  Length: 112
  Transaction ID: 79
  Cookie: 0x004c0000c7774731
  Cookie mask: 0x0000000000000000
  Table ID: 0
  Command: OFPFC_ADD (0)
  Idle timeout: 0
  Hard timeout: 0
  Priority: 60000
  Buffer ID: OFP_NO_BUFFER (4294967295)
  Out port: OFPP_ANY (4294967295)
  Out group: OFPG_ANY (4294967295)
  Flags: 0x0001
  Pad: 0000
  Match
    Type: OFPMT_OXM (1)
    Length: 29
    OXM field
    OXM field
    OXM field
      Class: OFPXMC_OPENFLOW_BASIC (0x8000)
      0001 011. = Field: OFPXMT_OFB_IPV4_SRC (11)
      .... ...0 = Has mask: False
      Length: 4
      Value: 192.168.1.148
    OXM field
      Class: OFPXMC_OPENFLOW_BASIC (0x8000)
      0001 101. = Field: OFPXMT_OFB_TCP_SRC (13)
      .... ...0 = Has mask: False
      Length: 2
      Value: 5020
    Pad: 000000
  Instruction
    Type: OFPIT_APPLY_ACTIONS (4)
    Length: 24
    Pad: 00000000
    Action
  Instruction
    Type: OFPIT_METER (6)
    Length: 8
    Meter ID: 2
OpenFlow 1.3
```

Fig. 19. OpenFlow messages to create a flow por each device (OFPT_FLOW_MOD).

the external IP of 192.168.1.148 (ROVS) and port 5000, and we have associated a meter with ID 1 (which internally is configured with 100 Mbps). Besides, for the “Daughter’s tablet” device the match corresponds with the same IP address and the port 5020 and we have associated it the meter with ID 2 (which internally corresponds to 50 Mbps).

Finally, Figs. 20 and 21 show the real time evolution of the assigned bandwidth for both devices (“Son’s smartphone” and “Daughter’s tablet”) using Wireshark. It can be observed in both graphs that the imposed bandwidth restrictions are efficiently applied for both devices using the new implemented approach and the response and fluctuations are quite stable.

6. Summary

We have proposed and experimentally integrated an SDN solution implemented over legacy GPONs, which permits residential users to control their home bandwidth resources, efficiently and dynamically, using a very easy-to-use web interface. Indeed, our SDN residential

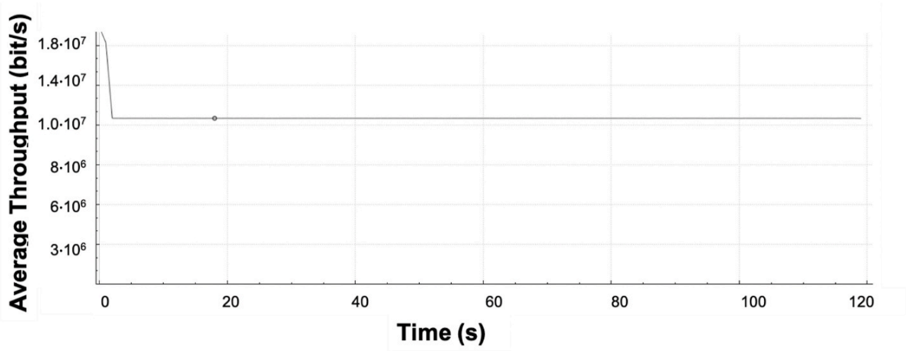


Fig. 20. Real-time evolution of the bandwidth assigned to the “Son’s smartphone” device with a maximum bandwidth restriction of 100 Mbps.

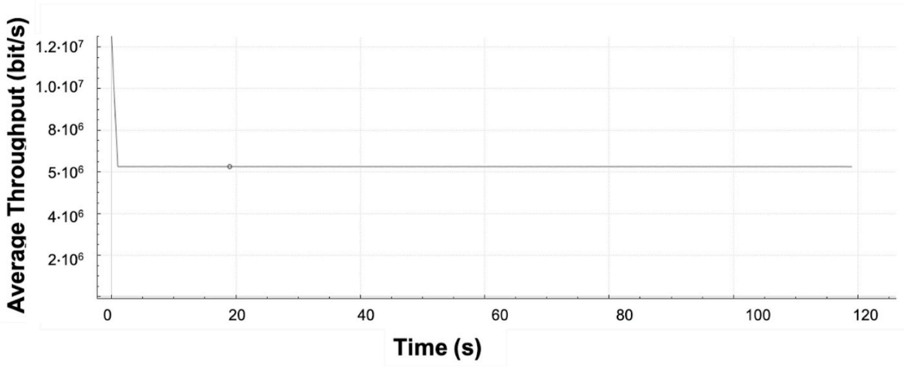


Fig. 21. Real-time evolution of the bandwidth assigned to the "Daughter's tablet" device with a maximum bandwidth restriction of 50 Mbps.

network management proposal permits users to adjust their bandwidth demands by paying for the network use in a dynamic way. Furthermore, residential users have freedom to imposed bandwidth restrictions to devices connected to the residential network. This solution can enable new business models for network operators and service providers in which users can control their contracted bandwidth in a very easy and fast way by means of SDN techniques. Indeed, this solution could be interesting for hotels or university residence halls willing to offer differentiated services in their residential networks to their guests.

The proposal has been experimentally validated in a GPON testbed using OpenFlow and the OpenDayLight (ODL) controller, and the results show a very efficient and fast performance of the implemented strategy. In fact, when users demand a temporary increase of their bandwidth conditions, it has been shown that the ODL controller executes the command in a very fast way (around 2 s) and the transition to the new bandwidth levels keeps stable. On the other hand, residential users can efficiently manage their bandwidth resources in a very transparent way. Then, it has been shown that when the residential user imposes a bandwidth restriction on any device connected to the residential network, the whole system configures itself quickly, and the SDN policy is kept efficiently as the transitions and the fluctuations to the new bandwidth levels when categories are imposed along the time are quite fast and low, respectively.

Finally, the SDN network management approach over GPONs presented in this paper allows subscribers to easily interact with their residential networks, promoting a better user QoS experience by means of SDN techniques, and enabling at the same time new business models for network operators and service providers. Even more, the proposal avoids modifying the GPON layer, since the GPON traffic and QoS requirements of network subscribers are controlled by the SDN solution in layer 3. Consequently, a generalization of the SDN environment proposal for network operators can be possible without changing the whole architecture. Furthermore, although an abstraction layer has to be deployed in the PON devices (OLT, ONTs), the proposal brings several advantages for Internet Providers and network operators. On the one hand, even though legacy PON devices fulfil the GPON standard, sometimes the management software provided by manufactures to configure the network and its subscribers does not allow interoperability among devices of different manufacturers. In this way, our OpenFlow proposal permits to control legacy GPON equipment of different vendors since every device (OLT, ONT) understands the OpenFlow protocol, thanks to the emulated SDN layer implemented on each of them. On the other hand, our proposal permits to simultaneously control several PONs using a single SDN controller since it can be able to send OpenFlow instructions to different PONs to configure them in an efficient way.

Authorship statement

Noemi Merayo, Conception and design of study, acquisition of data, analysis and/or interpretation of data, revising the manuscript critically for important intellectual content, Approval of the version of the manuscript to be published. David de Pintos, Conception and design of study, analysis and/or interpretation of data, revising the manuscript critically for important intellectual content, Approval of the version of the manuscript to be published. Juan C. Aguado, acquisition of data, revising the manuscript critically for important intellectual content, Approval of the version of the manuscript to be published. Ignacio de Miguel, Conception and design of study, revising the manuscript critically for important intellectual content, Approval of the version of the manuscript to be published. Ramón J. Durán, analysis and/or interpretation of data, revising the manuscript critically for important intellectual content, Approval of the version of the manuscript to be published. Patricia Fernández, acquisition of data, Drafting the manuscript, Approval of the version of the manuscript to be published. Rubén M. Lorenzo, Drafting the manuscript, Approval of the version of the manuscript to be published. Evaristo J. Abril, Drafting the manuscript, Approval of the version of the manuscript to be published.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We would like to thank the anonymous reviewers for their insighful comments and suggestions, which helped to improve our proposal. This work has been funded by Consejería de Educación de la Junta de Castilla y León (VA085G19), the Spanish Ministry of Science, Innovation and Universities (TEC2017-84423-C3-1-P, RED2018-102585-T) and the INTERREG V-A España-Portugal (POCTEP) program (0677_DISRUPTIVE_2_E).

References

[1] K.J. Kerpez, J.M. Cioffi, G. Ginis, M. Goldburg, S. Galli, P. Silverman, Software-defined access networks, IEEE Commun. Mag. 52 (2014) 152–159.
[2] Open Networking Foundation (ONF), OpenFlow, URL, https://www.opennetworking.org/.
[3] NETCONF Configuration Protocol, Network Configuration Working Group, URL, https://tools.ietf.org/wg/netconf.
[4] RESTCONF protocol, internet engineering task force (IETF), URL, https://tools.ietf.org/html/rfc8040.
[5] A.S. Thyagaturu, A. Mercian, M.P. McGarry, M. Reinsslein, W. Kellener, Software defined optical networks (SDONs): a comprehensive survey, IEEE Commun. Surv. Tutorials 18 (2016) 2738–2786.
[6] Europe Broadband Market at September 2018, FTTH Council, URL, https://www.ftthcouncil.eu/documents/FTTH%20Council%20Europe%20%20Panorama%20at%20September%202018.pdf.

[7] Europe Broadband status market forecast by 20202025, FTTH Council, URL, https://www.ftthcouncil.eu/documents/Reports/2019/FTTH%20Council%20Europe%20%20Forecast%20for%20EUROPE%202020-2025.pdf.

[8] A.M. Alshnta, M.F. Abdollah, A. Al-Haiqi, SDN in the home: a survey of home network solutions using Software Defined Networking, J. Cogent Eng. 5 (2018) 1–8.

[9] S. Kim, S. Sundaresan, M. Chetty, N. Feamster, W.K. Edwards, Communicating with caps: managing usage caps in home networks, in: ACM SIGCOMM 2011 Conference, 2011, pp. 470–471.

[10] M. Chetty, H. Kim, S. Sundaresan, S. Burnett, N. Feamster, W.K. Edwards, uCap: an internet data management tool for the home, in: 33rd Annual ACM Conference on Human Factors in Computing Systems, 2015, pp. 3093–3102.

[11] Y. Yiakoumis, S. Katti, T.-Y. Huang, N. McKeown, K. Yap, R. Johari, Putting home users in charge of their network, in: ACM Conference on Ubiquitous Computing, 2012, pp. 1114–1119.

[12] H. Gharakheili, L. Exton, V. Sivaraman, Managing home routers from the cloud using software defined networking, in: 13th IEEE Annual Consumer Communications & Networking Conference (CCNC), 2016, pp. 1–2.

[13] H. Gharakheili, L. Exton, V. Sivaraman, J. Matthews, C. Russell, Third-party customization of residential Internet sharing using SDN, in: International Telecommunication Networks and Applications Conference, 2015, pp. 214–219, https://doi.org/10.1109/ATNAC.2015.7366815.

[14] R. Mortier, T. Rodden, T. Lodge, D. McAuley, C. Rotsos, A. Moore, A. Koliousis, J. Sventek, Control and understanding: owning your home network, in: Communication Systems and Networks, 2012, pp. 1–10, https://doi.org/10.1109/COMSNETS.2012.6151322.

[15] NOX SDN controller. https://github.com/noxrepo.

[16] M. Amiri, H. Al Osman, S. Shimohammdi, SDN-enabled game-aware network management for residential gateways, in: IEEE International Symposiym on Multimedia, 2017, pp. 330–333, https://doi.org/10.1109/ISM.2017.60.

[17] Mininet, An instant virtual network on your laptop (or other PC) - Mininet, Mininet. http://mininet.org/.

[18] Y. Yiakoumis, K.-K. Yap, S. Katti, G. Parulkar, N. McKeown, Slicing home networks, in: 2nd ACM SIGCOMM Workshop on Home Networks, 2011, pp. 1–6, https://doi.org/10.1145/2018567.2018569.

[19] R. Challa, S.M. Raza, S. Jeon, H. Choo, P. Thorat, Refactoring future residential networks into slices, in: IEEE Conference on Network Function Virtualization and Softwared Defined Networks, 2018, pp. 1–6, https://doi.org/10.1109/NFV-SDN.2018.8725676.

[20] S. Chaabnia, A. Meddeb, Slicing aware QoS/QoE in software defined smart home network, in: IEEE/IFIP Network Operations and Management Symposium, 2018, pp. 1–5, https://doi.org/10.1109/NOMS.2018.8406195.

[21] S.S.W. Lee, K.-Y. Li, M.-S. Wu, Design and implementation of a GPON-based virtual OpenFlow-enabled SDN switch, IEEE/OSA J. Lightwav. Technol. 34 (2016) 2552–2561.

[22] S.S.W. Lee, K. Li, W. Liu, C. Chen, H. Fang, T. Wong, Embedding bandwidth-guaranteed network-based virtual Ethernet switches in SDN networks, IEEE/OSA J. Lightwav. Technol. 35 (2017) 5041–5055.

[23] R.G. Clegg, J. Spencer, R. Landa, M. Thakur, J. Mitchell, M. Rio, Pushing software defined networking to the access, in: Software Defined Networks Third European Workshop on, 2014, pp. 31–36, https://doi.org/10.1109/EWSDN.2014.11.

[24] R. Gu, Y. Ji, P. Wei, S. Zhang, Software defined flexible and efficient passive optical networks for intra-datacenter Communications, IEEE/OSA Opt. Switch. Netw. 14 (2014) 289–302.

[25] A. Amokrane, J. Hwang, J. Xiao, N. Anerousis, Software defined enterprise passive optical network, in: IEEE Int. Conf. Network Service Management, 2014, pp. 406–411, https://doi.org/10.1109/CNSM.2014.7014203.

[26] P. Parol, M. Pawlowski, Future proof access networks for B2B applications, Informatica 38 (2014) 193–204.

[27] H. Khalili, D. Rincón, S. Sallent, J.R. Piney, An integrated SDN-based architecture for passive optical networks, in: Broadband Communications Networks: Recent Advances and Lessons from Practice, InTech, London, 2018, pp. 215–237.

[28] C. Qian, Y. Li, Q. Zhang, B. Cao, Z. Li, M. Wang, Staged priority-based dynamic bandwidth allocation in software-defined hybrid passive optical network, Opt. Eng. 57 (2018) 1–8.

[29] W. Xia, C. Gan, W. Xie, S. Ma, Bandwidth allocated mechanism and its algorithm for multi-subsystem-based virtual passive optical network in metro-access optical network, IET Commun. 10 (2016) 154–161.

[30] N. Afraz, M. Rufini, A sharing platform for Multi-Tenant PONs, IEEE/OSA J. Lightwav. Technol. 6 (2018) 5413–5423.

[31] S. Yin, T.S. Shen, Y. Bi, J. Jin, T. Oyama, L.G. Kazovsky, A novel quasi-passive, software-defined, and energy efficient optical access network for adaptive intra-PON flow transmission, IEEE/OSA J. Lightwav. Technol. 33 (2015) 4536–4546.

[32] A.F. Pakpahan, I. Hwang, A.A. Nikoukar, OLT energy savings via software-defined dynamic resource provisioning in TWDM-PONs, IEEE/OSA J. Opt. Commun. Netw. 9 (2017) 1019–1029.

[33] S. McGettrick, F. Slyne, N. Kitsuwan, D.B. Payne, M. Ruffini, Experimental end-to-end demonstration of shared N:M dual-homed protection in SDN-controlled long-reach PON and pan-European core, IEEE/OSA J. Lightwav. Technol. 34 (2018) 4205–4213.

[34] The Open Source Project VOLTHA, URL, https://www.opennetworking.org/voltha/.

[35] OpenOLT agent, URL, https://guide.opencord.org/openolt/.

[36] S.E.B.A. OpenNetworking, URL, https://opennetworking.org/seba/.

[37] OpenSource Access Manager (OSAM) Use Case, URL, https://wiki.onap.org/display/DW/OpenSource+Access+Manager+%28OSAM%29+Use+Case.

[38] Open Network Automation Platform (ONAP), URL, https://www.onap.org/.

[39] R. Flores Moyano, D. Fernández, L. Bellido, C. González, A software-defined networking approach to improve service provision in residential networks, Int. J. Netw. Manag. 27 (2017), e1984.

[40] R. Flores Moyano, D. Fernández, N. Merayo, C.M. Lentisco, A. Cárdenas, NFV and SDN-based differentiated traffic treatment for residential networks, IEEE Access 8 (2020) 34038–34055.

[41] J. Azofra, N. Merayo, J.C. Aguado, I. de Miguel, R. Durán, F.J. Ruíz, C. Vega, D. Fernández, P. Fernández, R.M. Lorenzo, E.J. Abril, Implementation of a testbed to analysis am SDN based GPON, in: 2018 European Conference on Optical Communication, 2018, pp. 1–3, https://doi.org/10.1109/ECOC.2018.8535516.

[42] Open vSwitch, URL, https://www.openvswitch.org/.

[43] OpenDayLight SDN controller, URL, https://www.opendaylight.org/.

[44] Onos Open source controller (ONOS), URL, https://www.onosproject.org/.

[45] OpenFlow Switch Specification and meters, OpenFlow switch specification, URL, https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf.

[46] J.A. Hatem, A.R. Dhani, S. Elbassuoni, Deep learning-based dynamic bandwidth allocation for future optical access networks, IEEE Access 7 (2019) 97307–97318.

[47] P. Kourtessis, W. Lim, N. Merayo, Y.-M. Yang, John M. Senior, Efficient T-CONT-agnostic bandwidth and wavelength Allocation for NG-PON2, IEEE/OSA J. Opt. Commun. Netw. 11 (2019) 383–396.

[48] Telnet vendor web page, Telnet R-I, URL, https://www.telnet-ri.es/.

[49] Wireshark: Network Protocol Analyzer, URL, https://www.wireshark.org/.

[50] P.K. Sharma, J.H. Park, Y.-S. Jeong, J.H. Park, SHSec: SDN based secure smart home network architecture for internet of Things, Mobile Network. Appl. 24 (2019) 913–924.