Deployment of an SDN-based GPON control agent to manage network configurations

Noemí Merayo
Universidad de Valladolid
Valladolid, Spain
0000-0002-6920-0778

Juan Carlos Aguado
Universidad de Valladolid
Valladolid, Spain
0000-0002-2495-0313

Ignacio de Miguel
Universidad de Valladolid
Valladolid, Spain
0000-0002-1084-1159

Ramón J. Durán Barroso
Universidad de Valladolid
Valladolid, Spain
0000-0003-1423-1646

Patricia Fernández
Universidad de Valladolid
Valladolid, Spain
0000-0001-5520-0948

Rubén M. Lorenzo
Universidad de Valladolid
Valladolid, Spain
0000-0001-8729-3085

Evaristo J. Abril
Universidad de Valladolid
Valladolid, Spain
0000-0003-4164-2467

***Abstract*—The current wide deployment of PON (Passive Optical Network) technologies implies that there is a wide variety of PON devices causing incompatibility problems when configuring and controlling them simultaneously. Although these devices follow the same standards, each manufacturer adds its own implementations, so there are no universal devices. SDN (Software Defined Networking) is a technology for dealing with these problem, since equipment from different manufacturers can be managed simultaneously, as SDN separates the control layer from the data layer. This paper proposes the development of an SDN agent based on the OpenFlow protocol to enable and facilitate the management of PONs, as well as being compatible with legacy PONs. In fact, this SDN agent allows to translate the OpenFlow commands sent by the SDN controller into the native language of PON devices.**

Keywords— Software Defined Networking, Passive Optical Networks, OpenFlow, Agent, Network Management.

I. Introduction

Passive Optical Networks (PONs) have become the most widely deployed access technology worldwide today. In fact, in the European Union 160 million households are connected by FTTH/B (Fibre To The Home/Building) technology. However, future predictions foresee a greater increase in the next five years, reaching 263 millions in 2025 [1]. Under this scenario, PON infrastructures are leading this market, as it is expected that the 73% of FTTH infrastructures will be PONs. Indeed, the worldwide market of GPONs (PONs based on the Gigabit standard) is expected to reach USD 8.3 billion by 2025. On the other hand, there is an explosion of heterogeneous connected devices and of highly demanded services with critical Quality of Service (QoS) requirements, and so networks are suffering an extreme increase in their traffic. Therefore, it is highly necessary to provide next generation networks with intelligence, security and efficiency in its management and control. Software Defined Networking (SDN) can face these challenges and, thus, its integration in PON technologies is becoming increasingly important [2].

In SDN, the network control comes off the hardware and is given to a software application called SDN controller, so that the network can be intelligently and centrally controlled, using software applications and specific protocols, such as OpenFlow [3]. In order to provide this intelligent and centralized management, SDN separates the control plane of the network from the data plane that forwards the traffic through network devices. As a consequence, operators can manage, supervise and update the entire network regardless of the underlying network technology. Moreover, SDN provides additional advantages in access networks, such as PONs. For example, it permits a fast control and reconfiguration of services and applications of residential subscribers. Furthermore, SDN can support the coexistence of PON equipment from multiple vendors, as it provides a common programmable interface.

As a consequence, many SDN proposals are focusing on PONs. For example, there are many studies about turning legacy PON devices, OLTs (Optical Line Terminals) and ONTs (Optical Network Units), into SDN-controllable devices [4]. Other research studies are working on moving dynamic bandwidth allocation (DBA) and service configuration policies out of the PON intelligence to an external SDN controller which manages this network configuration [5]. Besides, as residential networks are becoming very complex due to the large number of connected devices and the increasing number of applications with stringent QoS requirements, many proposals focus on SDN management techniques in residential networks [6]. Finally, other studies propose to develop virtual PONs [7] with SDN.

On the other hand, the wide deployment of PON architectures in the last mile means that there is a wide variety of manufacturers of PON optical devices, and that incompatibility problems arise between these devices when controlling their configuration simultaneously. Although these devices follow the same PON standards, each manufacturer adds its own implementations, so there are no universal PON devices. However, SDN technologies are promising in the field of next-generation optical networks and, in particular in PON networks, because SDN protocols can be used to simultaneously manage PON equipment from different manufacturers, as SDN proposes the separation of the control layer from the data layer.

Indeed, in previous works [8, 9] we have proposed and experimentally demonstrated the integration of SDN over legacy GPONs. For that aim, we integrated an abstract layer in GPON devices (OLT, ONTs) to emulate the performance of OpenFlow switches (OVS, Open Virtual Switches) [8, 9]. This approach implies having a virtual layer programmed on an external device (a Raspberry Pi, a Banana Pi, or a mini-computer) on each OLT and each ONT, which leads to greater complexity. In contrast, in this paper, a novel SDN agent is proposed and demonstrated. It has been developed using the well-known OpenFlow protocol, and translates the OpenFlow commands sent by the SDN controller into the native language of the PON devices (OLTs) without requiring the use of all those additional external hardware devices.

Therefore, this mechanism enables to manage and configure the network in a more efficient way. This new approach is closer to the VOLTHA (Virtual OLT Hardware Abstraction) open source project of the CORD initiative [10], which abstracts the PON network as a programmable Ethernet switch to be controlled by an SDN controller, and which interacts with PON hardware devices using vendor-specific protocols through OLT and ONU adapters by means of the OpenOLT agent. However, the OpenOLT agent uses Broadcom's Broadband Adaptation Layer (BAL) software for interfacing with chipsets in OLTs, and part of this software is not open source. Thus, some proprietary source code is required to build the drivers.

In summary, we have deployed a testbed combining SDN technologies and GPON networks, and we have developed an SDN agent using OpenFlow that allows the management of PON networks and is compatible with commercial GPON equipment. The remaining of this paper is structured as follows. Section II presents the design of the SDN agent developed for GPONs and Section III introduces the programming of the OpenFlow agent based on the OpenFlow standard. Then, Section IV describes the validation and results of the experimental testbed. Finally, Section V summarizes the main conclusions.

II. Design of an SDN Agent based on OpenFlow for GPONs

The implemented SDN-GPON scenario allows operators/ISPs (Internet Service Providers) to configure a legacy GPON using an SDN controller (ONOS [11]) and OpenFlow switches based on Open Virtual Switches (OVS) [12]. The proposal includes an SDN agent (located on the OLT side) which captures the messages coming from the ONOS controller and gives them a response as if it were an OpenFlow virtual switch. Moreover, the agent also captures the flows sent by ONOS containing OpenFlow language commands to configure the GPON network and the end-user services/profiles (connected to the ONTs/ONUs) through the OLT (which is the main management device of a GPON network). In this way, the OpenFlow agent extracts the configuration parameters sent to it by ONOS by means of OpenFlow messages and interacts with the GPON API (Application Programming Interface) in its language. The API of the GPON creates the services (e.g., Internet, voice, video) contracted by the users. Therefore, the agent is capable of transforming commands sent by the SDN controller using OpenFlow into the internal language of each GPON device. The only change when managing GPON/PON equipment from different manufacturers would be the internal API of each device, which is dependent on the vendor. Thus, it would be necessary to program an adaptation layer between the OpenFlow-based agent and the particular management API of each device. The most important feature of the agent is that it can be deployed at software level in any OLT from any manufacturer, as it is a software layer implemented as a bridge between an SDN controller and a GPON/PON network. In this way, the same SDN controller (ONOS in this case) could control PON equipment from different manufacturers using the same OpenFlow agent. Figure 1 shows a flexible network scenario with the proposed OpenFlow agent, where the same SDN controller (ONOS) can interact with different PON networks (each with equipment from different manufacturers). For that aim, the OpenFlow agent interacts with the API layer of each particular OLT, i.e., with the management API of each PON network. For the programming of the OpenFlow agent, the python-openflow library, developed by Kytos [13], has been used, which allows the creation of OpenFlow messages (version 1.3/1.0).

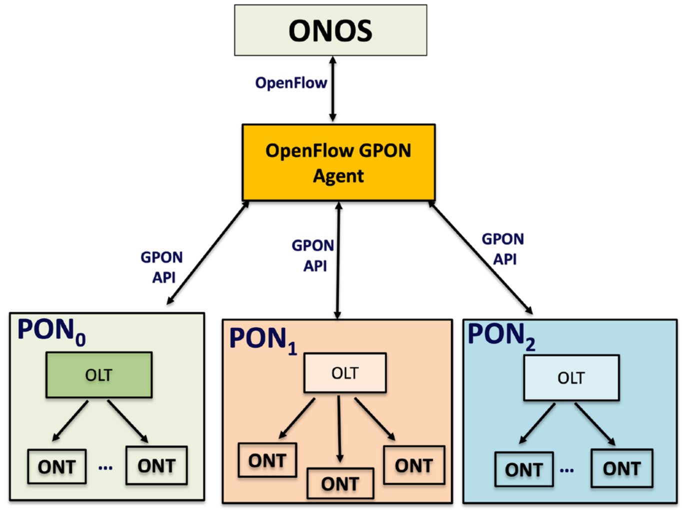

Fig. 1: OpenFlow agent to manage PONs with equipment from different vendors.

III. Communication of the OpenFlow Agent with the SDN Controller and the PON

We now explain how the OpenFlow agent interacts, on the one hand, with a central SDN controller, ONOS, and on the other hand, with the API of a commercial GPON OLT.

A. Communication between the SDN controller and the OpenFlow agent

The first step is to establish an initial communication between the SDN controller (ONOS) and the OpenFlow agent. When an OpenFlow switch connects to an SDN controller, a set of OpenFlow messages are exchanged, as it can be observed in Fig. 2. Wireshark has been used to capture this message sequence. The proposed OpenFlow agent emulates the behaviour of an OpenFlow switch, creating real OpenFlow messages (version 1.3/1.0), and exchanging them with the ONOS controller.

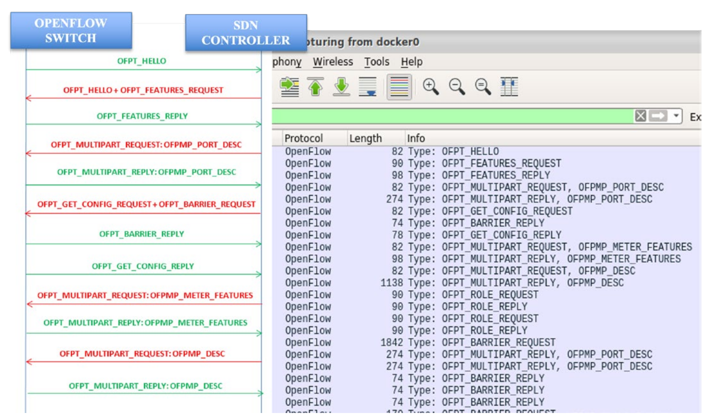

Fig. 2: Initial communication between the OpenFlow switch and ONOS

Once the initial communication is established, the OpenFlow agent responds to ONOS requests by means of OpenFlow messages, depending on the type of information requested. Specifically, the most important messages interchanged by the controller and the SDN agent to configure, update or delete services (e.g., Internet, voice, video) are as follows:

- OFPT_FLOW_MOD: the SDN controller sends this message to add or to delete a flow. The flows are used to configure a service on a specific user (ONT) defined by its specific MAC or IP address.

- OFPT_METER_MOD: Meters are used to associate a maximum bandwidth to a service. In our proposal, the QoS requirements are related to the maximum bandwidth levels, which are implemented with the help of OpenFlow meters that are associated with flows. A meter measures the rate of packets assigned to a specific flow and allows to control its rate, so when this rate is higher than the value defined in the meter, the packets are discarded. Thus, a meter defines a rate limiting band, i.e., the maximum bandwidth.

B. *Communication between the OpenFlow agent and the PON*

At this point, the OpenFlow agent is able to receive the ONOS flows and meters and store them in lists with the characteristics of the services to be configured in the GPON network. So, the next step is to connect the OpenFlow agent to the GPON network, in order to configure the services (voice, data, video) in their native language. Communication with the GPON network is done through an API that interacts with the access network using the PON standard commands to configure the network and its services. This API defines the following functions:

- *connect()*: it establishes a connection to the command line interface (CLI) of the OLT.
- *get_Port()*: it gets the port number of the OLT where the ONTs are connected.
- *get_ID_ONU()*: it returns a list with the MAC addresses of the ONTs in the GPON.
- *service_Internet()*: it configures an Internet service in the GPON.
- *video_Internet()*: it configures a video service in the GPON.
- *delete_configuration()*: it deletes a service.

IV. Validation of the SDN-GPON Testbed

To demonstrate the feasibility of the proposal, we describe an experimental test by creating an Internet service on one ONT (users) connected to our GPON setup (Fig. 3). The GPON infrastructure of the laboratory is configured with equipment from the manufacturer Telnet-RI [14]. In particular, we use the SmartOLT 350, which in a production network would be located at the central office of the operator/service provider, and which complies with the ITU-T G.984.x/G.988 recommendations and implements a GPON interface of 2.488 Gbps in the downstream channel and 1.244 Gbps in the upstream. Moreover, the GPON testbed is equipped with level 3 model ONTs (Wave Access 3021), which integrate router functionalities, and Level 2 ONTs (Wave Access 512), i.e., without router functionalities. For this test, we installed the proposed OpenFlow agent in the computer that manages the GPON network in the lab, and we installed ONOS in a virtual machine.

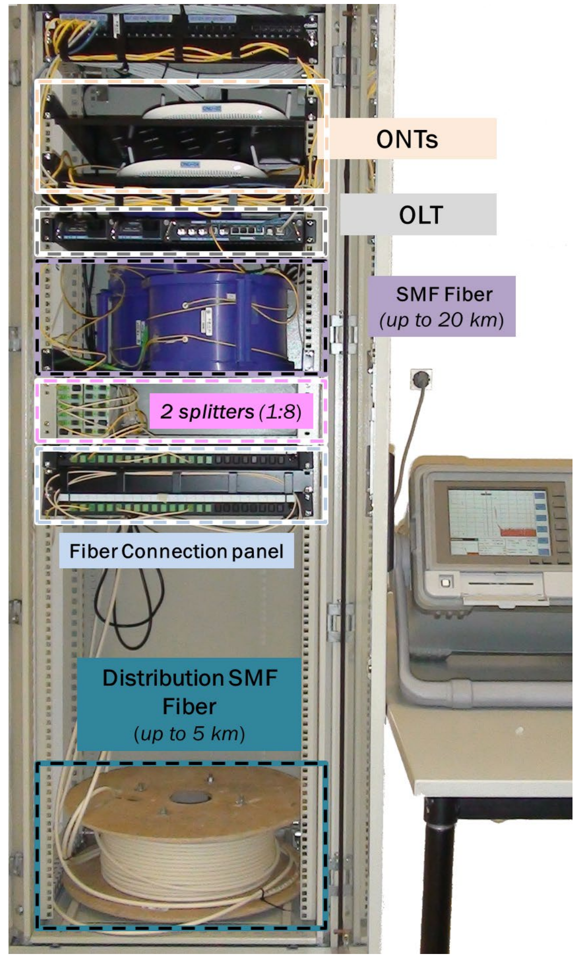

Fig. 3: GPON testbed in the Optical Communications laboratory

Once the network scenario is configured, the OpenFlow agent and ONOS are launched so that services (Internet, video) can be configured and sent to the GPON users (ONTs) via the OpenFlow protocol by means of flows and meters. These flows and meters are created using the ONOS interface, as it can be observed in Fig. 4 for the configuration of meters. In our case, the meter will be associated with a DROP parameter of 50000 Kbps (50 Mbps), i.e., services will be created with a maximum permitted bandwidth of 50 Mbps, with and meter identifier “id=1”. Besides, Fig. 5 shows the ONOS interface to configure a service my means of flows with the associated parameters (VLAN, meter with the associated bandwidth, ONT destination). This flow, associated with an Internet service, has destination MAC address 78:3d:5b:01:f7:30 (a GPON ONT), VLAN 806 and has the associated meter previously created “meterId=1” (50 Mbps), as it can be observed in Fig. 5.

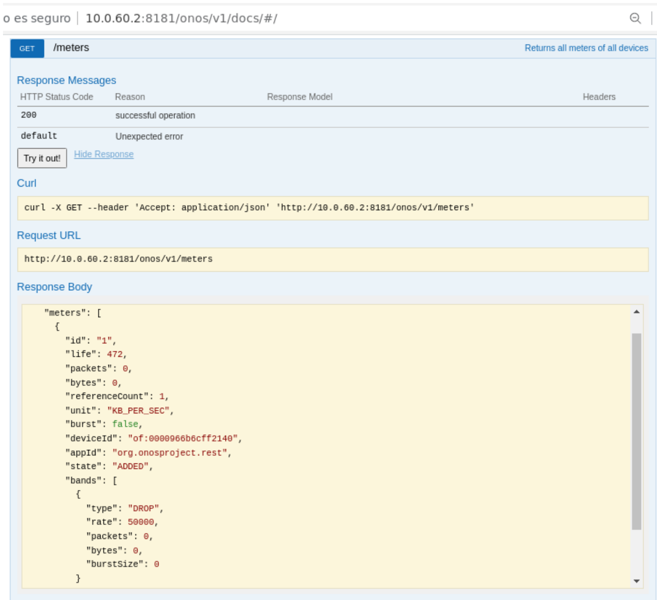

Fig. 4: ONOS interface to configure a meter with a maximum bandwidth

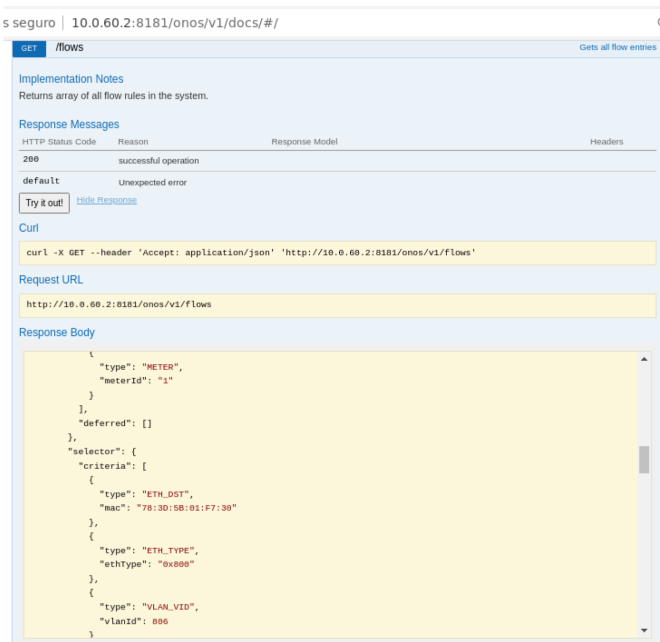

Fig. 5: ONOS interface to create a flow to configure a service

Besides, Fig. 6 shows the commands executed in the OLT to configure this service when the agent interacts with the OLT by means of the API. The responses returned by the OLT for each of these commands are also shown. As can be seen, all the configuration commands are executed correctly as they return "*Response result: Command success*". In fact, it can be observed that the configuration is done for the ONT (also known as ONU, Optical Network Unit) with local identifier 0, "*onu-local 0*", and that one OMCI port (of the GPON standard) "*omci-port 0*" is configured. Then, it can be also observed how some parameters related to the dynamic bandwidth allocation process in GPONs are configured (*alloc-id* and priority queues, T-CONT). Later, other configured parameters are the VLAN ("*vlan-tag1 806*"), the assigned maximum bandwidth of 50 Mbps for the downstream channel ("*policing downstream profile committed-max-bw 49984 committed-burst-size 1023 excess-max-bw 49984*") and the maximum upstream bandwidth of 50 Mbps ("*sla 800 service data status-report nsr gr-bw 45 gr-fine 0 be-bw 50 be-fine 0*").

olt-device 0
OLT CLI(DEV0)# olt-channel 1
OLT CLI(DEV0 CH1)# onu-local 0
OLT CLI(DEV0 CH1 LOC-ONU0)# omci-port 0
Response result: Command success
OLT CLI(DEV0 CH1 LOC-ONU0)# alloc-id 800
OLT CLI(DEV0 CH1 ONU(OMCI)0)# t-cont set slot-id 128 t-cont-id 0 alloc-id 800
Response result: Command success
OLT CLI(DEV0 CH1 ONU(OMCI)0)# mac-bridge-pcd create instance 2 bridge-id-ptr 1 port-num 2 tp-type gem tp-ptr 2 port-priority 0 port-path-cost 1 port-spanning-tree-ind true encap-method llc lanfcs-ind forward
Response result: Command success
OLT CLI(DEV0 CH1 ONU(OMCI)0)# gem-interworking-termination-point create instance 2 gem-port-nwk-ctp-conn-ptr 2 interwork-option mac-bridge-lan service-profile-ptr 1 interwork-tp-ptr 0 gal-profile-ptr 0
Response result: Command success
*OLT CLI(DEV0 CH1 ONU(OMCI)0)# vlan-tagging-filter-data create instance 2 forward-operation h-vid-a **vlan-tag1 806** vlan-priority1 7 vlan-tag2 null vlan-priority2 null vlan-tag3 null vlan-priority3 null vlan-tag4 null vlan-priority4 null*
Response result: Command success
OLT CLI(DEV0 CH1)# vlan uplink configuration port-id 800 min-cos 0 max-cos 7 de-bit disable primary-tag-handling false
*OLT CLI(DEV0 CH1)# **policing downstream profile committed-max-bw 49984 committed-burst-size 1023 excess-max-bw 49984 excess-burst-size 1023***
OLT CLI(OLT0 PON)# dba pythagoras 1
*OLT CLI(OLT0 CH1 PON-DBA(Pythagoras)**)# sla 800 service data status-report nsr gr-bw 45 gr-fine 0 be-bw 50 be-fine 0***
OLT CLI#

Fig. 6: Commands executed in the OLT to configure the services

Finally, Fig. 7 shows the measured throughput of the Internet service associated with the ONT (obtained by means of Wireshark). It can be seen that the ONT has been allocated the maximum bandwidth of 50 Mbps previously configured by the ONOS controller.

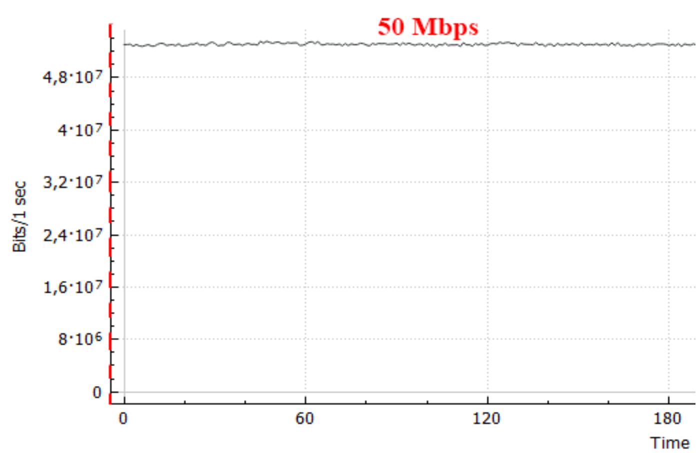

Fig. 7: Throughput of the Internet services measured with Wireshark

V. Conclusion

In this paper, we have described the design and development of a novel OpenFlow agent for managing and configuring PON networks and, in particular, GPON architectures, using the SDN ONOS controller. The OpenFlow agent emulates the behaviour of an OpenFlow switch, and is thus able to communicate with ONOS using OpenFlow messages. On the other hand, it can also communicate with OLTs in their native language using the specific APIs provided by the vendor, without the need to emulate SDN abstract layers in hardware devices. The proposed solution has been experimentally validated using OpenFlow and the ONOS controller over a legacy GPON.

Future work involves three different action lines, two of them related to research objectives and one related to teaching purposes. The first research line is focused on extending the proposed OpenFlow agent for its integration in XGSPONs (10G PONs). The second research line is focused on employing the SDN-enabled GPON testbed as the supporting wired infrastructure for a set of use cases related to connected vehicles, edge computing and computation offloading, which are currently under development. Finally, the third line is related to teaching purposes. The GPON testbed has already been used to train students on the physical layer of access networks [15], and with the introduction of these novel SDN capabilities, it will also be used to train students in SDN and OpenFlow technologies, helping them to acquire experimental skills, including the use and configuration of optical communication equipment.

ACKNOWLEDGMENT

This work has been funded by Consejería de Educación de la Junta de Castilla y León and by the European Regional Development Fund (VA231P20), and by the Spanish Ministry of Economy and Competitiveness and the Spanish Research Agency (grant PID2020-112675RB-C42 funded by MCIN/AEI/10.13039/501100011033).

REFERENCES

[1] "FTTH Council, "Europe Broadband status Market forecast by 2020-2025", https://www.ftthcouncil.eu/documents/Reports/2019/FTTH%20Council%20Europe%20-%20Forecast%20for%20EUROPE%202020-2025.pdf (Accessed: 20-Noember-2021).

[2] K. J. Kerpez, J. M. Cioffi, G. Ginis, M. Goldburg, S. Galli, and P. Silverman: Software-defined access networks, IEEE Communications Magazine, vol. 52, no. 9, pp. 152–159, Sept. 2014

[3] OpenFlow: Open Networking Foundation (ONF), in https://www.opennetworking.org/ (Accessed: 20-January-2022).

[4] S. S. W. Lee, K.-Y. Li, and M.-S. Wu: Design and Implementation of a GPON-Based Virtual OpenFlow-Enabled SDN Switch, IEEE/OSA Journal of Lightwave Technology, vol. 34, no. 10, pp. 2552–2561, May 2016.

[5] C. Qian et al. :Staged priority-based dynamic bandwidth allocation in software-defined hybrid passive optical network, Optical Engineerin, vol. 57, no. 12, Dec. 2018.

[6] M. Amiri, H. Al Osman and S. Shimohammdi: SDN-enabled Game-Aware Network Management for Residential Gateways, Proc. International Symposiym on Multimedia, Taichung, Taiwan, 2017, pp. 1-7.

[7] R. Challa, S.M. Raza, S. Jeon, H. Choo, and P. Thora: Refactoring Future Residential Networks into Slices, Proc. IEEE Conference on NFV-SDN, Verona, Italy, 2018, pp. 1-7.

[8] N. Merayo, D. de Pintos, J.C. Aguado, I. de Miguel, R.J. Durán, P. Fernández, R.M. Lorenzo, E.J. Abril, Experimental validation of an SDN residential network management proposal over a GPON testbed, Optical Switching and Networking, vol. 42. pp. 1-14, 2021, DOI: 10.1016/j.osn.2021.100631.

[9] N. Merayo, D. de Pintos, J.C. Aguado, I. de Miguel, R.J. Durán, P. Fernández, R.M. Lorenzo, E.J. Abril, An Experimental OpenFlow Proposal over Legacy GPONs to allow Real-Time service reconfiguration policies, Applied Sciences-Basel, vol. 11(3). pp. 1-17, 2021.

[10] VOLTHA: The Open Source Poject VOLTHA, in https://opennetworking.org/voltha (Accessed: 22-January-2022).

[11] ONOS: The ONOS SDN controller, in https://opennetworking.org/onos/ (Accessed: 22-January-2022).

[12] Open vSwitch: Production Quality, Multilayer Open Virtual Switch, in https://www.openvswitch.org/, [Accessed: 10-January-2022].

[13] Python-OpenFlow library, Kytos SDN. in https://github.com/kytos/python-openflow, (Accessed: 10-January-2022).

[14] Telnet R-I: Telnet vendor web page, in https://www.telnet-ri.es/ (Accessed: February-2022).

[15] N. Merayo, J.C. Aguado, I. de Miguel, R.J. Durán, P. Fernández, R.M. Lorenzo, E.J. Abril, A testbed and a simulation laboratory for training engineering students in optical access network technologies, Computer Applictions in Engineering Education, vol. 8. pp. 1-19, 2021, DOI: 10.1002/cae.22429.

.