This is a postprint version of the following published document: Merayo Álvarez, Noemí; Pintos, David de; Azofra Ovejero, Javier Néstor; Aguado Manzano, Juan Carlos; Miguel Jiménez, Ignacio de; Durán Barroso, Ramón José; Ruiz Pérez, Lidia; Fernández Reguero, Patricia; Lorenzo Toledo, Rubén Mateo; Abril Domingo, Evaristo José. Implementation of an automated control module in Python for a SDN GPON testbed. In: 11ª Reunión Española de Optoelectrónica, OPTOEL'19. Zaragoza: Universidad de Zaragoza, 2019, 6 p.



# Implementation of an automated control module in Python for a SDN GPON testbed

Noemí Merayo, David Pintos, Javier Azofra, Juan Carlos Aguado, Ignacio de Miguel, Ramón José Durán, Lidia Ruiz, Patricia Fernández, Rubén Mateo Lorenzo, Evaristo José Abril

Optical Communications Group, 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, noemer@tel.uva.es

Contact name: noemer@tel.uva.es

## ABSTRACT:

Passive Optical Networks (PONs) are the most deployed network infrastructure in the access segment. Indeed, the number of Fibre-to-the-Home (FTTH) subscribers in Europe increased with more than 59.6 million in September 2018. On the other hand, the automation of the PON configuration to lead to better control of the network management may provide many advantages to exploit the PON capabilities, for example to facilitate the integration of Software Defined Networking (SDN) strategies in this networks. Indeed, SDN may improve the network efficiency of networks regarding scalability, management and Quality of Service (QoS) performance. Therefore, we propose a network automation system in Python to control and configure a SDN-GPON solution implemented over a real GPON testbed.

**Key words:** Gigabit Passive Optical Networks (GPON); testbed; network automatization; Python control; Software Defined Networking (SDN)

## 1.- Introduction

The passive network infrastructure (Passive Optical Access Networks, PON) has emerged as the ideal solution to face the challenges of the access segment. Based on a point to multipoint architecture, it is quite cheap since it uses a single fiber from the Central Office (CO) to the distribution point (network subscribers). In addition, it does not require electrical power in the entire external plant of the network and uses optical splitters that divides the signal as many branches as subscribers connected to the network (tree topology) [1].

In the most basic configuration, a PON consists of an OLT (Optical Line Terminal) terminal and several stations (Optical Network Unit). The OLT works as an access node, connecting the optical access network with the backbone network. The optical fiber is responsible for transporting the packets within the network, using two different wavelengths, one to transport data from the OLT to the ONT (downstream channel at 1490 nm) and another to transport packets from the ONTs to the OLT (upstream channel at 1310 nm). In the downstream, a PON is a point-multipoint network, while in the upstream the network behaves like point-to-point. As a consequence, a MAC (Medium Access Control) protocol is required to manage data collisions of different users (ONTs) [2-3].

On the other hand, Software Defined Networking (SDN) and its integration in PONs can allow a more efficient operation and management of networks thanks to the abstraction of the control and data planes [4]. Therefore, the integration of SDN in PONs shows an important short term projection and many recent works focus on it [5-6]. Furthermore, since PON vendors provide different software tools to configure the network, it

can be very beneficial to automate the network control and make easier its management and configuration. As a consequence, the integration of SDN techniques together with the automatization of the PON configuration may provide a powerful control of the PON capabilities. Therefore, in this paper we describe the implementation of a SDN solution for a real GPON testbed that can be controlled by a global program in Python that automates the network configuration.

## 2.- Description of the real GPON testbed

We have deployed a real testbed composed of a Gigabit Passive Optical Access Network (GPON) in one of our laboratories, as it can be observed in Fig. 1. The GPON equipment belongs to the Telnet-RI vendor [7]. Then, we deploy an OLT SmartOLT 350 (located at the central office), which implements a full-duplex GPON interface of 2.488/1.244 Gbps (downstream and upstream respectively) and compliant with ITU-T G.984.1-4 [8]. It supports up to 4 GPON ports, each supporting up to 64 ONTs.

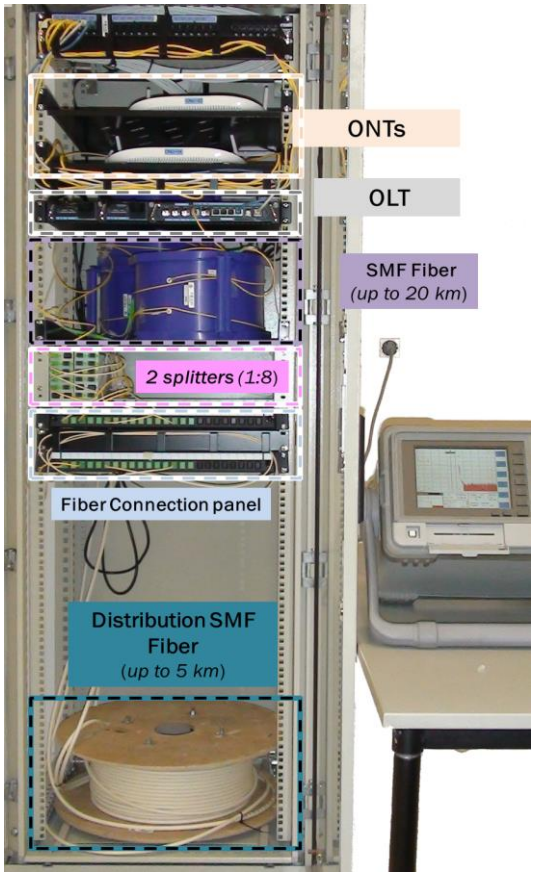


*Fig. 1: Real deployment of the GPON testbed in the Optical Communications laboratory.*

The connection between the OLT and the optical splitter includes three spools of Standard Single Mode Fiber (SSMF) of different lengths (two of 5 km and one of 10 km), that allows to configure the reach of the GPON network. The testbed is also equipped with two optical splitters 1:8, thus enabling the configuration of a two-stage splitting topology.

Then, the splitters are connected to the ONTs by means of distribution fibers. The length of each link can be individually configured, by means of a junction panel, from 100 meters up to 5 km of SMF fiber. In this way, the testbed can emulate realistic scenarios where different users are located at different distances of the central office (OLT). The junction box has 16 input ports and 16 output ports. The signal that comes from the splitter connects to an input port and it exits through an output port. The distance between the splitter and each ONT depends on the com-

This is a postprint version of the following published document: Merayo Álvarez, Noemí; Pintos, David de; Azofra Ovejero, Javier Néstor; Aguado Manzano, Juan Carlos; Miguel Jiménez, Ignacio de; Durán Barroso, Ramón José; Ruiz Pérez, Lidia; Fernández Reguero, Patricia; Lorenzo Toledo, Rubén Mateo; Abril Domingo, Evaristo José. Implementation of an automated control module in Python for a SDN GPON testbed. In: 11ª Reunión Española de Optoelectrónica, OPTOEL'19. Zaragoza: Universidad de Zaragoza, 2019, 6 p.



bination of input port and output port (the maximum is set to 5 km). In this way, Fig. 2 shows the general scheme of the network, where the distances of the junction box are detailed.

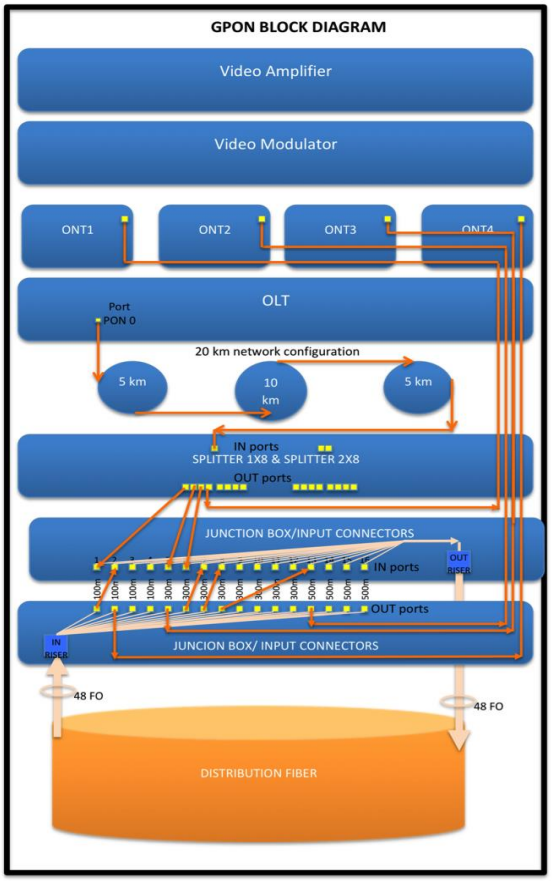


*Fig. 2: Block Diagram of the GPON testbed.*

For example, for $ONT_3$ as the output of the splitter is connected to the input port "5" and the output port "5" is connected to this ONT, the distance between the splitter and the ONT is 300 meters. Moreover, if we concatenate distances, for example for $ONT_4$, we connect one output of the splitter to the input port 1, the output port 1 to the input port 2 and the output port 2 to the $ONT_4$, the total distance is set to 200 meters. Then, the concatenation of the 15 input and output ports would be the maximum distance between the splitter and ONU (5 km of the distribution fiber). Finally, the testbed employs L3 (level 3) model ONTs, which means that they integrate router functionalities and L2 (level 2) ONTs [7]. Every ONT complies with the ITU-T G.984.x specifications.

## 3.- Implementation of SDN in the GPON

Software Defined Networking (SDN) in PONs can improve the Quality of Service (QoS) and the bandwidth and resource allocation as well as permitting an efficient network monitoring. As a consequence, in previous works [9] we have proposed an SDN solution with OpenFlow [10] using a SDN controller (OpenDaylight [11]) that controls the GPON configuration (service profiles) implementing OpenFlow Virtual Switches (OVS) [12] at the OLT and ONTs which emulate a SDN layer (Fig. 3). Then, we deployed a Central OVS (COVS) just after the OLT (inside a computer) to deal with the downstream traffic and services of every subscriber. Besides, we deployed a Remote

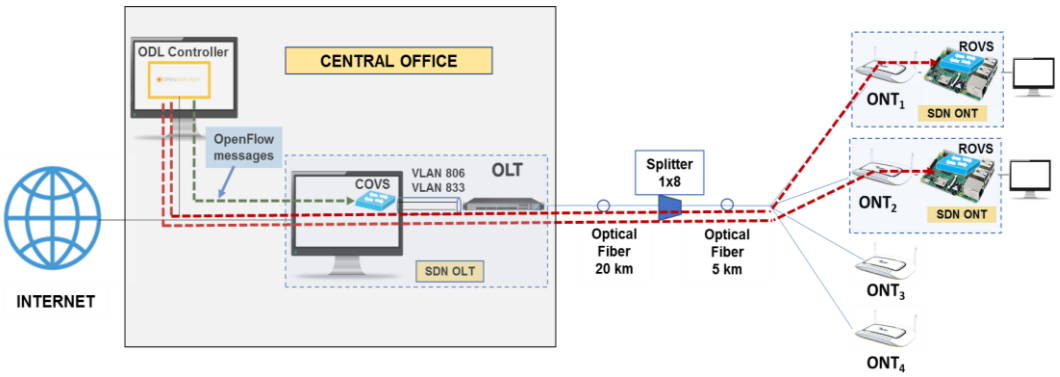


*Fig. 3: The OpenFlow SDN scenario implemented in the GPON testbed.*

This is a postprint version of the following published document: Merayo Álvarez, Noemí; Pintos, David de; Azofra Ovejero, Javier Néstor; Aguado Manzano, Juan Carlos; Miguel Jiménez, Ignacio de; Durán Barroso, Ramón José; Ruiz Pérez, Lidia; Fernández Reguero, Patricia; Lorenzo Toledo, Rubén Mateo; Abril Domingo, Evaristo José. Implementation of an automated control module in Python for a SDN GPON testbed. In: 11ª Reunión Española de Optoelectrónica, OPTOEL'19. Zaragoza: Universidad de Zaragoza, 2019, 6 p.



OVS (ROVS) at each ONT (installed in a Rasberry Pi to control the upstream traffic of each subscriber. It can be noticed that in a real and future implementation the SDN layer will be implemented inside the OLTs and ONTs. In this network scenario, the SDN controller can dynamically send flow messages to every OVS (COVS, ROVS's) to configure or modify services associated with each subscriber. Consequently, SDN permits a fast and centralized control of the GPON according to the real time demands of subscribers. As OpenFlow is based on flows sent by the ODL controller to every OVS, we define two flows for each service, one to configure the service at the upstream channel and another for the downstream. Furthermore, we configure meters associated with each flow in order to control the maximum guaranteed badwidth of the service at both channels (upstream/downstream).

## 4.- Automation of the GPON network management through Python

To configure the GPON testbed the vendor provides two different methods: TGMS (Telnet GPON Management System) and the Command Line Terminal (CLI) of the OLT by means of GPON native commands. The first one, TGMS, consists of a virtual machine configured by the vendor that gives access to an easy web interface through which the network services of the GPON are configured. In contrast, the CLI connects to the management port of the OLT to configure services and user profiles by means of GPON commands, which allows a greater level control than TGMS when configuring the global GPON. However, the used of lower level commands causes a complex control of the network so the development of a system to automate the management and configuration of the GPON (services, devices, profiles) in a transparent way will provide great advantages.

As a consequence, we have implemented a global program in Python combining the CLI (GPON commands) to allow a more efficient control of the optical access network. Then, Fig. 4 shows the main menu of the global program in Python.

```
Welcome to the service management system.
1. Service configuration
2. Attach service to ONT
3. Detach service from ONT
4. OpenFlow configuration
5. ONT Router configuration
6. Exit
What do you want to do?:
```

*Fig. 4: Main menu of the global program in Python.*

As it can be observed it provides several functionalities. To operate with the services, it must be enter option 1 "Service configuration". Within this menu (Fig. 5), the user can select whether to display all existing services (List of configured services), create a new service, modify a service, delete a service or return to the previous menu.

```
You have chosen: Service configuration

1. List of configured services
2. Create new service
3. Modify service
4. Delete service
5. Go back
Please, select what do you want to do:
```

*Fig. 5: Menu to configure services inside the GPON.*

If option 2 is selected (Create new service) the application opens the menu of Fig. 6. In it, all the parameters needed to configure a new service (type of service, VLAN, guaranteed bandwidth) have to be inserted. All services are stored in a local database.

```
You have chosen: Create new service

Type of service (Internet: 0, Video: 1): 0
VLAN [0-4095]: 806
VLAN Priority [0-7]: 0
Guaranteed downstream (Kbps) [0-2488000]: 30000
Excess downstream (Kbps) [0-2488000]: 0
Guaranteed upstream (Mbps) [0-1244]: 20
Excess upstream (Mbps) [0-1244]:
```

*Fig. 6: Option to create a new service.*

Moreover, the functionality "Modify service" permits users to modify one or several the parameters regarding the VLAN, guaranteed bandwidth or excess bandwidth (Fig. 7). Finally, using option 4 of the menu ("Delete service") allows to select one service of the database and delete it.

This is a postprint version of the following published document: Merayo Álvarez, Noemí; Pintos, David de; Azofra Ovejero, Javier Néstor; Aguado Manzano, Juan Carlos; Miguel Jiménez, Ignacio de; Durán Barroso, Ramón José; Ruiz Pérez, Lidia; Fernández Reguero, Patricia; Lorenzo Toledo, Rubén Mateo; Abril Domingo, Evaristo José. Implementation of an automated control module in Python for a SDN GPON testbed. In: 11ª Reunión Española de Optoelectrónica, OPTOEL'19. Zaragoza: Universidad de Zaragoza, 2019, 6 p.



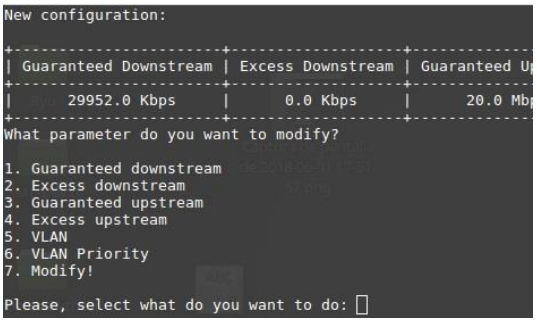


*Fig. 1: Menu to modify parameters of the exiting services.*

On the other hand, the global program (Fig. 8) permits to associate services to subscribers that is, to specific ONTs ("Attach service to ONT"). As it can be observed in this option, first appears the list of services configured in the database. From that list, the service identifier (ID) to associate with an ONT has to be chosen ("Select ID"). Subsequently, the ONTs connected to the OLT are displayed together with the local identifier given by the OLT (ONT ID). After choosing one specific ONT by means of its ID, the service chosen in the previous step is loaded in the ONT and this association is saved in the local database.

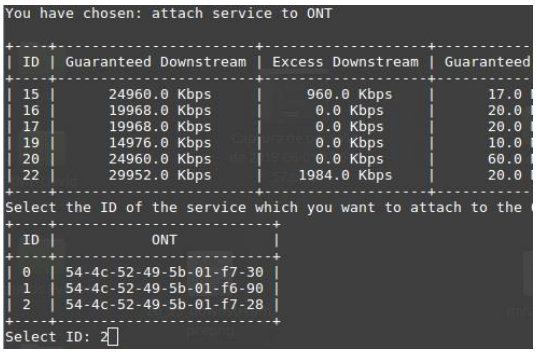


*Fig. 8: Association of a service to an ONT.*

Furthermore, the global program allows users to detach a service from an ONT ("Detach service from ONT"). In this case, the connected ONTs are first displayed with their corresponding identifier (ONTs ID) as it is shown in Fig. 9 in order to select the specific ONT. Then, it appears the configured services of that ONT to select the ID of the service to detach.

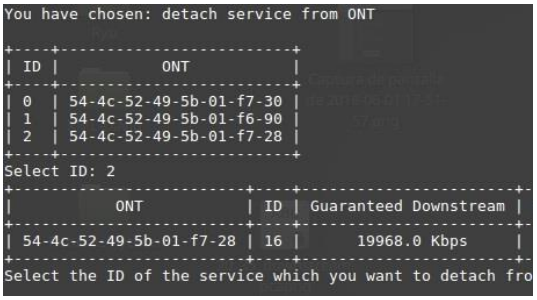


*Fig. 9: To detach a service from an ONT.*

On the other hand, to allow a more powerful control of the GPON, we have integrated a SDN management approach to deal with the network configuration using OpenFlow ("OpenFlow Configuration" in the main menu). This functionality allows administrators to configure the GPON by creating OpenFlow flows with the corresponding parameters of the services. When the OpenFlow option is selected it appears the new menu of Fig. 10, which permits to attach or detach services to one specific ONT using OpenFlow messages.

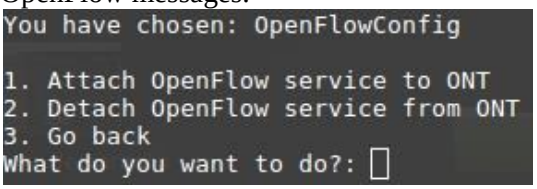


*Fig. 10: GPON configuration using OpenFlow.*

In this case, the program send flows and meters (to guarantee bandwidth levels) to the OVS of the corresponding ONT through HTTP PUT or DELETE requests to the OpenDayLight controller depending on whether we want to create or delete a service respectively. In Fig. 11 it can be observed a flow configuration in Python sent to one OVS.

Finally, it has been implemented a functionality to allow a remote management of the ONTs and its router using the CLI commands and the XML configuration file of the ONT, called "ONT router configuration" (Fig. 12). In fact, the functionalities regarding services ("Create a new service", "Show services", "Delete services") permits to configure the router of the ONT with the existing services in the OLT and to synchronize the OLT and the ONT with such service. Be-

This is a postprint version of the following published document: Merayo Álvarez, Noemí; Pintos, David de; Azofra Ovejero, Javier Néstor; Aguado Manzano, Juan Carlos; Miguel Jiménez, Ignacio de; Durán Barroso, Ramón José; Ruiz Pérez, Lidia; Fernández Reguero, Patricia; Lorenzo Toledo, Rubén Mateo; Abril Domingo, Evaristo José. Implementation of an automated control module in Python for a SDN GPON testbed. In: 11ª Reunión Española de Optoelectrónica, OPTOEL'19. Zaragoza: Universidad de Zaragoza, 2019, 6 p.



sides, using the "Policy routing" functionality it can be permitted to configure or delete static/dynamic routes depending on the type of delivered service (Multimedia, Internet, Voice over IP).


*Fig. 11: Configuration of flows for services in ONTs.*

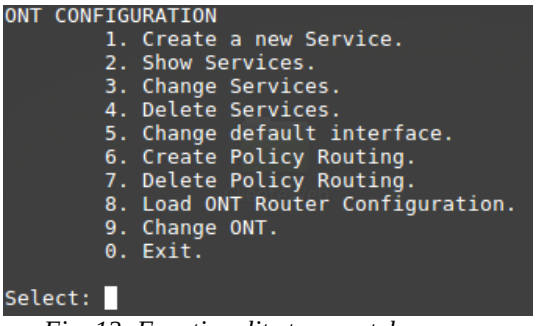


*Fig. 12: Functionality to remotely manage the ONT configuration.*

## 5.- Conclusion

In this paper, we described the deployment and automated configuration of a real GPON testbed of 25 km. Indeed, we have developed a global program in Python that permits to efficiently automate the management and control the GPON network. Then, it permits to create, modify or delete services which are stored in a local database. Moreover, the program can remotely configure the ONTs and its router in a very efficient and fast way. On the other hand, as SDN is acquiring a great importance in future networks and many proposals are focusing in its integration with PONs, we have proposed and described a novel SDN-GPON implementation over our real GPON testbed. Furthermore, we have developed a new functionality in the Python program that allows administrators to configure the SDN-GPON solution by means of the OpenFlow standard.

*Acknowledgements*: This work has been sponsored by the Spanish Ministry of Science & Innovation TEC2017-84423-C3-1-P

## *References*

[1] B. Lung, "Fiber to the Home Using a PON Infrastructure", IEEE/OSA Journal of Lightwave Technology, vol. 2, no. 1, pp. 4568-4583, 2006.

[2] Glen Kramer, Banerjee Mukherjee, Gerry Pesavento, "*IPACT a dynamic protocol for an Ethernet PON (EPON)*", IEEE Communications Magazine, vol. 40, no. 2, pp. 74-80, 2002.

[3] N. Merayo, P. Pavon-Marino, J. C. Aguado, R. J. Durán, F. Burrull, V. Bueno-Delgado, "*Fair bandwidth allocation algorithm for PONS based on network utility maximization*", IEEE/OSA Journal of Optical Communications and Networking, vol. 9, no. 1, pp. 75 – 86, 2017.

[4] A. S. Thyagaturu et al, "*Software Defined Optical Networks (SDONs): A Comprehensive Survey*", IEEE Communications Surveys & Tutorials, vol. 18, no. 1, 2016.

[5] S. W. Lee et al., "*Design and implementation of a GPON-based virtual OpenFlow-enabled SDN switct*", Journal of Lightwave Technology, vol. 34, no. 10, pp. 2552, 2016.

[6] P. Parol et al., "Towards networks of the future: SDN paradigm introduction to PON networking for business applications,". Proc, of the FedCSIS, Poland, 2013.

[7] Telnet Redes Inteligentes, https://www.telnet-ri.es/.

[8] International Telecommunication Union, "G.984.1 : Gigabit-capable passive optical networks (GPON): General characteristics," http://www.itu.int/rec/T-REC-G.984.1-200803-I, Accessed January 2017.

[9] N. Merayo et al., "*Dynamic bandwidth control and testbed validation of a SDN based GPON*", Proc, of the ECOC Conference, Rome (Italy), 2018.

[10] OpenFlow,https://www.opennetworking.org

[11] OpenDayLight, ttps://opendaylight.org.

[12] Open VSwitch, https://openvswitch.org