Article

An Experimental OpenFlow Proposal over Legacy GPONs to Allow Real-Time Service Reconfiguration Policies

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

Optical Communications Group of the Department of Signal Theory, Communications and Telematic Engineering. E.T.S.I. Telecomunicación, Campus Miguel Delibes, Universidad de Valladolid (Spain), Paseo de Belén 15, 47011 Valladolid, Spain; david9.krd@gmail.com (D.d.P.); jaguado@tel.uva.es (J.C.A.); ignacio.miguel@tel.uva.es (I.d.M.); rduran@tel.uva.es (R.J.D.); patfer@tel.uva.es (P.F.); rublor@tel.uva.es (R.M.L.); ejad@tel.uva.es (E.J.A.)
* Correspondence: noemer@tel.uva.es; Tel.: +34-983-423-000 (ext. 5549)

Abstract: The integration of Software Defined Networking (SDN) technologies in Passive Optical Networks (PONs) would provide great advantages to Internet Service Providers (ISPs) and Network Operators, since they can optimize the network operation and reduce its complexity. However, some tasks regarding online service and network configuration strategies are difficult to move to external SDN-controllers since they are time-critical operations. However, the control of some of these policies by SDN techniques could lead to better network and management configuration in a centralized and automatic way. As a consequence, we propose and experimentally test the integration of an OpenFlow approach over legacy Gigabit Passive Optical Networks (GPONs), which allows moving some global service configuration policies to an external SDN controller implementing an SDN management layer that adjust these strategies according to dynamic Quality of Service (QoS) requirements of services in residential users. The viability and efficiency of our approach are demonstrated using a GPON testbed and proposing a new business scenario for ISPs and Network Operators.

Keywords: software defined networking (SDN); gigabit passive optical networks (GPONs); online service configuration policies; experimental validation

1. Introduction

Software Defined Networking (SDN) is an emerging and open communications technology that allows an abstraction and separation of the control and data planes. This separation permits an efficient control of the network and higher scalability and management automation [1,2]. Then, SDN allows the network to be intelligently and centrally controlled using software applications, so operators can manage the network consistently and comprehensively, regardless of the underlying network technology. Then, with SDN a part of the datapath resides on the switches and routers of the network, but there is a central controller that makes high-level routing decisions over them using protocols such as OpenFlow [3], NETCONF [4], or RESTCONF [5].

Due to these advantages, the integration of SDN in Passive Optical Networks (PONs) is gaining high importance, and its implementation to provide different functionalities is being proposed extensively by many authors. In addition, it is worth noting the importance of PON technologies in the deployment of current access networks worldwide. Indeed, the FTTH Council in the latest news of December 2020 states that 202 million homes passed with FTTH/B by 2026 in Europe compared to 88.1 million in 2019 [6]. Moreover, the FFTH/B penetration is expected 73.3% in 2026 (43.3% in 2019). In this network context, PON technologies will become predominant in the coming years, going from 48.4% in September 2018 to 73% in 2025 [7].

Due to these factors, much attention is being paid to investigate the integration of SDN technologies in PON infrastructures. In this way, many research papers try to emulate

an SDN behavior on PON legacy equipment, so that PON devices can become SDN-controllable devices using SDN protocols [8–14]. In contrast, other proposals focus on moving certain features of current PON devices to external SDN controllers to achieve a centralized SDN control of the network, for example, bandwidth allocation policies or PON devices registration [15–23]. In this way, PONs are Point to MultiPoint (P2MP) networks based on a tree topology between the Optical Line Terminal (OLT) and the Optical Network Terminals (ONTs). In the upstream channel, from the ONTs to the OLT, PONs show a multipoint to point topology and every ONT shares the same transmission channel. Therefore, a MAC (Medium Access Control) protocol is required inside this channel in order to avoid collisions among data from different users (ONTs). Dynamic Bandwidth Allocation (DBA) based on Time Division Multiple Access (TDMA), is the most used strategy to share the channel capacity among ONTs (users) cycle by cycle, since the OLT dynamically distributes the available bandwidth depending on the current demand of users (ONTs) and its contracted services. The cycle time is the total time in which all ONTs connected to the PON transmit in a round robin discipline (TDMA).

As it can be noticed, the allocation process inside PON infrastructures is made between the OLT and the ONTs cycle by cycle, so its global outsourcing to an external SDN controller can cause an inefficient network performance in the bandwidth allocation process due to the latency between the SDN controller and the OLT, since SDN controllers are outside PONs at a certain distance. However, it could be worth that some decisions regarding DBA policies, such as the maximum bandwidth assigned to specific services (such as voice, video or data), can be managed by the SDN controller to provide an efficient and global SDN network management and configuration of the PON, although these policies do not imply the cycle-by-cycle control of the bandwidth allocation process, which would cause low efficiency due to the distance between the SDN controller and the OLT.

As a consequence, in this paper, we propose designing and implementing an SDN management layer over a GPON testbed able to deal with some service configuration strategies, such as the maximum allowed bandwidth to services, according to the real-time QoS requirements of network subscribers (ONTs). These global policies do not require a cycle-by-cycle communication between the SDN management layer and the OLT, thereby not degrading performance despite the distance between the SDN infrastructure and the PON. In contrast to other existing approaches, that most of them propose service provisioning or DBA policies by means of simulation models, we demonstrate the feasibility of moving specific strategies over a legacy GPON testbed which has implemented an end-to-end SDN solution using the OpenFlow protocol and SDN virtual switches (Open Virtual Switch, OVS) [24] connected to OLTs and ONTs, all of them controlled by an external SDN controller, in our case OpenDayLight (ODL) [25]. Therefore, we propose both a SDN solution over a GPON testbed and the implementation of an SDN management layer that deals with service configuration policies inside the GPON testbed. In fact, we propose that the SDN management layer can be able to provide and configure real-time Internet services and their associated maximum bandwidth to users that contract Internet plans according to their real-time demand and contracted bandwidth requirements using the SDN controller and the OpenFlow protocol. Furthermore, we check the viability of our proposal by designing a new business model for network operators and providers.

The paper is organized as follows. Section 2 describes the state of the art regarding the integration of SDN in PONs. Section 3 explains the experimental implementation of our SDN approach over GPONs and the description of the online service configuration policies over our network scenario. Section 4 presents the experimental results and discussion of our SDN-GPON proposal using a particular business model. Finally, in Section 5, the most relevant conclusions obtained in this experimental study are shown.

2. State of the Art

Many researches are focusing on the integration of SDN in PONs. On the one hand, some studies integrate OpenFlow switching paradigms in the PON control layer. In this

way, authors in [8] propose a SDN-OLT solution in GPONs using virtual switches and the OpenFlow standard in order to emulate a SDN layer in the OLT. In addition, authors in [9] propose to implement virtual switches inside ONUs able to select OFDM channels for TWDM-PONs. Other proposals regarding virtual switches for PONs were proposed in [10] to improve the efficiency and flexibility for data center networks, especially for intra-datacenter communications. The papers [11] describe an SDN-controller to simultaneously manage several SDN-OLTs in PON infrastructures. Other researches are working on mapping OpenFlow messages to native PON configuration commands [12] and defining extensions of the OpenFlow standard for PONs technologies [13]. In addition, the Open Source Virtual OLT Hardware Abstraction (VOLTHA) project [14] proposes to abstract PONs as programmable Ethernet switches that can be controlled by an SDN controller and therefore, to manage its network configuration and devices (OLT, ONTs).

On the other hand, issues related to the Dynamic Bandwidth Allocation (DBA) or the registration and management of ONTs are difficult to cover with SDN in PONs, due to the latency between the SDN controller and the OLTs. In this line, authors in [15] propose the SIEPON architecture where the ONT registration process and some global DBA policies are moved to an external SDN controller, since they are not time-critical operations. Authors in this paper describe the simulation model and they analyze some network parameters such as the delay and the throughput performance by a simulation study. In addition, authors in [16] describe a novel SDN architecture for EPONs integrating a reprogrammable DBA module inside the SDN controller in contrast to the hardware DBA modules (inside OLTs) existing in traditional PONs. In this proposal, an SDN controller deals with several DBA algorithms and it deploys them in the OLT according to the network global status and the traffic demands of ONTs. This proposal and the results are supported with simulations. Authors in [17] experimentally demonstrate a novel SDN architecture for remote unified control with SA-FS to dynamically provision and adjust services according to the real-time Quality of Service (QoS) demand. Regarding TWDM-PONs, authors in [18] designed a novel software-defined solution to fulfil QoS requirements using a centralized SDN controller that dynamically allocates bandwidth and wavelengths to users. This model and the algorithm's performance are supported with simulations. In the same way, paper [19] describes a new SDN architecture and bandwidth allocation policy to guarantee QoS requirements in TWDM-PONs. Therefore, it can be noticed that most of the research related to DBA strategies or online service reconfiguration policies controlled by SDN are implemented using simulation models, but they have not been tested in real PON network environments with legacy equipment.

Although we do not focus on protection and energy savings in PONs, it is worthy to mention that other research works on the use of SDN in PONs have considered those aims. For instance, authors in [20] developed an experimental SDN architecture to allow protection and dynamic service provisioning in a multi-wavelength PON in coordination with a core SDN-based network, and authors in [21] propose moving the power control of OLTs and ONTs to the SDN controller to achieve energy savings. Finally, SDN solutions are also applied in Virtual PONs (VPON), such as authors in [22] that carry out a simulation analysis of a novel software-defined Hybrid Passive Optical Network (HPON) architecture proposed over multiple VPONs managed by a priority-based DBA mechanism called SPB-DBA. In addition, the Flex PON architecture proposed in [23] defines a full-service Software-Defined Solution to provide a flexible and a programmable service provisioning in VPONs. In the article, authors experimentally demonstrate flex link using Digital Signal Processing (DSP) technologies.

Therefore, it can be stated that the integration of SDN technologies in PONs could provide great advantages to network operators, since they can optimize the network operation and reduce the network complexity. However, some tasks regarding online service and network configuration or dynamic bandwidth allocation are difficult to move to external SDN controllers since they are time-critical operations, although some of these policies can be very interesting to be managed by SDN controllers to provide a more efficient

control in PON architectures. As a consequence, we propose and experimentally test an SDN solution over legacy GPONs, which allows moving some global service configuration policies to an SDN controller by means of programming an external SDN management layer that dynamically adjusts these policies according to the stipulated dynamic QoS requirements of residential users. This proposal can enable new business models for network operators and providers, so the viability of our solution is demonstrated proposing a new possible business scenario for users and operators. However, other business models can be applied with our strategy to exploit the possibilities it offers. Finally, our proposal also demonstrates the feasibility of moving some QoS services and network configurations out of the PON network to an external SDN controller. To the best of our knowledge, this is the first time that a real-time reconfiguration service policy is experimentally tested in a legacy GPON using SDN techniques apart from the approach described in [17]. However, in the architecture presented in [17] they build a new OpenFlow-enabled OLT node with an embedded OpenFlow agent. To control this network architecture, authors propose extending the existing OpenFlow protocol to interact with the main characteristics of PONs (such as guard time, bandwidth, time slot, ONU identification) and the architecture permits dealing with the DBA algorithm inside the OLT. In contrast, in our proposal, we use the real OpenFlow standard that does not allow the control of PON parameters in its flow rules. Moreover, legacy GPON equipment does not permit modifying the DBA algorithm, since it is integrated in a chipset inside the OLT. Therefore, we propose a different solution using existing technologies to provide a realistic network scenario with legacy and commercial GPON equipment.

3. Experimental OpenFlow Proposal over Legacy GPONs to Allow Real-Time Service Reconfiguration Policies

3.1. Implementation of a SDN Approach over Legacy GPON Equipment

We propose integrating a SDN approach into commercial GPONs using the OpenFlow standard together with the ODL controller and a set of OVSs connected to GPON devices, OLTs and ONTs, to emulate a SDN layer on these devices. Among the different SDN controllers, we have deployed OpenDayLight (ODL), although other OpenFlow compatible controllers could be deployed. For instance, ODL and ONOS are quite featured SDN controllers, since they show high modularity, good productivity, effective for large scale networks and a good Graphical User Interface (GUI) [26,27]. Moreover, both are open source and benefit from a large developer and user communities under the Linux Foundation Networking. Moreover, ODL can support so many applications, such as new IoT southbound interfaces, that could make it one of the most important future SDN controllers [26]. As a consequence, we selected ODL in our proposal, although other high-performance SDN controllers, such as ONOS, could be deployed in the same way. In this network scenario, the ODL controller will be able to manage the GPON configuration and its services/profiles according to the stipulated QoS requirements by sending OpenFlow messages to every OVS, as it can be observed in Figure 1.

Therefore, we deploy an OVS (Central OVS, COVS) on a computer connected to the OLT to implement a SDN layer on the OLT to manage the GPON downstream channel (traffic coming from outside the GPON). In the same computer, a router and a DHCP server were installed to control the IP addresses of the connected ONTs. Moreover, we deployed OVSs (Residential OVS, ROVS) on mini computers connected to each ONT to implement and SDN layer on the ONTs to manage the GPON upstream channel (traffic coming from network subscribers). It can be noticed that in the future, the Openflow switches (COVS and ROVS) should be integrated inside OLTs and ONTs (SDN based OLT/ONT), but as commercial GPON equipment do not support SDN it is necessary to emulate a SDN layer.
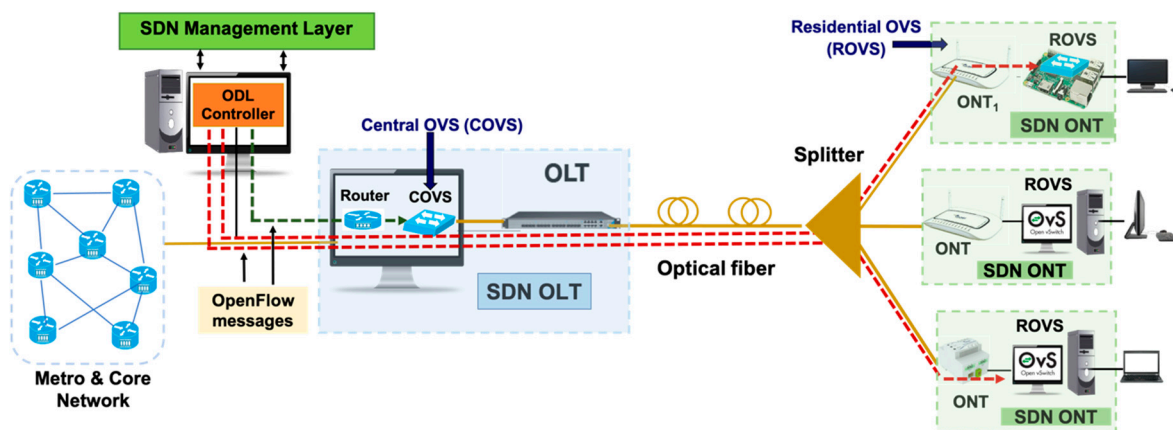
Figure 1. The software defined networking (SDN)-OpenFlow scenario implemented in the gigabit passive optical networks (GPONs) testbed.

In order to dynamically control our SDN-GPON approach, we have implemented an SDN management layer controlled by ISPs or Network Operators that communicates with SDN controllers (a single ODL controller in our case) to modify the GPON configuration according to the real-time QoS requirements and the traffic conditions. Therefore, the SDN controller/s and the management layer belong to ISPs/Network Operators that provide Internet connections and services to their residential subscribers, such as data (Internet), VoIP (Voice over IP), and HDTV (High-Definition TV). Therefore, the ODL controller sends OpenFlow messages to the different OVSs (OLT, ONTs) following the instructions of the SDN management layer to configure the QoS requirements of services (e.g., guaranteed bandwidth). These instructions are kept in entries (called flows) of the OpenFlow tables that are created and modified by the SDN controller, so that each entry corresponds to specific service QoS requirements. The flow tables are real-time modified and sent by the ODL by means of flows (one flow corresponds to one entry) encapsulated in OpenFlow messages to the different OVSs when QoS requirements or services should be updated in OLTs and ONTs. Each flow has several matches (match conditions), so each provides a condition that must be met in the packets to be part of the flow, that is, a filter condition for packets at each OVS. In our SDN approach, packets are filtered with the MAC address of the ONT to distinguish the traffic of each residential user. Furthermore, as one service in GPONs must be configured for both channels (upstream, downstream) two flows must be created in the ODL, one sent to the corresponding user through its COV (with the downstream QoS requirements) and another to the corresponding ROVS (with the upstream QoS requirements). In addition, since a maximum bandwidth is assigned to each service as an important QoS requirement (Internet, Video, VoIP) to control, for example, the guaranteed bandwidth, this parameter is reflected in the OpenFlow standard using meters [28] which are directly associated with the flows of the said service. Therefore, a meter measures the rate of packets and controls the maximum rate of its associated flow, so it controls the maximum bandwidth associated with the corresponding service of one residential user.

3.2. Design of a SDN Management Layer to Control Service Reconfiguration Policies on the SDN-GPON Model

As it was said before, we have deployed an OpenFlow-GPON scenario to propose moving some time-critical service reconfiguration policies to an external SDN controller that provide a full SDN control of the GPON network operation and configuration. In order to do it, we have programmed an SDN management layer that interacts with the ODL controller, which is responsible for periodically monitoring the demanded bandwidth of residential users (ONTs traffic) and to apply real-time policies to efficiently reconfigure the maximum allocated bandwidth of their contracted services, as it can be observed in

Figure 2. Indeed, our approach configures Internet services and their associated maximum bandwidth to users according to real-time bandwidth requirements of users by means of flows that are sent by the SDN controller following the instructions of the SDN management layer. Our OpenFlow-based proposal provides several advantages for ISPs and network operators. First, although legacy OLT/ONT devices are built compliant with the GPON standard, many times the management software and the Application Programming Interfaces (APIs) provided by vendors to configure the network are quite different, thus causing interoperability and compatibility issues between the equipment of different vendors. Therefore, our OpenFlow proposal permits controlling the legacy GPON equipment of different vendors since every device (OLT, ONT) understand the same protocol, that is OpenFlow, due to the emulated SDN layer implemented on each of them, as it can be observed in Figure 2.
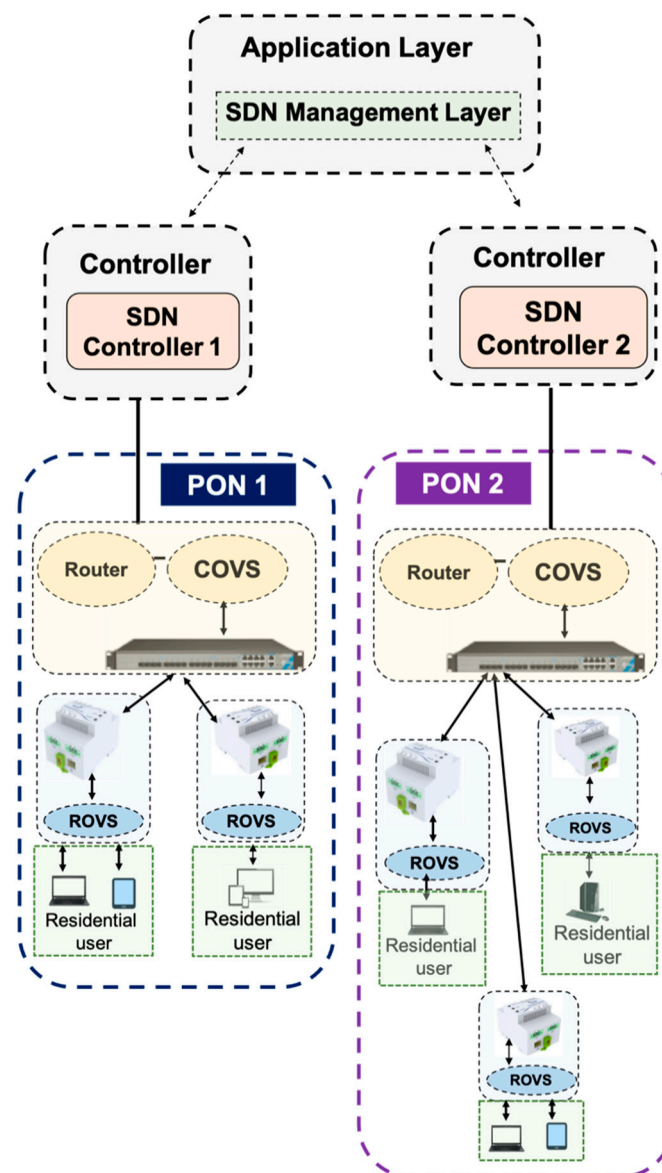

Figure 2. Block diagram of our OpenFlow approach over GPONs to apply external service configuration policies.

Secondly, our proposal allows simultaneously controlling several SDN controllers and GPONs using the same SDN management layer in a centralized, efficient, and dynamic way. Then, as it is shown in Figure 2, the SDN management layer can deal with a set of

GPONs sending configuration instructions to its corresponding SDN controller. Moreover, in our approach, any SDN controller capable of understanding the OpenFlow protocol can be deployed in the network scenario (ODL, ONOS), since the SDN management layer is able to interact with every controller. As a consequence, these advantages optimize the network operation and reduce its complexity, since our approach allows remotely controlling and allocating flows to provide and configure services in GPONs in a dynamic, programmable, centralized, and unified way.

Therefore, to carry out this dynamic control, the SDN management layer needs to know the packet statistics of every OVS (COVS, RVOS), so these statistics are periodically sent by every OVS to the ODL and the ODL to the SDN management layer. These statistics are sent in specific OpenFlow messages, called OFPM_METER messages, which contain the number of packets and bytes processed by the meters inside the OVS, the number of packets and bytes deleted, and the time the message is sent. Then, the SDN management layer is continually executing a dynamic bandwidth management program, which is listening to the messages sent by every OVS so when the program receives a message, it extracts the different statistics of the message together with the MAC address of the corresponding residential user (ONT). These data provide the real-time demanded bandwidth of each user, since one OVS is located at each residential home just before the ONT. In order to calculate the mean requested bandwidth using these OpenFlow statistics of the different residential OVSs (ROVSs), fixed sliding windows that store this information are used, one for each service of the residential subscriber. Each sliding window contains N samples (*win_size*), and its operation is represented in Figure 3. As it can be observed, each sample contains the total number of bytes sent by the ONT $\left(Bytes^{ONT_i,service_j}\right)$ of one specific service that passes over the ROVS, and the time in which that statistics was received $(t_N, t_{N+1}, t_{N+2}, \ldots)$. Each time a sample is inserted, the estimation of the mean requested bandwidth (in Mbps) for the specific residential user and service is updated with every sample of the window $\left(\overline{B_{request}^{ONT_i,service_j}}\right)$. To calculate the term $\left(\overline{B_{request}^{ONT_i,service_j}}\right)$, at a certain instant, the mean is done with all the $Bytes^{ONT_i,service_j}$ values contained in the samples and the time they were sent. Moreover, the first sample of the queue will be discarded when the maximum duration of the window is exceeded, that is, *win_size* samples. Sliding windows show a good performance, since they contain the updated requested bandwidth of residential users and their services.
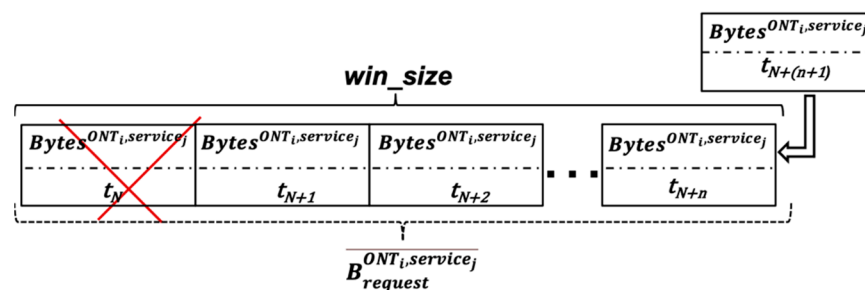

Figure 3. Operation of the sliding window implemented in the algorithm.

The process of assigning bandwidth to users (ONTs) and their services inside PON architectures is led by the OLT. Then, DBA algorithms (based on TDMA) are the most widespread methods, since the OLT allocates bandwidth to users in a dynamic way depending on the priority of the contracted services of users and its real-time demand. In this way, among the different bandwidth assignation disciplines, the DBA algorithm implemented in the OLT in our GPON testbed (as well as in legacy GPONs) follows a limited scheme, one of the most widespread policies in PONs due to its high efficiency and easy operation [29]. In this scheme, the OLT gives the required bandwidth to each service (*service_j*) of one ONT (*ONT_i*) in one cycle as long as its demand is lower than an established maximum

bandwidth $\left(B_{max}^{ONT_i,service_j} \right)$ in Mbps, which consists of a minimum guaranteed bandwidth $\left(B_{guaranteed}^{ONT_i,service_j} \right)$ that has to be ensured plus an extra best effort or non-guaranteed bandwidth $\left(B_{effort}^{ONT_i,service_j} \right)$, which is optional and only depends on the configuration made by the operator or service provider inside the DBA (Equation (1)).

$$B_{max}^{ONT_i,service_j} = B_{guaranteed}^{ONT_i,service_j} + B_{effort}^{ONT_i,service_j} \tag{1}$$

In case the user asks for more than this maximum, the algorithm will allocate at most this maximum value. In this way, the DBA algorithm in the OLT must ensure the guaranteed bandwidth and it may provide non-guaranteed bandwidth in case GPON resources are available. In contrast, if the demanded bandwidth is lower than the guaranteed bandwidth, the algorithm will offer this demand. Therefore, a limited scheme makes the cycle time adaptive depending on the updated demand of every ONT.

Therefore, in our OpenFlow approach, the bandwidth is not only controlled by the DBA algorithm in the OLT but is also managed by the SDN layer to provide a tighter control of this process. The SDN layer limits the maximum bandwidth for each residential user (and services) and the global available bandwidth at both channels of the GPON (upstream, downstream). In this way, the DBA is configured to treat residential users equally, but allocating bandwidth considering real-time traffic demands, as the differentiation and the fulfilment of bandwidth guarantees is provided by the emulated SDN layer implemented on GPON devices (OVS). Therefore, the OVSs (ROVS, COVS) drops those packets exceeding the maximum assigned bandwidth at each channel. In this way, the maximum associated with each service $\left(B_{max}^{ONT_i,servicej} \right)$ can be dynamically modified according to specific bandwidth allocation policies implemented by ISPs/Network Operators, allowing a real-time bandwidth reconfiguration according to the real-time requested bandwidth of residential users. Consequently, we propose an online service reconfiguration policy, in which users contract a guaranteed bandwidth $\left(B_{guaranteed}^{ONT_i,service_j} \right)$ and it can be offered an extra maximum according to the priority and conditions of the contracted Service Level Agreement (SLA) up to a maximum limit $\left(B_{max}^{ONT_i,servicej} \right)$. Therefore, users do not need to request more bandwidth whenever they need it, but it is the external SDN management layer that monitors changes and decides to increase or decrease the bandwidth of residential users at a given time. Under this network scenario, the external SDN management layer by means of the ODL controller transparently gives a user some bandwidth in excess if it demands it and there is enough capacity in the GPON. Then, a bandwidth management program (located in the SDN management layer) is constantly listening for the arrival of OpenFlow messages (OFM_METER) and collecting the value of the demanded bandwidth of the sliding window that corresponds to one service of one specific ONT, calculating the average of the said window in real-time, as it can be observed in the flowchart of Figure 4.

Every time the sliding window updates the mean requested bandwidth $\left(\overline{B_{request}^{ONT_i,service_j}} \right)$ an instance is launched to the service reconfiguration policy to detect if the average bandwidth of the window is higher than the current maximum bandwidth, called $B_{max,\ updated}^{ONT_i,servicej}$ in our algorithm. To do this, the algorithm first connects to the internal database that stores the maximum bandwidth $\left(B_{max}^{ONT_i,service_j} \right)$ and the guaranteed bandwidth of the contracted service. In the next step, the management program must assure the user an assigned rate always greater or equal than the basic contracted bandwidth $\left(B_{guaranteed}^{ONT_i,service_j} \right)$ in case the residential user demands it. Then, the program must verify that the requested bandwidth $\left(\overline{B_{request}^{ONT_i,service_j}} \right)$ does not exceed the maximum contracted bandwidth $\left(B_{max}^{ONT_i,service_j} \right)$. If all the conditions are met, the bandwidth management algorithm makes sure that the available resources are enough for the requested new bandwidth. In that case, the algorithm

modifies this value $\left(B_{max,updated}^{ONT_i,service_j} \right)$, updates the database, and sends a request to the ODL controller to add new flows to the corresponding ONT (COVS, ROVS) with the new QoS requirements and discard the existing flows with the old QoS requirements. Otherwise, the maximum bandwidth $\left(B_{max,updated}^{ONT_i,service_j} \right)$ is not updated and no commands are sent to the ODL controller.
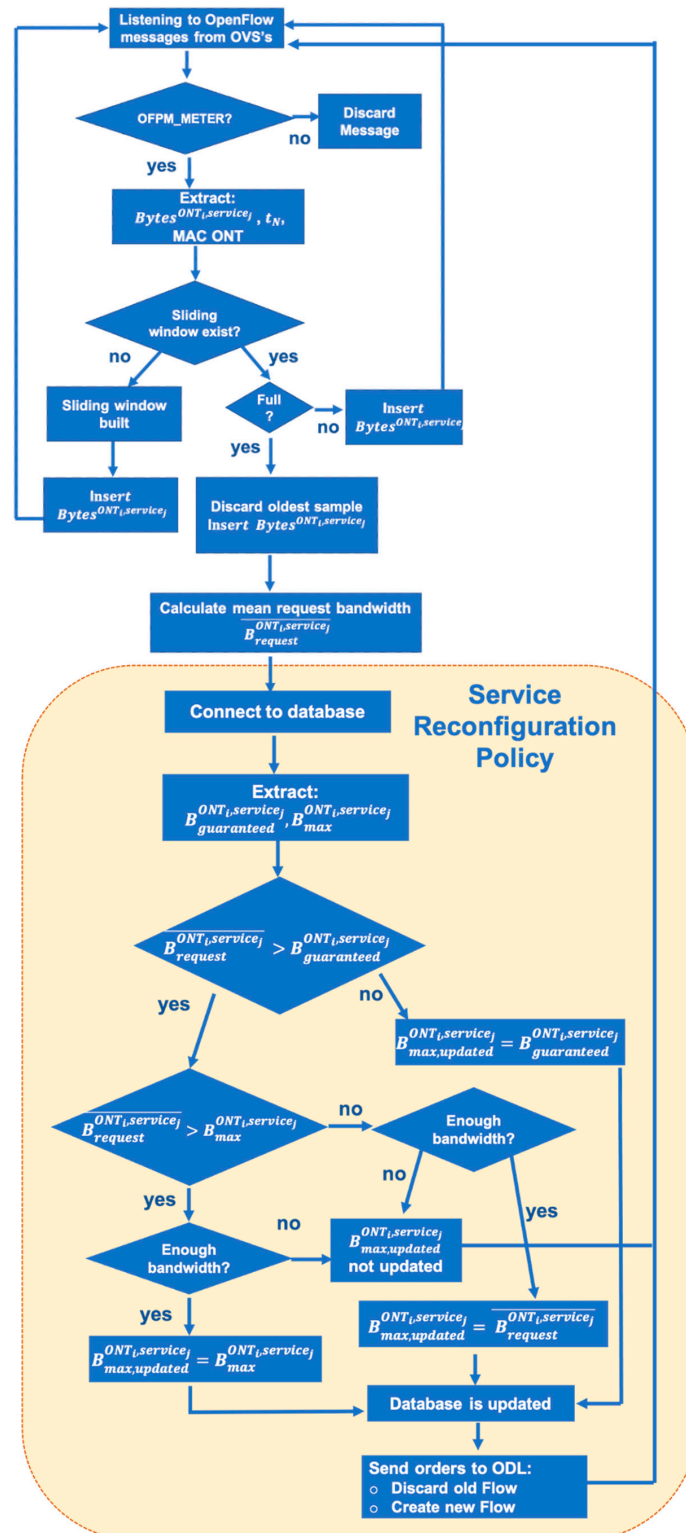

Figure 4. Flow chart of the implemented service reconfiguration policy.

Therefore, it could be very appealing that some online service management strategies can be moved out of the OLT to be centralized and controlled by the external SDN management layer. In addition, it avoids the synchronized time required in PON devices each time a service reconfiguration is made over the OLT and the ONTs. Then, we propose that the external SDN management layer could have implemented different service reconfiguration strategies. Finally, this approach avoids modifying the legacy GPON layer and it provides the full functionality of the integrated OpenFlow solution.

4. Experimental Setup and Results

To demonstrate the efficiency of our approach we have defined a new business model for ISPs/Network Operators and we check its viability using a legacy GPON. In fact, we have set up a network scenario which offers different priority SLAs based on flexible Internet plans so that users contract a fixed guaranteed bandwidth (basic bandwidth) plus an extra bandwidth, and when network subscribers demand this extra bandwidth and there are available resources, it can be given at the expense of a special pricing. Contrary to most of the proposed online service reconfiguration policies or DBA algorithms managed by SDN which are implemented with simulations, we have experimentally demonstrated our proposal using a GPON testbed with legacy equipment.

4.1. Description of the Legacy GPON Testbed

The commercial GPON (Figure 4) used to demonstrate the viability of our approach consists of an OLT (SmartOLT 350 of Telnet-RI vendor [30]), that supports 2.5 Gbps at the downstream channel and 1.25 Gbps at the upstream channel. The GPON testbed shows a tree topology configuration with a single splitting level. In this way, the OLT is connected to the optical splitter by means of three spools of Standard Single Mode Fiber (SMF) of different lengths (total length of 20 km). Then, the splitter (1:8) is connected to every ONT using SMF distribution fibers and the distance to each ONT is configurable, from 100 m up to 5 km, using a connection panel, as it can be noticed in Figure 5. This configurable network scenario allows setting up realistic situations where ONTs are located at different distances from the Central Office (OLTs). Finally, the GPON is equipped with L3 (with router functionalities) and L2 (without router functionalities) ONTs that comply with the ITU-T G.984.x/G.988 recommendations. The ONT models are the Wave Access 3021 (L3) and the Wave Access 512 (L2) [30]. Finally, in our particular approach, we attach computers (mini computers) to build routers with flexible capabilities by means of OVSs. For our experimental study, we connect five ONTs, two L2 ONTs, and three L3 ONTs.
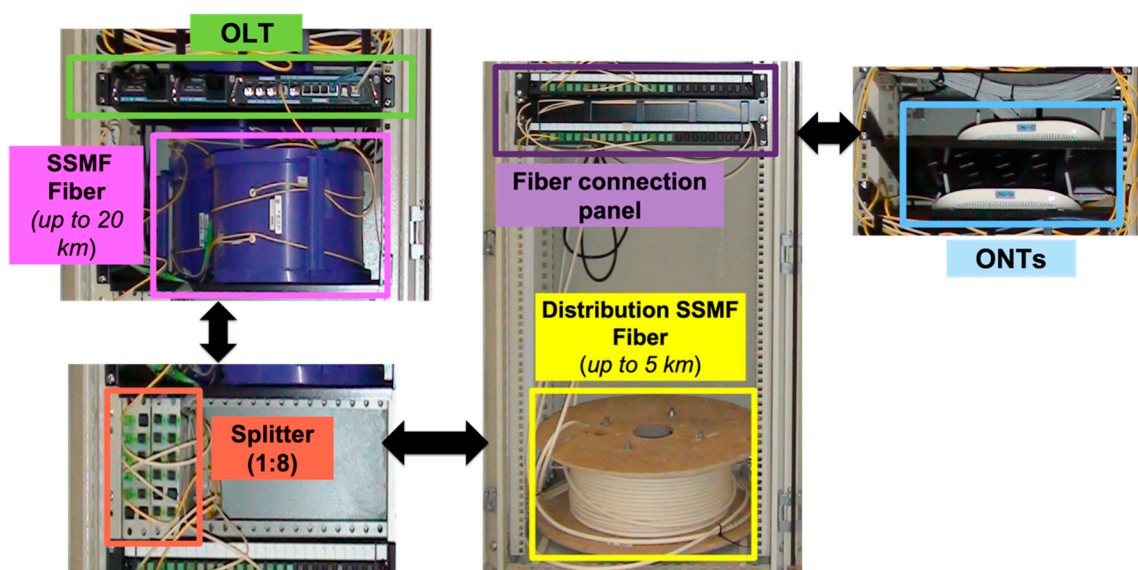

Figure 5. Real structure of the GPON testbed and its devices.

4.2. Description of the Case of Use

In our case of use, the policy that dynamically manages the maximum bandwidth allocated to services of residential users is implemented and demonstrated. As an example, we design Flexible Internet Plans differentiated by a basic bandwidth (guaranteed bandwidth) plus an excess bandwidth (Table 1). In case the basic bandwidth is below 100 Mbps, the excess bandwidth is considered as 20% of the basic (Basic Internet Plan), while if the basic bandwidth is higher than 100 Mbps, a 30% of excess bandwidth is considered (Premium Internet Plan). These maximum rates are calculated independently for the upstream and downstream bandwidth although they are symmetric. It should be noted, that although the proposed Internet plans in our case consider a certain amount of the basic bandwidth (20% and 30%), the operators and ISPs could stipulate other levels or conditions for the excess bandwidth. Then, these Flexible Internet plans ensure a fixed basic bandwidth to residential users plus an extra bandwidth, so that while users do not require this extra bandwidth it will be free to use transparently among other users who have greater bandwidth needs, so there is greater control over the total available bandwidth to distribute it in real-time and efficiently between all users, thus making the most of all network resources.

Table 1. Flexible Internet plans offered by Internet service providers (ISPs)/network operators.

Flexible Internet Plans	Basic Bandwidth	Excess Bandwidth
Basic Flexible Plan	≤100 Mbps	+20% Mbps
Premium Flexible Plan	>100 Mbps	+30% Mbps

4.3. Selection of Parameters in the Sliding Window

The first analysis regarding the sliding window used to calculate the average requested bandwidth of residential users, compared the response time of the dynamic bandwidth algorithm considering different window sizes. The optimal performance of the algorithm implies an optimum choice of some parameters, such as the size of the sliding window, which contains the last samples of the requested bandwidth of users. Then, if the number of samples is too large, the sliding window may contain quite old values of the requested bandwidth which can make the algorithm not react fast enough when changes in traffic conditions appear. In contrast, if the size of the sliding window is very short, the real-time estimation of the mean requested bandwidth cannot be reliable as the number of samples is not enough. Then, Figure 6a–d show the time that the algorithm takes to react when the associated bandwidth of one network subscriber (ONT) is being modified according to its real-time demands, considering different window sizes (5, 10, 20, and 30 samples). All the graphs are represented with Wireshark that can capture real- time network data [31]. As it can be observed in Figure 6c, the window of 20 samples has a response of approximately 16 s, while higher windows sizes such as 30 samples (Figure 6d) add a delay of approximately 20 s. In contrast, if low window sizes are considered, five and 10 samples (Figure 6a,b), the response time is lower, 8 and 12 s, respectively.

However, a too small window can be more sensitive to possible transmission peaks and fluctuations, as it can be seen in Figure 7. As a consequence, to test our dynamic service reconfiguration algorithm, a 10-samples window is chosen since the delay and the fluctuations are low and its performance is quite similar to high window sizes (20/30 samples).
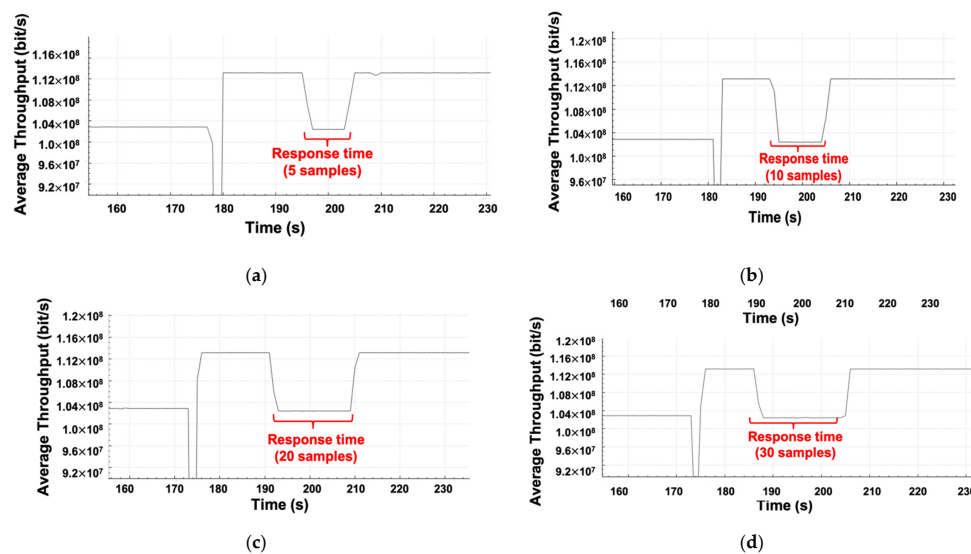
Figure 6. Test done on the size of the sliding window: (**a**) Five samples; (**b**) 10 samples; (**c**) 20 samples; (**d**) 30 samples.
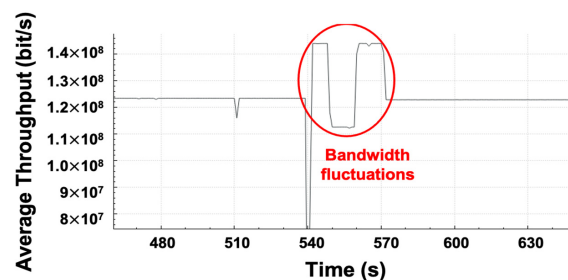

Figure 7. Real-time fluctuations on the average throughput due to a fall in the transmission rates.

4.4. Analysis and Results of the Algorithm Performance of the SDN Management Layer

To validate and analyze the performance of the algorithm experimentally, we have considered the availability of the two Flexible Internet Plans (Basic and Premium) previously mentioned, and we have assumed that a network subscriber (one ONT in the testbed) contracts a Flexible Internet Plan (Basic or Premium) and changes its real-time bandwidth demand, whereas the remaining four ONTs in the testbed contract a static Internet Plan. The experimental analysis was carried out at both channels, but in this paper, only the results of the upstream channel are presented for lack of space. However, the performance of the downstream channel is similar. This study should be considered as a proof of concept, since a typical PON will have a higher number of ONTs (e.g., 32–64) with varying demands than we have in our testbed. Nevertheless, to the best of our knowledge, this is the first time that a real-time reconfiguration service policy is experimentally tested in a legacy GPON using SDN techniques.

The Basic Flexible Internet plan consists of a guaranteed bandwidth of 100 Mbps plus an excess bandwidth of 20% (that is 20 Mbps), and the Premium Flexible Internet plan shows a guaranteed bandwidth of 150 Mbps with a 30% of excess bandwidth (45 Mbps), both symmetric services. Therefore, for the Basic Internet plan, as shown in Figure 8, the subscriber (ONT) that contracts this plan begins to transmit at 100 Mbps (60 s), and during the following intervals of 5 min (300 s) increases the transmission rate first to 110 Mbps, then to 120 Mbps, and finally to 130 Mbps, ending with a final period transmitting again at 100 Mbps. In addition, Table 2 indicates the transmitted bandwidth at the different interval times and the bandwidth expected to be offered to the residential user at each interval, according to the performance of the online service reconfiguration algorithm. Since the basic contracted bandwidth is equal to 100 Mbps (Table 1), the excess bandwidth is expected to increase up to 20% of this value (Basic Flexible Plan), so the maximum

bandwidth assigned to the user will be 120 Mbps, whenever the user demands it and there is enough available bandwidth. Therefore, in the fourth interval (660–960 s), although the user demands 130 Mbps, the maximum assigned bandwidth is limited to 120 Mbps (Table 2).
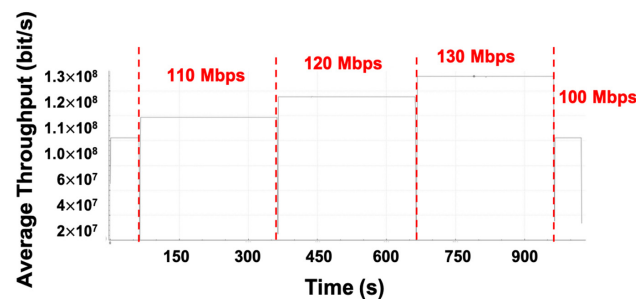

Figure 8. Real-time evolution of the requested bandwidth for the residential user that contract the basic flexible Internet plan.

Table 2. Demanded and received bandwidth for the basic flexible Internet plan.

Interval Time	User Transmission Rate (Mbps)	Expected Allocated Bandwidth (Mpbs)
0–60 s	100 Mbps	100 Mbps
60–360 s	110 Mbps	110 Mbps
360–660 s	120 Mbps	120 Mbps
660–960 s	130 Mbps	120 Mbps
960–1020 s	100 Mbps	100 Mbps

Then, Figure 9 shows the real-time evolution of the allocated bandwidth for the service of the ONT measured using Wireshark when the algorithm is working. Therefore, in the first Interval (0–60 s) the service is not modified, since the transmitted rate is equal to the basic contracted bandwidth (100 Mbps). In addition, it can be observed that at the beginning of each interval, a short step appears due to the reaction time of the OVS to a change in the user's transmission rate, as the OVS needs a few seconds (around 5–7 s) to notice an increase. If we analyze each interval one by one, in the second (60–360 s), where the residential user demands 110 Mbps, we observe the first step, followed by the time the algorithm needs to process the real-time demanded bandwidth of the sliding window, and then the instant where the algorithm provides the offered bandwidth to 110 Mbps (its bandwidth demand). The third Interval (360–660 s), where the user demands a rate of 120 Mbps, the result is similar and 120 Mbps are given (the maximum excess bandwidth). However, in the fourth interval (660–960 s) the residential user increases its demand up to 130 Mbps, but the algorithm limits this demand, since the maximum bandwidth associated with this Internet plan is 120 Mbps. Finally, in the last interval (960–1020 s), the user decreases its demand to 100 Mbps, so the algorithm assigns this bandwidth instantaneously. Consequently, it can be noticed that the algorithm implemented in the SDN layer shows a fast and stable response according to the contracted Internet plans of residential users. Moreover, it should be noted that DBA algorithms may need extra time to react to changes in the transmission rate of users as well as the OVS.
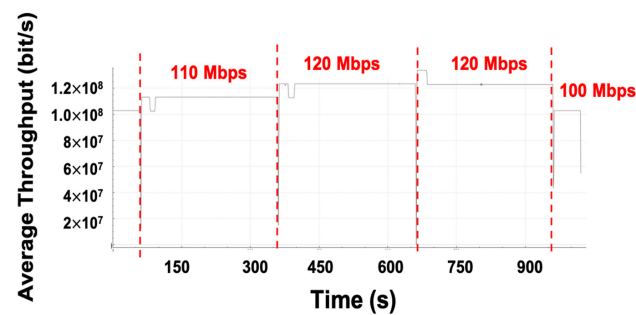
Figure 9. Real-time evolution of the allocated bandwidth for the residential user with the basic flexible Internet plan.

For the Premium Internet Plan, we consider a residential user that contracts an Internet service, which consists of a basic symmetric service of 150 Mbps plus an excess bandwidth of 30% (in that case 45 Mbps or more). However, in that case, the transmission rate of the residential user starts at 150 Mbps for 60 s and after that, it is increased in five min intervals (300 s). It has been considered that in each interval the bandwidth demand is increased by 20 Mbps, so that the evolution of the demanded bandwidth in the different period times is 150, 170, 190, 210, and finally 150 Mbps again. All these levels are reflected in Table 3, which represents the demanded bandwidth by the user in the different time intervals and the expected bandwidth that will be received according to the algorithm's performance.

Table 3. Demanded and received bandwidth for the premium flexible Internet plan.

Interval Time	User Transmission Rate (Mbps)	Expected Allocated Bandwidth (Mpbs)
0–60 s	150 Mbps	150 Mbps
60–360 s	170 Mbps	170 Mbps
360–660 s	190 Mbps	190 Mbps
660–960 s	210 Mbps	195 Mbps
960–1020 s	150 Mbps	150 Mbps

Unlike the previous case, now the residential user has a basic bandwidth higher than 100 Mbps, in which the corresponding Internet Plan may provide an excess bandwidth of 30%, so the maximum bandwidth that could be assigned is 195 Mbps, as observed in Table 3 in the interval 660–960 s. Therefore, the algorithm will create new flows based on the average bandwidth of the window, until reaching the maximum established of 195 Mbps. Figure 10 represents the real-time evolution of the allocated bandwidth for the service of the ONT measured using Wireshark. During the first period of change 60–360 s, in which the requested transmission rate is 170 Mbps, the algorithm shows two small steps and two transition zones between them just after offering this requested bandwidth. As it was mentioned before, the first step corresponds with the time required by the OVS to detect the change in the transmission rate of the user. Meanwhile, the algorithms process the data of the sliding window and send the order to the ODL controller to change the maximum bandwidth. Then, the second step corresponds with the time that the OVS needs to receive, process, and update the new flow sent by the ODL with the new transmission rate (meter) so the OVS becomes unstable during this period of time. Finally, the OVS updates the new maximum permitted bandwidth to 170 Mbps, that is the real- time demanded bandwidth. For the second period of change and the third, the performance of the algorithm is similar. Therefore, it can be noticed that the algorithm is able to dynamically evolve the allocated bandwidth according to the stipulated QoS requirements although the OVS needs a certain time to react to real-time changes in the transmission rate of users. In the same way, typical DBA algorithms inside OLTs probably also take time to deal with real-time changes in the transmitted traffic of residential users.
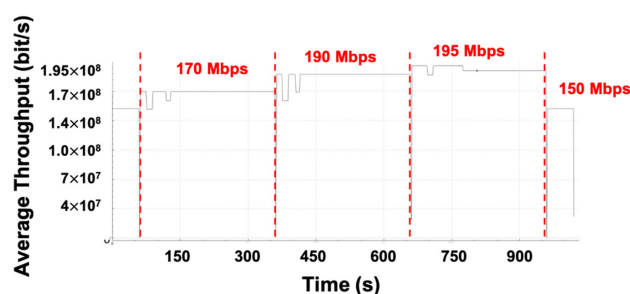
Figure 10. Real-time evolution of the allocated bandwidth for the residential user that contract the Premium Flexible Internet Plan.

5. Conclusions

In this paper, we have proposed and experimentally tested an OpenFlow-based approach over a legacy GPON to allow real-time service configuration policies applying SDN techniques. In this way, we have implemented an external SDN management layer able to communicate with SDN controllers, in this case ODL, to send orders through OpenFlow messages that configure and modify Internet services in residential users connected to the GPON. In particular, the SDN management layer can configure real-time Internet services and their associated maximum bandwidth to users according to their real-time demand and contracted bandwidth requirements. These service configuration policies do not require a cycle-by-cycle communication between the SDN management layer and the OLT, thereby not degrading performance despite the distance between the SDN infrastructure and the PON. Therefore, our approach provides a global network management and configuration of legacy GPONs using OpenFlow. Results have shown that the SDN management layer efficiently implements online service reconfiguration policies over residential users and their services demonstrate a good performance in terms of speed and efficiency, since it responds quite quick to real-time bandwidth requirements and fluctuations of bandwidth transitions are not very important.

This SDN proposal could provide great advantages for the GPON network and configuration. In fact, the approach allows ISPs or network operators to simultaneously control GPONs that belong to different providers, avoiding interacting with specific APIs of equipment from different manufacturers and minimizing compatibility problems. In our approach, every PON device can understand the OpenFlow protocol, since we have emulated a SDN layer over each device and consequently the SDN management layer interacts with SDN controllers (ODL in this case) in a transparent and automatic way. Even more, the implemented solution allows simultaneously controlling several SDN controllers and GPONs from the same SDN management layer in a centralized, efficient, and dynamic way. Finally, our approach allows moving global service configuration strategies out of the PON intelligence to an external SDN management layer which dynamically adjust these types of strategies according to the real-time QoS requirements, permitting ISPs/Network Operators to enable new business models for residential users. As a consequence, these advantages optimize the network operation and reduce its complexity, since it permits remotely controlling and configuring service profiles (Internet, VoIP, Video) in a dynamic, programmable, centralized, and unified way.

Author Contributions: Conceptualization, N.M., J.C.A., I.d.M., and R.J.D.; Methodology, N.M., J.C.A., I.d.M., R.J.D., P.F.; Software, D.d.P., N.M., and J.C.A.; Investigation, N.M., D.d.P., J.C.A.; Validation, I.d.M., R.J.D., P.F., R.M.L., and E.J.A., Writing—Original Draft Preparation, N.M., D.d.P., J.C.A., I.d.M., R.J.D., P.F.; Writing—Review & Editing, R.M.L. and E.J.A.; Supervision, R.M.L. and E.J.A.; Project Administration, N.M., R.J.D., I.d.M., and E.J.A. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kerpez, K.J.; Cioffi, J.M.; Ginis, G.; Goldburg, M.; Galli, S.; Silverman, P. Software-defined access networks. *IEEE Commun. Mag.* **2014**, *52*, 152–159. [CrossRef]
2. Thyagaturu, A.S.; Mercian, A.; McGarry, M.P.; Reisslein, M.; Kellerer, W. Software Defined Optical Networks (SDONs): A Comprehensive Survey. *IEEE Commun. Surv. Tutorials* **2016**, *18*, 2738–2786. [CrossRef]
3. Open Networking Foundation (ONF). Available online: https://www.opennetworking.org/ (accessed on 3 October 2020).
4. NETCONF Configuration Protocol. Network Configuration Working Group. Available online: https://tools.ietf.org/wg/netconf (accessed on 3 October 2020).
5. RESTCONF Protocol. Internet Engineering Task Force (IETF). Available online: https://tools.ietf.org/html/rfc8040 (accessed on 3 October 2020).
6. State of Fibre. New Market Forecast 2020–2026 Revealed. Available online: https://www.ftthcouncil.eu/home/latest-news/state-of-fibre-new-market-forecasts-2020-2026-revealed?news_id=3863&back=/resources/key-publications (accessed on 10 December 2020).
7. Europe Broadband Status Market Forecast by 2020–2025, FTTH Council. Available online: https://www.ftthcouncil.eu/documents/Reports/2019/FTTH%20Council%20Europe%20%20Forecast%20for%20EUROPE%202020-2025.pdf (accessed on 1 October 2020).
8. Lee, S.S.W.; Li, K.-Y.; Wu, M.-S. Design and implementation of a GPON-based virtual Open Flow-enabled SDN switch. IEEE/OSA. *J. Lightwave Technol.* **2016**, *34*, 2552–2561. [CrossRef]
9. Rouskas, G.N.; Dutta, R.; Baldine, I. A New Internet Architecture to Enable Software Defined Optics and Evolving Optical Switching Models. In Proceedings of the International Conference Broadband Communications Network Systems (BROADNETS), London, UK, 8–11 September 2008; pp. 71–76.
10. Gu, R.; Ji, Y.; Wei, P.; Zhang, S. Software defined flexible and efficient passive optical networks for intra-datacenter communications. *Opt. Switch. Netw.* **2014**, *14*, 289–302. [CrossRef]
11. Lee, Y.; Kim, Y. *A Design of 10 Gigabit Capable Passive Optical Network (XG-PON1) Architecture Based on Software De-Fined Network (SDN)*; (ICOIN): Siem Reap, Cambodia, 2015.
12. Amokrane, A.; Hwang, J.; Xiao, J.; Anerousis, N. Software defined enterprise passive optical network. In Proceedings of the 10th International Conference on Network and Service Management (CNSM) and Workshop, Rio de Janeiro, Brazil, 17–21 November 2014; pp. 406–411.
13. Parol, P.; Pawlowski, M. Towards networks of the future: SDN paradigm introduction to PON networking for business applications. In Proceedings of the 2013 Federated Conference on Computer Science and Information Systems), Kraków, Poland, 8–11 September 2013.
14. The Open Source Project VOLTHA. Available online: https://www.opennetworking.org/voltha/ (accessed on 3 October 2020).
15. Khalili, H.; Sallent, S.; Piney, J.R.; Rincón, D. A proposal for an SDN-based SIEPON architecture. *Opt. Commun.* **2017**, *403*, 9–21. [CrossRef]
16. Lia, C.; Guoa, W.; Wanga, W.; Hua, W.; Xiab, M. Programmable bandwidth management in software-defined EPON architecture. *Opt. Commun.* **2016**, *370*, 43–48. [CrossRef]
17. Zhang, J.; Zhao, Y.; Jialin, W.; Ji, Y.; Yi, L.; Han, J.; Lee, Y. Experimental demonstration of remote unified control for Open-Flow-based software-defined optical access networks. *Photonic Netw. Commun.* **2016**, *31*, 568–577.
18. Wang, F.; Liu, B.; Zhang, L.; Jin, F.; Zhang, O.; Tian, Q.; Tian, F.; Rao, L.; Xin, X. Dynamic bandwidth allocation based on multiservice in software-defined wavelength-division multiplexing time-division multiplexing passive optical net-work. *Opt. Eng.* **2017**, *56*, 036104. [CrossRef]
19. Hwang, I.-S.; Rianto, A.; Pakpahan, A.F. Software-defined Peer-to-Peer file sharing architecture for TWDM PON. In Proceedings of the 2018 27th Wireless and Optical Communication Conference (WOCC), Hualien, Taiwan, 30 April–1 May 2018; pp. 1–4.
20. Ruffin, M.; Slyne, F.; Bluemm, C.; Kitsuwan, N.; McGettrick, S. Software defined networking for next generation con-verged metro-access networks. *Opt. Fiber Technol.* **2015**, *26*, 31–41. [CrossRef]
21. Yan, B.; Zhou, J.; Wu, J.; Zhao, Y. Poster: SDN based energy management system for optical access network. In Proceedings of the 9th International Conference on Communications and Networking in China, Shanghai, China, 14–16 August 2014; pp. 658–659.
22. Quian, C.; Li, Y.; Zhang, O.; Cao, B.; Li, Z.; Wang, M. Staged priority-based dynamic bandwidth allocation in soft-ware-defined hybrid passive optical network. *Opt. Eng.* **2018**, *57*, 126101.

23. Zhou, L.; Peng, G.; Chand, N. Demonstration of a novel software-defined Flex PON. *Photonic Netw. Commun.* **2015**, *29*, 282–290. [CrossRef]
24. Open vSwitch. Available online: https://www.openvswitch.org/ (accessed on 15 November 2020).
25. Open Day Light SDN Controller. Available online: https://www.opendaylight.org/ (accessed on 15 November 2020).
26. Salman, O.; Elhajj, I.H.; Kayssi, A.; Chehab, A. SDN Controllers: A Comparative Study. In Proceedings of the 18th Mediter-ranean Electrotechnical Conference (MELECON), Lemesos, Cyprus, 18–20 April 2016.
27. Paliwal, M.; Shrimankar, D.; Tembhurne, O. Controllers in SDN: A Review Report. *IEEE Access* **2018**, *6*, 36256–36270. [CrossRef]
28. OpenFlow Switch Specification and Meters, OpenFlow Switch Specification. Available online: https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf (accessed on 15 November 2020).
29. Kramer, G.; Mukherjee, B.; Dixit, S.; Ye, Y.; Hirth, R. Supporting differentiated classes of service in Ethernet passive optical networks. *J. Opt. Netw.* **2002**, *1*, 280–298.
30. Telnet Vendor Web Page, Telnet R-I. Available online: https://www.telnet-ri.es/ (accessed on 4 October 2020).
31. Wireshark: Network Protocol Analyzer. Available online: https://www.wireshark.org/ (accessed on 4 October 2020).