Dynamic bandwidth control and testbed validation of a SDN based GPON

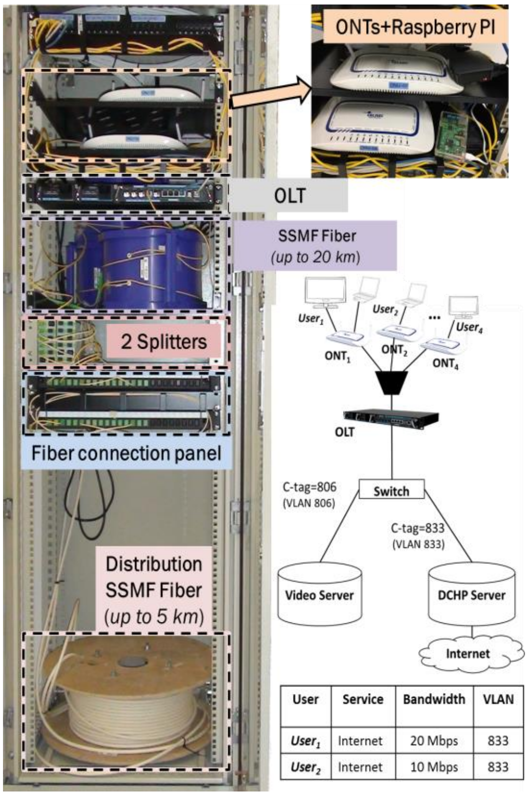
J. Azofra(1), N. Merayo(1), J.C. Aguado(1), I. de Miguel(1), R. Durán(1), F.J. Ruiz(2), C. Vega(2), D. Fernández(2), P. Fernández(1), R.M. Lorenzo(1), E.J. Abril(1)

(1) Grupo de Comunicaciones Ópticas, Universidad de Valladolid, Spain, noemer@tel.uva.es
(2) Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, Spain

Abstract *An end-to-end SDN based GPON scenario to provide fast, efficient and accurate QoS management is proposed and validated experimentally. It is designed to support a novel network management model that permits residential users to control their bandwidth network resources.*

Introduction

Software defined networking (SDN) abstraction and separation of the control and data planes improves the efficiency of networks, allowing a better control of network traffic, management automation and scalability, leading to reduction of operating and hardware costs [1]. The integration of SDN in passive optical networks (PONs) is gaining momentum, as demonstrated by proposals to embed an OpenFlow agent in the optical line terminal (OLT) of a GPON, which communicates with an external OpenFlow controller [2,3], the development of modules to map OpenFlow messages to PON configuration commands [4], or the definition of OpenFlow extensions for GPON systems [5]. In this paper, we present an OpenFlow-based solution, validated in a testbed, to ensure fulfilment of QoS requirements in GPON. In contrast to other works, we associate Openflow virtual switches at the OLT but also at the optical network terminals (ONTs). This approach has been designed to integrate with a new network management model based on a user-centric approach. The model allows residential users to manage their traffic flows and to define and control network resources by means of management apps [6].



User	Service	Bandwidth	VLAN
User$_1$	Internet	20 Mbps	833
User$_2$	Internet	10 Mbps	833

Fig. 1: Deployment and configuration of the GPON.

GPON testbed scenario

A GPON testbed (Fig. 1) has been implemented using equipment from the Telnet-RI vendor [7]. It includes an OLT SmartOLT 350, which implements a full-duplex GPON interface of 2.488 Gbps (downstream) and 1.244 Gbps (upstream), fully compliant with ITU-T G.984.x specifications. The connection between the OLT and the optical splitter includes three spools of Standard Single Mode Fiber (SSMF) of different lengths. The testbed is equipped with two optical splitters 1:8, thus enabling the configuration of a two-stage splitting topology if desired. Then, the splitters are connected to the ONTs by distribution fibers. The length of each link can be individually configured, using a connection panel, from 100 meters up to 5 km, so that the testbed can emulate realistic scenarios (different distance to the central office). The testbed employs L3 model ONTs (Wave Access 3021), which include router functionalities and comply with ITU-T G.984.x/G.988.

Implementation of SDN in the GPON testbed

Taking advantage of SDN accurate QoS control, we propose to manage GPON network services with OpenFlow using an external OpenDayLight (ODL) controller and several OVS (OpenFlow Virtual Switches) connected at the OLT and the ONTs (Fig. 2). ODL controls downstream and upstream traffic channels so that services comply with QoS requirements (guaranteed bandwidths), as well as VLAN identifiers associated with services but providing a more

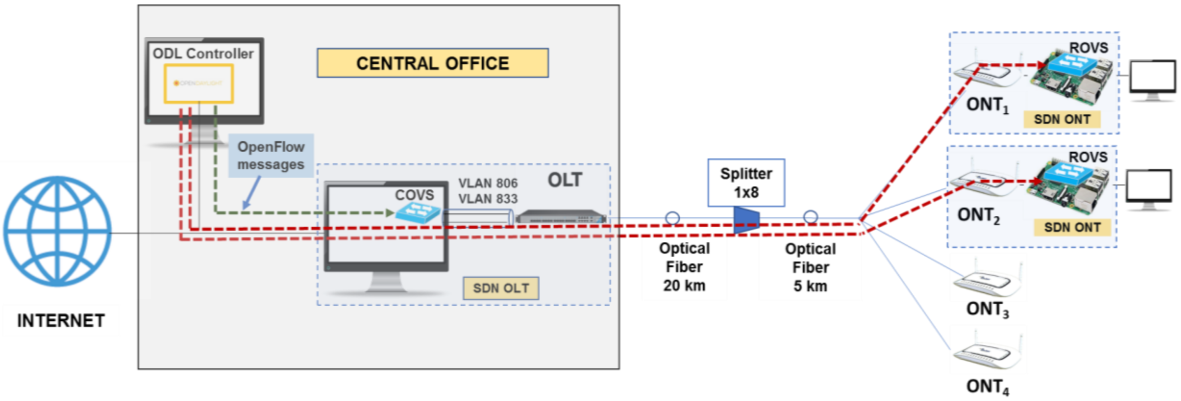


Fig. 2: The OpenFlow SDN scenario implemented in the GPON testbed.

flexible and dynamic control. Besides, the controller is able to dynamically modify services according to real network traffic or user requirements, allowing a powerful control of the GPON capabilities. For demonstration purposes, an OVS running in a computer is inserted just before the OLT (Central OVS, COVS) (Fig. 2) to manage the downstream traffic of user's services. Another OVS is deployed at the users' side (Remote OVS, ROVS) to control the upstream QoS requirements of their associated services. Such information is sent by the ODL controller to every OVS using OpenFlow messages through the conventional GPON channel. In a real implementation, the COVS and ROVS should be integrated inside the OLT and the ONT respectively (SDN based OLT/ ONT). However, as SDN based ONTs are not available, we use a Raspberry Pi to embed the ROVS between the user and the ONT (Fig. 2).

One important issue is the definition of the flow table sent by the ODL controller to the different OVS's. Two flows entries are created for each service (Fig. 3): one for the downstream traffic flow (COVS) and another for the upstream traffic flow (ROVS). The match instruction and fields of each flow differ depending on the channel (Fig. 3). Besides, we use OpenFlow meters to measure and control the rate of packets assigned to each flow. For each service we attach one meter to each flow, one to control the maximum upstream bandwidth and another for the maximum downstream bandwidth. Each meter entry consists of a meter identifier and the meter bands (Fig. 3). Each meter band contains the band type (how packets are processed), the band-rate (rate that, if exceeded, triggers the processing of packets according to the band type) and some optional arguments. The controller can query the switch about which of the "Optional" meter band types it supports. We use the action "Drop", which defines a rate limiter band, and packets are discarded if that rate is exceeded.

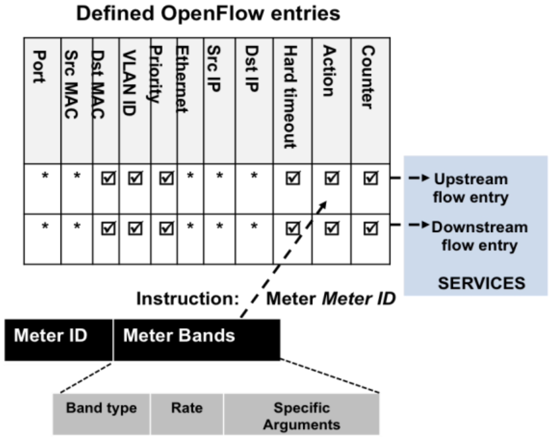
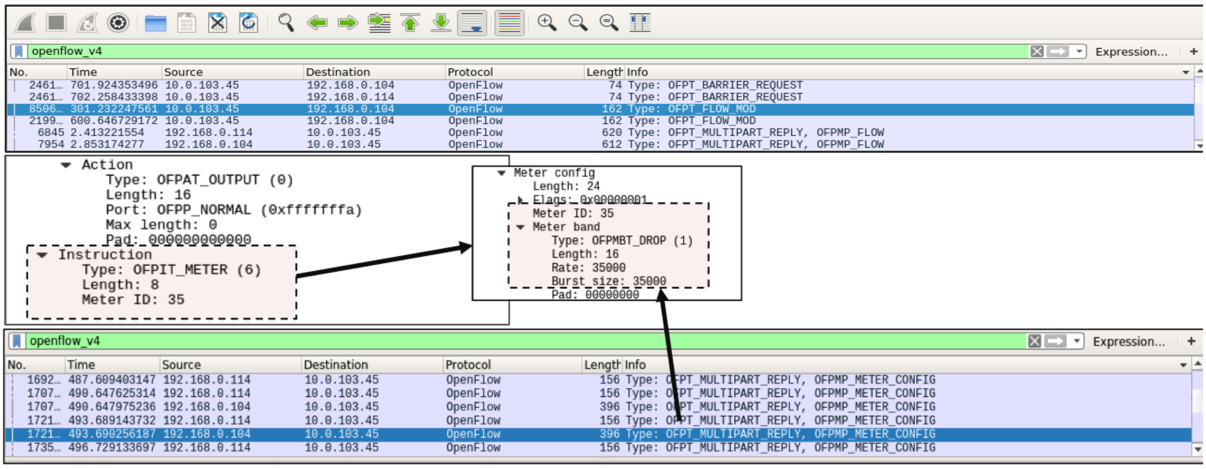


Fig. 3: Definition of the OpenFlow entries for each service.

Experimental scenario and results

We have setup a dynamic network scenario where the operator has a business model in which users have a contracted basic bandwidth, but are allowed to increase it temporarily, e.g., when using highly demanding services such as online games or Ultra High Definition television. An example of the initial test configuration of the GPON is shown in Fig. 1 (right). For $User_1$ (ONT_1) we define an Internet service with a guaranteed bandwidth of 20 Mbps (symmetric) and for $User_2$ (ONT_2) another service of 10 Mbps (symmetric). To differentiate services, VLAN tagging is used, in particular 833 for the Internet service.

The ODL controller builds the configuration and sends the flow entries to the corresponding OVS's to manage the service traffic. For each service the controller sends OpenFlow messages to the COVS and ROVS, each of them associated with the initial corresponding meter. For ONT_1 we set a meter with the band type to 20 Mbps (meter ID=20) and for ONT_2 10 Mbps (meter ID=10). In order to demonstrate the dynamicity in bandwidth request and allocation, we set that during five minutes (300 s), both



```
openflow_v4                                                          Expression...  +
No.     Time          Source         Destination    Protocol  Length Info
 2461…  701.924353496 10.0.103.45    192.168.0.104  OpenFlow      74 Type: OFPT_BARRIER_REQUEST
 2461…  702.258433398 10.0.103.45    192.168.0.114  OpenFlow      74 Type: OFPT_BARRIER_REQUEST
 8506…  301.232247561 10.0.103.45    192.168.0.104  OpenFlow     162 Type: OFPT_FLOW_MOD
 2199…  600.646729172 10.0.103.45    192.168.0.104  OpenFlow     162 Type: OFPT_FLOW_MOD
  6845  2.413221554   192.168.0.114  10.0.103.45    OpenFlow     620 Type: OFPT_MULTIPART_REPLY, OFPMP_FLOW
  7954  2.853174277   192.168.0.104  10.0.103.45    OpenFlow     612 Type: OFPT_MULTIPART_REPLY, OFPMP_FLOW
```

```
    ▾ Action
        Type: OFPAT_OUTPUT (0)
        Length: 16
        Port: OFPP_NORMAL (0xfffffffa)
        Max length: 0
        Pad: 000000000000
  ▾ Instruction
      Type: OFPIT_METER (6)
      Length: 8
      Meter ID: 35
```

```
▾ Meter config
    Length: 24
  ▸ Flags: 0x00000001
    Meter ID: 35
  ▾ Meter band
      Type: OFPMBT_DROP (1)
      Length: 16
      Rate: 35000
      Burst_size: 35000
      Pad: 00000000
```

```
openflow_v4                                                          Expression...  +
No.     Time          Source         Destination    Protocol  Length Info
 1692…  487.609403147 192.168.0.114  10.0.103.45    OpenFlow     156 Type: OFPT_MULTIPART_REPLY, OFPMP_METER_CONFIG
 1707…  490.647625314 192.168.0.114  10.0.103.45    OpenFlow     156 Type: OFPT_MULTIPART_REPLY, OFPMP_METER_CONFIG
 1707…  490.647975236 192.168.0.104  10.0.103.45    OpenFlow     396 Type: OFPT_MULTIPART_REPLY, OFPMP_METER_CONFIG
 1721…  493.689143732 192.168.0.114  10.0.103.45    OpenFlow     156 Type: OFPT_MULTIPART_REPLY, OFPMP_METER_CONFIG
 1721…  493.690256187 192.168.0.104  10.0.103.45    OpenFlow     396 Type: OFPT_MULTIPART_REPLY, OFPMP_METER_CONFIG
 1735…  496.729133697 192.168.0.114  10.0.103.45    OpenFlow     156 Type: OFPT_MULTIPART_REPLY, OFPMP_METER_CONFIG
```

Fig. 4: Openflow messages of the upstream services sent to ONTs.

users require (by means of a management app) to increase their guaranteed bandwith to 35 Mbps and 20 Mbps, respectively. Once that period expires (or when the user cancels it), they return to their initial configuration. Fig. 4 shows a Wireshark capture of OpenFlow messages. It can be noticed that for ONT$_1$ the controller sends at second 300 a FTP_FLOW_MOD message with the instruction to increase its bandwidth to 35 Mbps (meter ID=35). As it can be observed, this information is contained in a OFMP_METER_CONFIG message. Fig. 5 shows the measured throughput for the upstream services for both ONTs. It can be seen that at second 300, both ONTs increase their bandwidth to the requested values, and the transition to the new levels is very fast (around 2 seconds) and accurate. After that, in second 600 (when the 5 minutes period expires), they return to their initial bandwidth configurations.

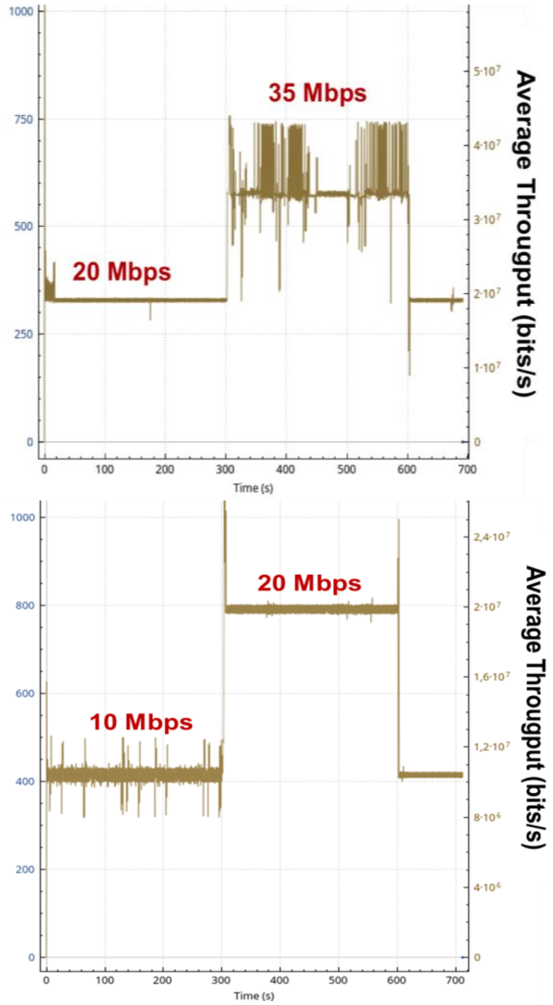


Fig. 5: Wireshark packet capture of the upstream services of ONT$_1$ and ONT$_2$.

Conclusions

We have proposed, and experimentally demonstrated, an end-to-end SDN solution for GPONs, which allows residential users to efficiently and dynamically adjust their bandwidth demands, thus enabling new business models for the network operator.

Acknowledgements

Funded by Spanish Ministry of Science & Innovation TEC2014-53071-C3-2-P, TEC2015-71932-REDT, TEC2017-84423-C3-1-P and TEC2015-67834-R

References

[1] A. S. Thyagaturu et al, “Software Defined Optical Networks (SDONs): A Comprehensive Survey,” IEEE Communications Surveys & Tutorials, vol. **18**, (2016).

[2] S. W. Lee et al., “Design and implementation of a GPON-based virtual OpenFlow-enabled SDN switch,” J. Lightw.Technol., vol. **34**, no. 10, p. 2552 (2016).

[3] R. Gu et al., “Software defined flexible and efficient passive optical networks for intra-datacenter Communications,” Opt. Switching Netw., vol. **14**, (2014).

[4] A. Amokran, et al., “Software defined enterprise passive optical network,” Proc. IEEE ICNSM, Rio de Jan. (2014).

[5] P. Parol et al., “Towards networks of the future: SDN paradigm introduction to PON networking for business applications,” Proc. FedCSIS, Poland (2013).

[6] R. Flores et al., “A user-centric SDN management architecture for NFV-based residential networks”, Comp. Standards & Interfaces, Vol. 54, (2017).

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