JST: Smart Systems and Devices
Volume 32, Issue 1, January 2022, 051-058

Toward Fully Virtualization of the Gigabit Passive Optical Networks (GPON)

Nguyen Tai Hung
School of Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi, Vietnam
Email: hung.nguyentai@hust.edu.vn

Abstract

The automation of access optical networks is becoming more and more important in the research literature, and how to apply the software defined network (SDN) and network functions virtualization (NFV) is being now intensively investigated. In the specific case of PON networks, given the fact that the OLT houses all of the controls and intelligent functionalities, application of the centralized control mechanism envisioned by SDN/NFV makes even more sense. However, the aspects related to service automation and management when transforming the traditional GPON networks into softwarized and virtualized architecture, are relatively unexplored. Furthermore, nowadays, the sharing of access infrastructure between the different service networks (5G, IoT, Broadband Internet, etc.) is the trend and obligation in order to reduce significantly both CAPEX & OPEX for telecom operators and, in that circumstance, the virtualized and centralized-management access network architecture can give a big help. In this paper, we present our proof-of-concept implementation of the GPON network (OLT in exact) functions and services in SDN-based and virtualized environments. Proposed architectures, implementation details as well as functional and performance test results will be presented and discussed.

Keywords: GPON (gigabit passive optical network), SDN, NFV, kubernete, openflow.

1. Introduction

Passive Optical Networks [1] are currently used in access networks. In theory, there are several optical network technologies but the most widely deployed are EPON [2] and GPON [3]. As shown in Fig. 1, PON is composed of a central office (Optical Line Terminal, OLT), passive optical splitters, and one terminal (Optical Network Unit, ONU/ONT) at each one of the customer premises, in a tree topology. First-generation PONs use Time Division Multiplexing (TDM) for sharing the medium in a point-to-multipoint scenario but nowadays most of PON systems such as GPON use Ethernet as an uplink standard. In the downstream, the OLT broadcasts the data frames towards the ONTs, while in the upstream an arbitration mechanism is needed in order to avoid collisions when ONTs send frames to the OLT. The GPON standard, which will be the focus of this work, uses the Ethernet format for the data frames.

In reality, the deployment and management of access networks, like GPON, are very complicated and costly processes. Thus, there is a strong demand for sharing the (optical) access infrastructure nowadays. However, sharing of optical access networks is today not common across the world. In places where sharing is in place, the approach has typically been limited either to fiber unbundling, that is, where the optical access is point-to-point fiber, or to higher layer NGA bitstream. In NG-PON2, where multiple wavelengths are available, it is possible, in principle, to separate OLOs by wavelength. However, there are still a number of technical issues. One issue relates to the ownership of the Optical Line Terminations (OLTs), as multiple wavelengths could interfere in certain cases, if not controlled by the same system. Another issue is that the allocation of a PON wavelength to one OLO is static and inefficient, preventing capacity unused by one PON to be used by other operators. Furthermore, NG-PON2 currently defines only 8 wavelengths for Time Division Multiplexing (TDM) access and the technology is not yet widespread, due to the high cost of the end-users tunable Optical Network Terminals (ONTs). More dynamic techniques have been discussed, for example in [4, 5], and could be made technically feasible with the recent development of SDN and NFV in the central office [6].

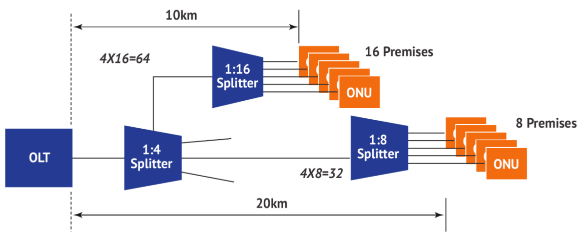

Fig. 1. A typical GPON network

ISSN: 2734-9373
https://doi.org/10.51316/jst.155.ssad.2022.32.1.7
Received: September 17, 2021; accepted: November 2, 2021

In the rest of this section, we briefly introduce some of the prevailing optical access technologies and review their major challenges with respect to multi-tenancy (meaning, model for sharing optical infrastructure amongst operators).

Recently, technology evolutions based on Software Defined Network (SDN) and Network Function Virtualization (NFV) have enabled network multi-tenancy that increases the flexibility and level of network control automation and management processes, in ways that were not possible before. These characteristics arise because of virtualization, which enables different entities to get access to a subset of the network resources while giving the illusion of fully owning that part of the infrastructure. This separates the operations of one tenant fully from other tenants that share the same optical infrastructure.

Software Defined Networking (SDN) [7] is a new paradigm for network management that introduces network programmability and separation of the data and control planes, migrating the complex, "intelligent" functions from the network devices to a centralized controller. These characteristics allow the SDN controller to obtain a centralized and accurate view of the state of the network, thus opening the way for the optimization in several ways (bandwidth assignment, QoS guarantees, minimization of power consumption, or resilience, to name a few possibilities). The SDN architecture defines two interfaces in the controller. The northbound interface defines an API that allows network management systems (NMS) or ad-hoc applications to communicate with the controller. The southbound interface communicates the control plane that runs in the controller with the data plane that resides in the network equipment (switch). The original southbound API, and probably the most popular, is OpenFlow (OF). The OpenFlow protocol [8] defines a standardized instruction set that allows the controller to manage any OpenFlow-enabled device and specify the path to be followed by traffic flows through the network of switches, by means of the definition of matching rules (based on packet header fields) and operations to be performed (e.g. forwarding to a specified port, modifying the headers, or dropping the packet). But Openflow is not the only option to control the access switches (very diversified in reality), that's why in this paper, we will propose a comprehensive architecture that supports several south-bound protocols in order to meet with various types of the hardware switches.

Also, on this paper, the virtualized GPON OLT architecture will be proposed together with a proof-of-concept implementation details. The provided results are, though, open for discussion of further researches. The article includes the following section: Session 2 presents the recent researches on the field of applying SDN/NFV technologies to the access networks. The proposed architecture and PoC implementation details are demonstrated in session 3. Functional and performance test results are given in section 4, and finally conclusions about the research as well as future development directions are in the last session.

2. Related Works

Initial access network virtualization concepts were introduced in [9], with a Software Defined Access Network (SDAN), where access network management and control functions are virtualized. This can speed up service creation, streamline operations and enhance customer satisfaction in multi-operator environments. SDAN works with port-level, physical cable, and logical bit stream unbundling. SDAN moves storage and computing functions from Network Elements (NEs) to the controller and provides a common interface to control functions accessed by multiple operators.

The authors of [10] developed an architecture for PON networks based on SDN including an OpenFlow extension for traffic mapping and forwarding capabilities, with no effect on data link layer latency. In [11] an architecture control plane for converged metro-access networks under SDN is described. [12] presents a novel software-defined optical access network (SDOAN) architecture. The purpose of developing a Service-Aware Flow Scheduling (SA-FS) strategy is to assign network bandwidth resources in an efficient and flexible way. In another project [13], the authors describe a GPON architecture where an OpenFlow agent is located in the OLT to communicate with the SDN controller, claiming that the approach can connect several sites in different locations in a cost effective manner. Central Office Re-architected as a Datacenter (CORD) is a novel architecture developed in a project of big telecom operators like AT&T, NTT, etc. aimed at substituting the telephone exchange hardware with software-based equipment, converting central offices into datacenters, with the purpose of speeding the deployment and increasing the efficiency of services. CORD decouples the control and data planes, using the Open Network Operating System (ONOS) SDN controller, and virtual machines running on top of OpenStack. CORD is currently supported by service providers such as AT&T and NTT Communications.

As we can see, most of the related works so far concentrated on the inter-connection between PON components and the SDN controller in order to make centralized control access network thus making the sharing of optical infrastructure amongst operators possible. In contrast, our work is to fully make the GPON OLT to be virtualized that will bring most of its functions to the cloud. The purpose is to help making the lifecycle of access services automated and thus reducing heavily the CAPEX and OPEX for the access operators.

3. The Virtualized OLT

3.1. Proposed Architecture

Traditionally, telecom operators running a lot of so called Center Office (CO) that are housing dozen types of access equipment like the switches, OLT, BTS/eNodeB, etc. together with the outside plant for the cabling, the access networks connecting to the end user CPEs (Customer Premise Equipment), as illustrated in Fig. 2.

From the figure we can see that, for optical access network, there are three main systems at CO, namely OLT, Ethernet Aggregation Switches and Broadband Gateway (BRAS), which together, will provide the functions of connectivity termination, user/connection authentication as well as aggregation of all access sessions to the backbone network. And as stated above, in the 5G/IoT era where these are combined, demands of communication service quality will diversify drawing attention to a network virtualization technology capable of simultaneously delivering multiple quality requirements on a single physical (optical) infrastructure.

In network virtualization theory, the optical access network, metro network, core network, edge server and cloud server hardware are abstracted with a SDN controller and NFV-MANO, which are currently being subjects to be further studied in network virtualization field. Then, using the various abstracted virtual functions, the orchestrator constructs virtual networks with different requirements and provides them to the users. Various IoT services (automated driving, remote control, smart city, agricultural automation, ultra-reality, digital cinema, etc.) with different requirements can be realized on these virtual networks.

On this work, we're only focused on the research and development of optical access network virtualization, OLT in exact. Fig. 3 shows our comprehensive framework for GPON network virtualization. There are two main components on this framework are ONOS which will function as SDN controller and various white-boxes hardware of ONU, OLT, AGG (Aggregation switches). The proposed control protocols (south-bound i/f) are diversified to accommodate the various types of HW equipment's from multiple standards/vendors, though, we only implemented two main protocols namely Openflow (OF) and P4, on our test-bed. Furthermore, on this framework, all control software are virtualized and deployed as Containers using Kubernetes technology.

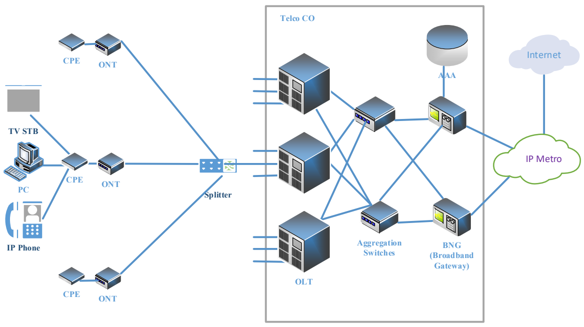

Fig. 2. The Central Office in traditional network architecture of telecom operators

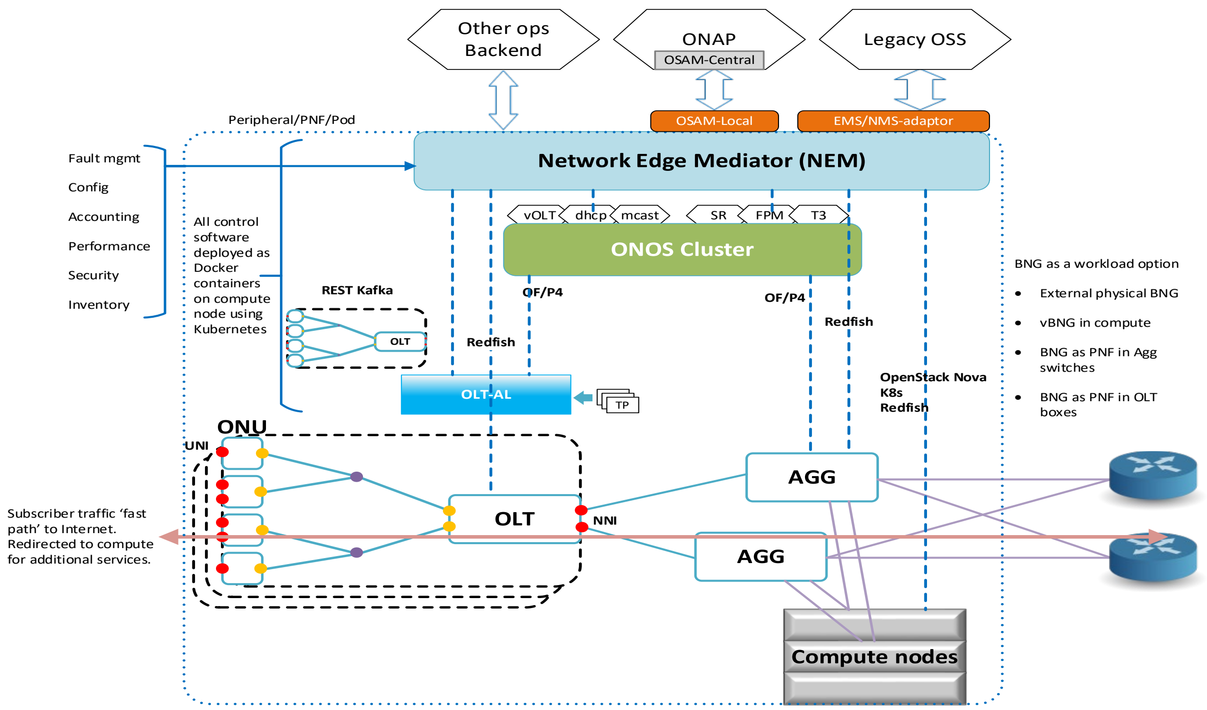

Fig. 3. The comprehensive framework for virtualization of the Central Office for GPON networks

Amongst others, our major contribution here, is to propose and implement a blue marked functional block called **OLT-AL** (OLT Abstraction Layer) which helps to hide all the internal details of OLT & ONU and reveal it to ONOS as quasi-Ethernet switches with UNI and NNI ports. That makes ONOS be able to control any type of OLT & ONU white-box hardware from any vendors and totally solve the inter-operability problem in access network environment.

When talking about virtualization of OLT, we need to understand the concept of Dynamic Bandwidth Allocation (DBA) algorithms, which allows the OLT to orchestrate the access to the shared medium and to adapt dynamically to the changing requests from the ONUs. In the GPON architecture, an entity called Multi Point Control Protocol (MPCP) manages the upstream channel and harmonizes the transmission of data from ONUs to OLTs. In addition, it manages operations such as terminal discovery, registration, and bandwidth allocation. MPCP uses two standard messages for DBA operations, namely GATE and REPORT. REPORTs are used by ONUs to request transmission opportunities, and GATEs are sent by the OLT to grant a transmission slot (time and length) to a specific ONU.

That said, and basically, our proposed solution for OLT virtualization will divide the vendor-locked OLT box into three layers: (i) layer 1 is only HW boxes (bare metal or white boxes) consisting of backplane, the control and interface cards, (ii) the hardware abstraction layer (OLT-AL) which provides the communication stacks for controller to manage and control the OLT HW boxes, and finally (iii) the SDN Controller which provides major functions of an OLT to its applications such as VLAN, AAA or multicast. The control interface we use on this architecture is Openflow and P4 and for the SDN controller we use ONOS from ONF (Open Network Foundation) and thus our contribution here is to design (Fig. 6) and implement the OLT-AL.

Again, the OLT-AL provides an abstraction for the PON by modeling it as an OpenFlow switch to the SDN controller, with UNI (User to Network Interface) and NNI (Network to Network Interface) ports, hiding internal details about the OLT and ONUs. Its working principles are as illustrated in Fig. 6 and 7 below. More specifically, as we can see from Fig. 6, each time Affinity router receives an Openflow message from ONOS or a command from CLI, it sends the message to two cores, API Handler within the cores will receive the message and decide whether to execute it or keep it standby. Depending on the request, it can be executed by either Device Agent or Logical device agent (flow decomposer). And we can see, from the bottom of Fig. 6, that series of OLT and ONU adaptors make it works with as many OLT HW vendors as possible.

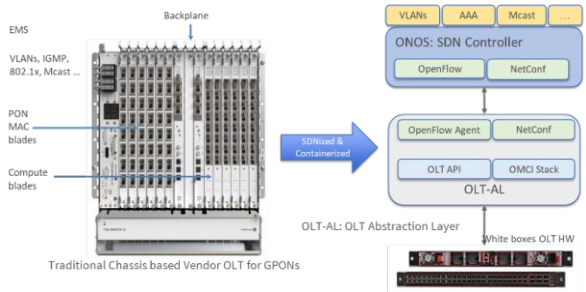

Fig. 5. The proposition of SDN based GPON OLT

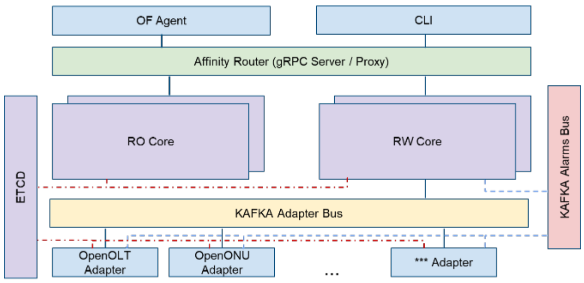

Fig. 6. OLT-AL architecture

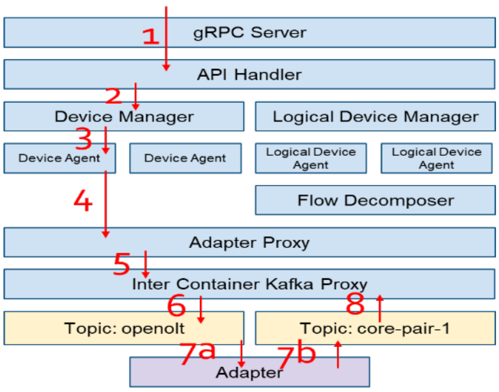

Fig. 7. Request execution flow inside the RW core

The detailed request execution flow's illustrated in Fig. 7 and with steps of (1) when the operator enables a device the request is passed to the API Handler; (2) API Handler creates a new thread to manage the request and invokes Device Manager; (3) DM loads the Device Agent for the correct device and invokes the correct method; (4) DA calls Adapter Proxy; (5) AP prepares the data to be sent over Kafka bus and define the topics to send the message and receive the response; (6) Inter Container Proxy sends the message and starts listing on the response topic; (7) Adapter processes the request and respond, finally, (8) Inter Container Proxy receives the response and returns the value.

For the containerization of control processes (of OLT) we use Kafka architecture for inter-container communication in support of Kafka bus and adapters.

3.2. Integration Works

Within the scope of the project and in collaboration with VNPT R&D Center, we have developed an integrated test-bed to test the proposed design with configuration as depicted in Fig. 8. Components on this configuration are inclusive of RG (Linux box); ONU (Alpha or any other BCM based ONU); OLT (ASFVOLT16); BNG (Linux box with a 10G NIC); Dev - Linux (Ubuntu) box used for development and running OLT-AL+ONOS.

The main job here is to develop OLT-AL module and integrated it with ONOS (SDN Controller) in order to control the OLT white-box. The integrated system then can provide basic GPON network services, such as ONU registration, subscriber authentication, DHCP, connectivity and monitoring. Examples of call flow details of ONU registration and subscriber authentication are depicted in Fig. 9 and 10.

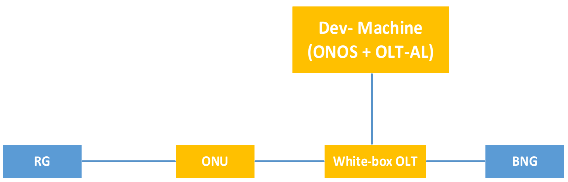

Fig. 8. The integration of in house developed components for testing virtualized GPON services

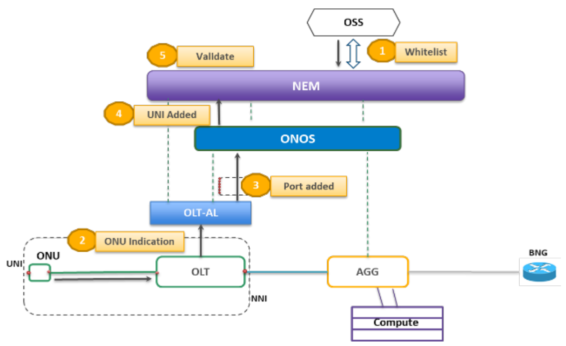

Fig. 9. ONU registration procedures

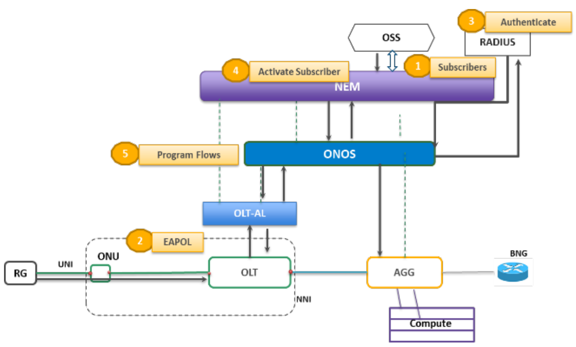

Fig. 10. 802.1x authentication procedures

All software and hardware components are pre-installed as a container Pod, this includes internal physical connectivity and internal private addressing between software and hardware components. As a result, one Pod internally includes the following components:

- NEM, OLT-AL and ONOS services deployed using Kubernetes (k8s) with internal (local) docker registry
- 1 physical AGG switch
- Up to 6 Edge/Core OLTs with a single NNI port connected to the AGG switch and up to 26 ONTs on each PON port
- 3 compute nodes connected to AGG switch
- 1 GE management switch to which all OLTs, AGG and compute node management ports are connected. This management switch is ONLY for internal pod connectivity; it has no connectivity outside of the pod
- Pod will be racked up and connected to external BNG by VNPT R&D.

After the integration, the components will be initiated and running. For example, Fig. 11 and 12 below illustrate the OLT and ONT bring-up procedures.

The steps to bring the OLT up are following:

- Operator creates an Abstract OLT record in NEM. In the Abstract OLT it includes the Slot information for the actual physical OLT.
- NEM's vOLT service pre-provisions the OLT - makes a 'preprovision-olt' call to OLT-AL.
- NEM's vOLT service brings up the OLT - makes an 'enable' call to OLT-AL.
- NEM's vOLT service queries OLT-AL after OLT enable to get OLT information and PON ports to build its inventory.

Similarly, steps to bring the ONT up, are as follow:

- The customer connects and boot up their ONT. The ONT will be detected on the PON by the OLT and be brought up automatically in a bottom-up discovery model
- ONOS sends an event to NEM noting that a new UNI port with a particular ONT serial number has been added. NEM will determine whether the ONT is valid by consulting local pre-provisioned data (i.e. a 'whitelist')
- A whitelist includes information that the OSS has already pre-provisioned for an ONT serial number and has allocated a name and OLT slot/pon-port ('location') according to the Abstract OLT model

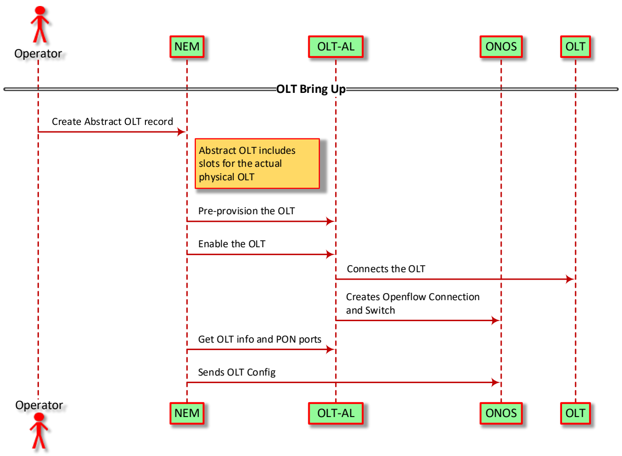

Fig. 11. OLT bring-up work flow

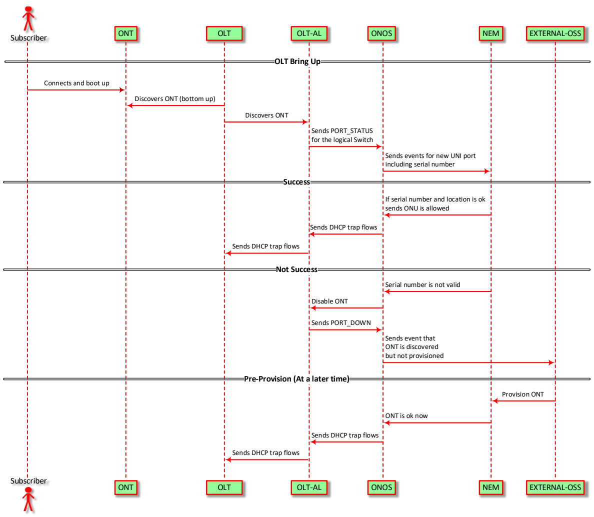

Fig. 12. ONT bring-up work flow

Other procedures like Subscriber Add, DHCP L2 Relay, etc. are also designed, implemented and initiated successfully in our federated test-bed (the test-bed has been connected with VNPT R&D facility) and, as a results, we can test most of GPON services as it has been done in traditional network. Some of the testing results will be show on the next section.

4. Testing Results and Performance Evaluation

We have setup the configuration for components of the test-bed as per Fig. 8 above with following steps:

Step 1: Component Setup

RG Setup: Install WPA supplicant (normally pre-installed with Ubuntu); *BNG Setup:* Configure double-tagged vlan interface; *Platform Setup:* Install Single-Node Kubernetes and Helm; *Helm charts:* Get the helm charts on dev machine; *OLT-AL Installation:* Running a pre-built OLT-AL Image (images are downloaded from docker-hub); *Install ONOS*; *ONOS apps*.

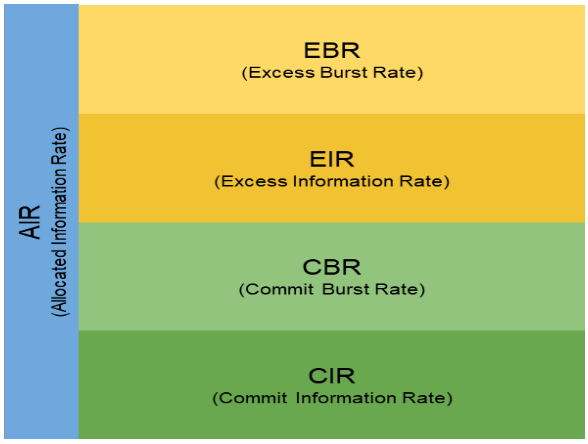

Fig. 13. OLT-AL Bandwidth Profile

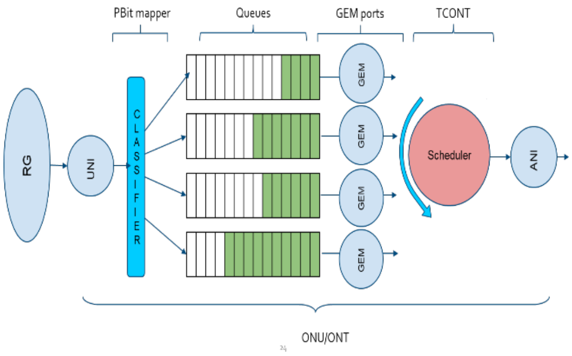

Fig. 14. OLT-AL Technology Profile

Step 2: Then we have to configure the OLT-AL Bandwidth Profile and Technology Profile.

Bandwidth Profiles (Fig. 13) specify how much bandwidth a subscriber is allocated, both upstream and downstream.

Technology Profiles (Fig. 14) allow the operator to define queuing strategies based on the type of traffic, often referred to as "Service Type" (eg: Business, Residential)

Step 3: Component Initiation

All components of RG, ONOS, OLT-AL helms, BNG are then initiated.

Step 4: Traffic Testing

Next step we conduct the ping test from RG to BNG successfully. And more importantly, we are able to verify from the traffic meters with traffic flow volume, uplink (as depicted on Fig. 15 and 16) and downlink (Fig. 17 and 18) below. When we test at speed of 10Mbps for uplink and 15 Mbps for downlink, the packet drop may have happened.

```
Tx rate - 5Mbps (Upstream Traffic)
[ ID] Interval      Transfer    Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0- 1.0 sec  593 KBytes  4.86 Mbits/sec  0.495 ms    0/  413 (0%)
[  3]  1.0- 2.0 sec  594 KBytes  4.87 Mbits/sec  0.460 ms    0/  414 (0%)
[  3]  2.0- 3.0 sec  594 KBytes  4.87 Mbits/sec  0.483 ms    0/  414 (0%)
[  3]  3.0- 4.0 sec  594 KBytes  4.87 Mbits/sec  0.469 ms    0/  414 (0%)
[  3]  4.0- 5.0 sec  593 KBytes  4.86 Mbits/sec  0.472 ms    0/  413 (0%)
[  3]  5.0- 6.0 sec  594 KBytes  4.87 Mbits/sec  0.485 ms    0/  414 (0%)
[  3]  6.0- 7.0 sec  594 KBytes  4.87 Mbits/sec  0.467 ms    0/  414 (0%)
[  3]  7.0- 8.0 sec  594 KBytes  4.87 Mbits/sec  0.481 ms    0/  414 (0%)
[  3]  8.0- 9.0 sec  593 KBytes  4.86 Mbits/sec  0.480 ms    0/  413 (0%)
[  3]  9.0-10.0 sec  594 KBytes  4.87 Mbits/sec  0.469 ms    0/  414 (0%)
[  3]  0.0-10.3 sec  5.96 MBytes  4.87 Mbits/sec  0.482 ms    0/ 4252 (0%)
```

Fig. 15. Traffic Parameters: Traffic type – UDP / Packet size – 1470bytes / Test Duration – 10sec / Direction - RG->BNG

```
Tx rate - 10Mbps
[ ID] Interval      Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0- 1.0 sec  1.17 MBytes  9.85 Mbits/sec  0.366 ms    0/  838 (0%)
[  3]  1.0- 2.0 sec  1.16 MBytes  9.71 Mbits/sec  1.189 ms    0/  826 (0%)
[  3]  2.0- 3.0 sec  1.16 MBytes  9.71 Mbits/sec  1.230 ms    0/  826 (0%)
[  3]  3.0- 4.0 sec  1.16 MBytes  9.71 Mbits/sec  1.193 ms    0/  826 (0%)
[  3]  4.0- 5.0 sec  1.16 MBytes  9.74 Mbits/sec  1.213 ms    0/  828 (0%)
[  3]  5.0- 6.0 sec  1.16 MBytes  9.71 Mbits/sec  1.220 ms    0/  826 (0%)
[  3]  6.0- 7.0 sec  1.16 MBytes  9.71 Mbits/sec  1.177 ms    0/  826 (0%)
[  3]  7.0- 8.0 sec  1.16 MBytes  9.71 Mbits/sec  1.221 ms    0/  826 (0%)
[  3]  8.0- 9.0 sec  1.16 MBytes  9.74 Mbits/sec  1.195 ms    0/  828 (0%)
[  3]  9.0-10.0 sec  1.16 MBytes  9.71 Mbits/sec  1.202 ms    0/  826 (0%)
[  3]  0.0-10.3 sec  11.9 MBytes  9.73 Mbits/sec  1.216 ms    0/ 8504 (0%)
```

Fig. 16. Traffic Parameters: Traffic type – UDP / Packet size – 1470bytes / Test Duration – 10sec / Direction – RG->BNG

```
Tx rate - 10Mbps
[ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0- 1.0 sec  1.17 MBytes  9.85 Mbits/sec  0.366 ms    0/  838 (0%)
[  3]  1.0- 2.0 sec  1.16 MBytes  9.71 Mbits/sec  1.189 ms    0/  826 (0%)
[  3]  2.0- 3.0 sec  1.16 MBytes  9.71 Mbits/sec  1.230 ms    0/  826 (0%)
[  3]  3.0- 4.0 sec  1.16 MBytes  9.71 Mbits/sec  1.193 ms    0/  826 (0%)
[  3]  4.0- 5.0 sec  1.16 MBytes  9.74 Mbits/sec  1.213 ms    0/  828 (0%)
[  3]  5.0- 6.0 sec  1.16 MBytes  9.71 Mbits/sec  1.220 ms    0/  826 (0%)
[  3]  6.0- 7.0 sec  1.16 MBytes  9.71 Mbits/sec  1.177 ms    0/  826 (0%)
[  3]  7.0- 8.0 sec  1.16 MBytes  9.71 Mbits/sec  1.221 ms    0/  826 (0%)
[  3]  8.0- 9.0 sec  1.16 MBytes  9.74 Mbits/sec  1.195 ms    0/  828 (0%)
[  3]  9.0-10.0 sec  1.16 MBytes  9.71 Mbits/sec  1.202 ms    0/  826 (0%)
[  3]  0.0-10.3 sec  11.9 MBytes  9.73 Mbits/sec  1.216 ms    0/ 8504 (0%)
```

Fig. 17. Traffic Parameters - Traffic type – UDP / Packet size – 1470bytes / Test Duration – 10sec / Direction - BNG->RG

```
Tx rate - 15Mbps
[ID] Interval      Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[4]  0.0- 1.0 sec  1.18 MBytes  9.89 Mbits/sec  1.202 ms    0/  841 (0%)
[4]  1.0- 2.0 sec  1.16 MBytes  9.71 Mbits/sec  1.204 ms    0/  826 (0%)
[4]  2.0- 3.0 sec  1.16 MBytes  9.74 Mbits/sec  0.762 ms  290/ 1118 (26%)
[4]  3.0- 4.0 sec  1.16 MBytes  9.71 Mbits/sec  0.780 ms  449/ 1275 (35%)
[4]  4.0- 5.0 sec  1.16 MBytes  9.71 Mbits/sec  0.791 ms  448/ 1274 (35%)
[4]  5.0- 6.0 sec  1.16 MBytes  9.71 Mbits/sec  0.790 ms  449/ 1275 (35%)
[4]  6.0- 7.0 sec  1.16 MBytes  9.71 Mbits/sec  0.784 ms  449/ 1275 (35%)
[4]  7.0- 8.0 sec  1.16 MBytes  9.74 Mbits/sec  0.793 ms  449/ 1277 (35%)
[4]  8.0- 9.0 sec  1.16 MBytes  9.71 Mbits/sec  0.784 ms  450/ 1276 (35%)
[4]  9.0-10.0 sec  1.16 MBytes  9.71 Mbits/sec  0.762 ms  449/ 1275 (35%)
[4]  0.0-10.8 sec  12.6 MBytes  9.74 Mbits/sec  0.810 ms 3799/12755 (30%)
```

Fig. 18. Traffic Parameters - Traffic type – UDP / Packet size – 1470bytes / Test Duration – 10sec / Direction - BNG->RG

5. Conclusion

We have described a novel architectural design for SDN-controlled GPON networks, together with our implementation details. Our architecture minimizes the management and operational complexity of the GPON, while optimizing the flexibility and controllability of the network. Though, the work here focuses more on the virtualization of OLT, our design has proposed a comprehensive framework for virtualization of entire optical access network. The SDN based virtualization covers all components from OLT, Aggregation Switches and Broadband Gateway (BNG or BRAS). Our implementation can help also to pave an initial way for packaging all of those virtualized software and hardware components on one POD that makes very easy and quickly deployment for network operators. Several test-bed tests have demonstrated a significant improvement in data packet delay and downstream and upstream throughput when compared with the legacy GPON architecture.

Our work is now to move to the phase of development for power-saving algorithms in the OLT and the ONUs, and its management by the SDN control plane. Power saving can be obtained by setting the transmitters and the receivers of both the OLT and ONUs to a sleep state when there is no traffic. The SDN controller could modify the behaviour of the DBA and use the GATE/REPORT messages to switch the state of the ONU between sleep, doze, and active mode. That will make SDN-based GPON virtualization even more senses.

References

[1] Jane M Simmons. Optical Network Design and Planning, Springer; 2014. https://doi.org/10.1007/978-3-319-05227-4

[2] Glen Kramer, Gerry Pesavento, Ethernet passive optical network (EPON): building a next generation optical access network, IEEE Communications Magazine, 2002; 40:66-73. https://doi.org/10.1109/35.983910

[3] Ivica Cale, Aida Salihovic, and Matija Ivekovic. Gigabit passive optical network-GPON. In: 29th International Conference on Information Technology Interfaces; 25-28 June; Cavtat, Croatia. IEEE; 2007. pp. 679-684.

https://doi.org/10.1109/ITI.2007.4283853

[4] Ruffini, M.; Payne, D.B. Business and ownership model case studies for next generation FTTH deployment, 2016. (accessed on 3 October 2019). Appl. Sci. 2019, 9, 4566 24 of 28, Available online: https://mruffini.files.wordpress.com/2016/02/business_mod_case_studies_final.pdf

[5] Payne, D.B.; Ruffini, M. Local Loop Unbundling Regulation: Is It a Barrier to FTTH Deployment?, 2016. (accessed on 3 October 2019), Available online: https://mruffini.files.wordpress.com/2016/02/lluwhite-paper-final.pdf

[6] Peterson, L.; Al-Shabibi, A.; Anshutz, T.; Baker, S.; Bavier, A.; Das, S.; Hart, J.; Palukar, G.; Snow, W. Central office re-architected as a data center. IEEE Commun. Mag. 2016, 54, 96–101 https://doi.org/10.1109/MCOM.2016.7588276

[7] Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-defined networking: A comprehensive survey. Proceedings of the IEEE. 2015; 103(1):14-76. https://doi.org/10.1109/JPROC.2014.2371999

[8] Nick McKeown, Tom Anderson, Hari Balakrishnan MIT, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review. 2008; 38(2):69-74. https://doi.org/10.1145/1355734.1355746

[9] 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. https://doi.org/10.1109/MCOM.2014.6894466

[10] Pawel Parol, and Michal Pawlowski. Towards networks of the future: SDN paradigm introduction to PON networking for business applications. In: Computer Science and Information Systems (FedCSIS), Federated Conference on; 8-11 Sept. 2013; Krakow, Poland. IEEE; 2013.

[11] M.Ruffin, F.Slyne, C.Bluemm, N.Kitsuwan, S.McGettrick, Software defined networking for next generation converged metro-access networks, Optical Fiber Technolog. 2015; 26:31-41. https://doi.org/10.1016/j.yofte.2015.08.008

[12] Hui Yang, Jie Zhang, Yongli Zhao, Jialin Wu, Yuefeng Ji, Yi Lin, Jianrui Han, Young Lee, Experimental demonstration of remote unified control for OpenFlow-based software-defined optical access networks, Photonic Network Communications, 2016; 31(3):568-577. https://doi.org/10.1007/s11107-015-0547-6

[13] Steven S. W. Lee, Kuang-Yi Li, and Ming-Shu Wu. Design and implementation of a GPON-based virtual openflow-enabled SDN switch. Journal of Lightwave Technology. 2016; 34(10):2552-2561. https://doi.org/10.1109/JLT.2016.2540244