Experimental demonstration of virtual OLT-based PON slicing for co-existence of different services

ChanSung Park
Network Reaserch Division
Electronics and Telecommunications Research Insitute (ETRI)
Deajeon, South Korea
chansung18@etri.re.kr

YongWook Ra
Network Research Division
Electronics and Telecommunications Research Institute (ETRI)
Daejeon, South Korea
nyw0242@etri.re.kr

HwanSeok Chung
Network Research Division
Electronics and Telecommunications Research Institute (ETRI)
Daejeon, South Korea
chung@etri.re.kr

***Abstract*— We experimentally demonstrate PON slicing based on virtual PON and 25Gbps based white-box type OLT, which provides logical separation of physical resources, ensuring the connectivity, latency, and throughput.**

Keywords*— *PON Slicing, Virtual PON, SDN controller

I. Introduction

In recent years, as 5G communication has become popular, there are a lot of new services such as smart factory, personal broadcasting, immersive media, and the combination of large-capacity big data and artificial intelligence. These demands for new applications with diverse requirements are driving Time-division multiplexing-passive optical network (TDM-PON) technology to slice services based on multiple logical networks. Especially, flexibility and agility play an important role so that TDM-PON can provide an infrastructure optimized for each service.

To satisfy these diverse requirements, there are four technologies required in optical access networks: software-defined network (SDN), network function virtualization (NFV), hardware (HW) abstraction, and TDM-PON slicing. SDN centralizes the control plane to provide control and management of network elements, while NFV enables the use of virtualized environments to provide application agility. The conceptual combination of SDN and NFV enables agile, flexible, and cost-effective network construction. In addition, HW abstraction of optical line terminal (OLT) and all connected optical network units (ONUs) to the SDN controller allows to overcome vendor-locked limitations. TDM-PON slicing is a key enabler that provides each service with a tailored set of TDM-PON resources according to the characteristics of flowing traffic via dedicated slices to meet service-specific requirements such as bandwidth and low latency. As a result of the above-mentioned technologies, TDM-PON infrastructures transform into flexible and agile platforms.

To gain benefits of cloudification and virtualization to broadband access networks, SDN enabled broadband access (SEBA) is the most successful study of the core principles of SDN and disaggregation OLT [1]. However, SEBA have mostly focused on FTTH/FTTB deployment. There have also been substantial efforts to achieve convergence of time-critical and non-time-critical applications in shared optical access networks. However, the previous studies mostly focused on channel bonding and low-latency oriented dynamic bandwidth allocation (DBA) [2-3], network slicing based on emulators using Co-DBA [4] in PON. There have been no experimental demonstrations involving all four technologies (i.e., SDN, NFV, HW abstraction, TDM-PON slicing) to support mobile services as well as conventional broadband services. The focus of this paper is the control and management aspects of access network slicing realized by disaggregated TDM-PON.

In this paper, we demonstrate PON slicing with a 25Gbps white-box type physical PON and SEBA based vPON for the co-existence of different services. In addition, we show that control and management aspects of PON slicing. The results of monitoring performance metrics such as statistics and latency of a slice for mobile service are also presented.

II. PON Architecture

A. Virtual PON

TDM-PON system consists of two blocks, namely virtual PON (vPON) and physical PON (pPON). vPON is a modified SEBA-based software platform and controls pPON for virtualization, hardware abstraction, and PON slicing. Therefore, vPON includes open network operating system (ONOS) for SDN controller, slicing applications, hardware abstraction (VOLTHA) with OLT/ONU adapters. pPON also includes physical OLT (pOLT) and physical ONU (pONU).

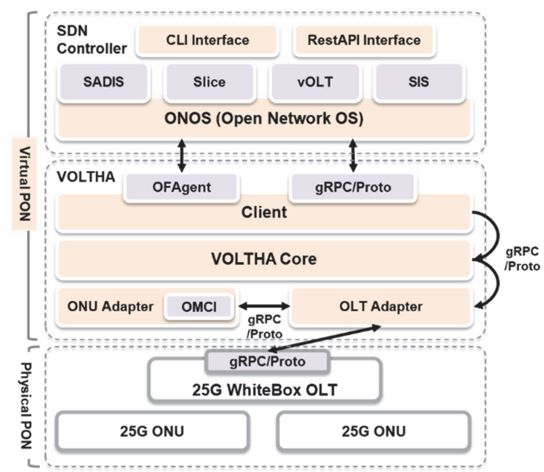

Fig. 1. High-level PON Architecture of Virtual PON and Physical PON.

Fig. 1 shows the overall architecture of physical as well as virtual PON. In vPON, ONOS consists of various applications for network functions and managements, and it also comes with profile databases of slice instances and service flows. By the interworking from applications to applications and databases, logical isolations of the underlying network resources are determined, and the decisions for slicing are sent to VOLTHA via gRPC/protobuf and OpenFlow (OF) messages. Specifically, Slice application provisions slice instances to VOLTHA after looking up the corresponding slice profiles through SIS (Slice Information Service) application. Also, when network operators try to install service flows, Slice application dynamically checks if the flows can be accommodated within any existing slice instances based on the service requirements by referring SADIS (Subscriber Access Device Information) application. Then, if allowed, vOLT application organizes OF message and send it to VOLTHA to establish the flow in a given slice instance. VOLTHA decomposes and dispatches the any messages to OLT/ONU adapters. OLT adapter abstracts the underlying different kinds of pOLTs. After ONU discovery process is complete, the ONU adapter builds and sends ONU management and control interface (OMCI) messages to establish connections to pONUs by running the OMCI state machine.

B. *Applications*

To realize PON slicing, there are four applications on ONOS: SADIS, SIS, Slice, and vOLT. SADIS and SIS are the caching applications as well as connection points to the underlying databases outside ONOS. SIS manages the information of slice profiles such as slice ID, service type (i.e., Residential, Business, Mobile), T-Conts, DBA algorithm, and QoS, while SADIS manages subscriber-related information by referring to the slice ID to which service flow should belong from SIS. Slice application is responsible for provisioning slice instances and checking if service flows can be accommodated in certain slice instances. That is, Slice application confirms that if there is enough remaining bandwidth to accept the requests of service flow install in a designated slice instance which logically takes a part of a PON port. The actual installation of service flows is handled by vOLT, but the installation sequence is triggered by Slice application because the bandwidth sanity check should be done ahead of time. If it passes the sanity check, vOLT application builds an OF message with flow ID, up /downstream QoS, technology profile ID, and VLAN tags, then transmit it to VOLTHA through OF channel.

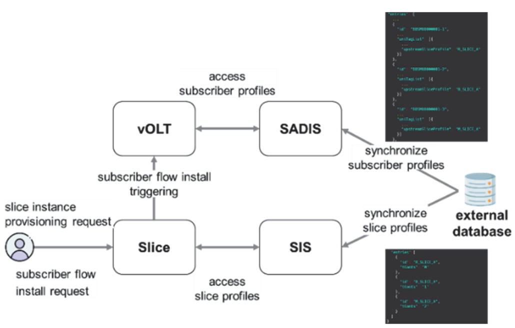

Fig. 2. Simplified schema of the slice workflow.

Fig. 2 shows such relationship between the four applications on ONOS. As you can see, Slice application is the central place to establish slice instances and install service flows by collaborating with vOLT, SADIS, and SIS applications. Note that all the hardware properties are notified from VOLTHA to Slice application via gRPC/protobuf streaming channel so that Slice application knows the capacity of the entire PON channel.

```
Procedure: slice instance provisioning
Inputs:
    slice_req, sis
Initialize:
    slice_instance
slice_profile = sis.get(slice_req.slice_id)
IF slice_profile EXIST
    remain_bw = slice_req.pon_port.remained_bw
    IF remain_bw >= slice_req.bw
        t_conts = slice_profile.alloc_tconts
        SEND req to voltha with t_conts
        UPDATE remain_bw −= slice_req.bw
        UPDATE slice_instance with slice_req
        ADD slice_instance to slice_manager
    ENDIF
ENDIF

Procedure: subscriber's service flow install
Inputs:
    sub_req, sadis, slice_app
sub_profile = sadis.get(sub_req.sub_id)
IF sub_profile EXIST
    slice_instance ← slice_app.get(sub_profile.slice_id)
    remain_bw ← slice_instance.remain_bw
    req_bw ← sub_profile.bw
    IF remain_bw >= req_bw
        SEND sub_req to volt application
        UPDATE remain_bw −= req_bw
    ENDIF
ENDIF
```

Fig. 3. Pseudo code of slice instance and service flow provisioning.

The pseudocode in Fig. 3 shows two procedures of the slice instance provisioning and service flow installation. Slice instance manager must manage the capacity of all PON ports because PON slicing isolates a certain amount of bandwidth in a given PON port. Also, it is important to note that the list of T-Conts should be determined and stored in the slice profile database before the provisioning time according to the network policy while other attributes such as bandwidth and DBA can be set dynamically at runtime. Once slice instances are provisioned, service flows can be installed on a designated slice instance. To this end, each service profile refers which slice instance it belongs to. In other words, there should be a slice instance with the same ID that a service profile can look up to. By doing this, Slice application can check if the requested bandwidth profile in a service flow can be accommodated by a designated slice instance.

III. Experimental Setup and Results

A. *Experimental Setup*

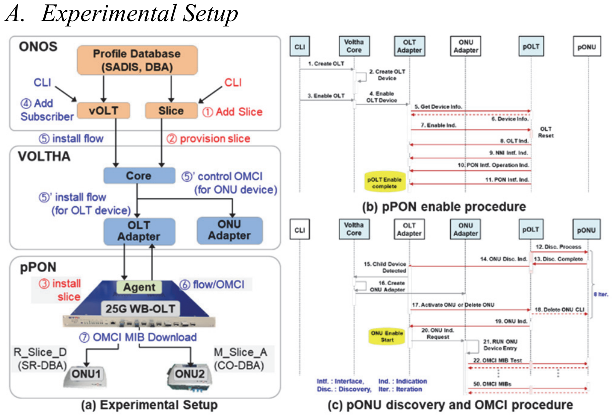

Fig. 4. (a) Experimental setup of PON slicing, (b) PON enable and (c) OMCI Procedure.

Fig. 4(a) shows an experimental setup of PON slicing with 1 x 25G WB OLT and 2 x 25G ONUs. In the first phase with procedures from ① to ③, we instruct provisioning requests of two slice instances (i.e., residential and mobile services) composed of SR-DBA and Co-DBA, respectively, through CLIs. Note that in procedure ③ of the first phase, the pPON enable procedure shown in Fig. 4(b) is completed. In the second phase with procedures from ④ to ⑦, we instruct the install requests of the two flows for each slice instances. Then, VOLTHA decomposes the OF message into two parts for flow installation and OMCI control. As shown in Fig. 4(c), when pONUs are discovered, the procedure ⑦ of downloading OMCI MIBs and OMCI procedure begin.

B. Results

Fig. 5 shows the two slice instances and the two flows are correctly provisioned and installed. In Fig. 5, we confirmed the PON slicing configuration with a custom designed GUI program and ONOS CLI. The topologies in Figs. 5(a) and 5(b) show that two slice instances, i.e., R_SLICE_D and M_SLICE_A, are well provisioned from pOLT1 to pONU1 and pONU2 respectively. Fig. 5(c) also shows the results of slice instance attribute information through CLIs.

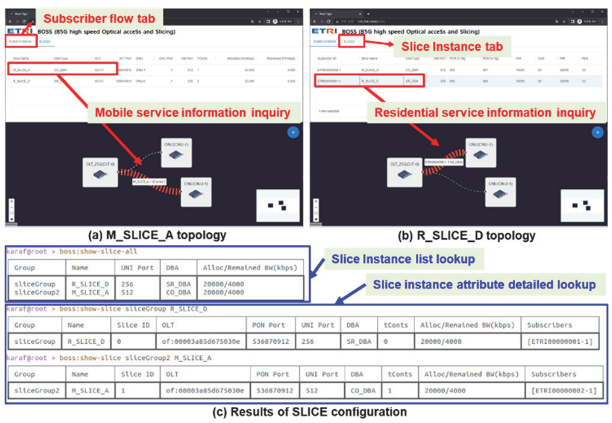

Fig. 5. Topology views and Results of PON Slicing.

It is important note that each slice instances share the same PON ports while their UNI ports differ, and different sets of T-Cont are assigned. That is because the combinatorial information of PON port, UNI port, and T-Conts distinguishes different slice instances so that it identifies which service flow should belong to which slice instance.

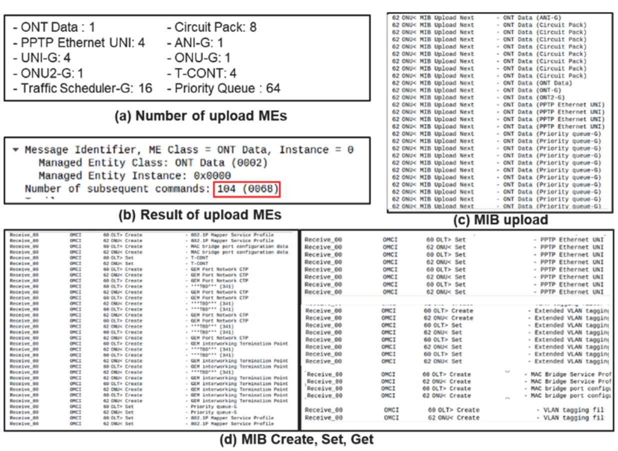

Fig. 6. Experimental results of OMCI procedure based on 25G PON.

Fig. 6 shows that two service flows are correctly established by snooping OMCI packets with Wireshark. This result confirms that the OMCI state machine of ONU adapter works correctly, and its decisions are sent to the agent of 25G WB OLT through OLT adapter's gRPC/protobuf channel.

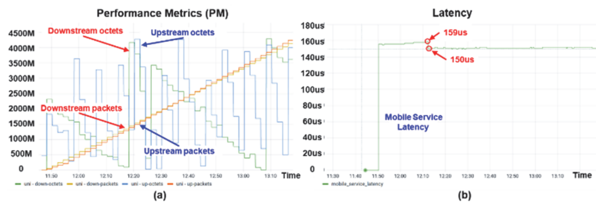

Fig. 7. (a) Performance metric (PM) and (b) latency measured at the flow on mobile slice instance.

Figs. 7 (a) and (b) show the measured results of the performance metrics (PM) and latency of the mobile service flow collected via Port Statistics indications and OMCI indications, respectively. The collected data is reorganized and exported by JSON exporter, and displayed with Grafana dashboard. The PM and latency are collected in every three minutes and five seconds respectively. The PMs shows the accumulated number of packets and size of octets over time as shown in Fig. 7(a). Also, the latency confirms that CO-DBA is correctly applied, and the stable latency between 150 and 159 μs is guaranteed which is suitable for mobile service in Fig. 7(b).

All these results indicate that vPON-based PON slicing provides a way to flexibly separate a single physical PON into multiple logical PONs to accommodate various services depending of their requirements such as latency and bandwidth in isolated manner.

IV. Conclusions

We presented vPON based PON slicing and control and management aspects for the co-existence of different services with various demands by vPON control. Higher priority slice, such as mobile service, was logically isolated from other slices, and the performance was not affected by heavy traffic from lower priority services. We confirmed that slice instances and service flows were flexibly configured by collecting performance metrics. As a result, the measured latency was maintained less than 159 μs to meet the requirements of mobile service.

Acknowledgment

This work was supported by IITP grant funded by the Korea government (MSIT) [2019-0-00452, B5G high speed optical access and slicing technology]

References

[1] S. Das, "From CORD to SDN Enabled Broadband Access (SEBA) [Invited Tutorial]," Journal of Optical Communications and Networking, vol. 13, no. 1, pp. A88-A99, January 2021, DOI: 10.1364/JOCN.402153.

[2] KwangOk Kim, et al., "High Speed and Low Latency Passive Optical Network for 5G Wireless Systems," IEEE/OSA Journal of Lightwave Technology, vol. 37, no. 12, pp. 2873-2882, June 15, 2019.

[3] HwanSok Chung, et al., "TDM-PON based optical access network for tactile internet, 5G, and beyond," IEEE Network, vol. 36, no. 2, March/April, 2022.

[4] H. Uzawa, K. Honda, H. Nakamura, Y. Hirano, K. Nakura, S. Kozaki, J. Terada, "Dynamic bandwidth allocation scheme for network-slicing-based TDM-PON toward the beyond-5G era", IEEE/OSA Journal of Optical Communications and Networking, vol. 12, no. 2, pp. A135-A143, 2020, DOI: 10.1364/JOCN.12.00A135.