Two-stage Abstraction for Disaggregated Modular OLT Architecture Supporting OpenFlow Control

Keita Nishimoto, Kota Asaka, Jun-ichi Kani, and Jun Terada
NTT Access Network Service Systems Laboratories, 1-1 Hikarinooka Yokosuka-shi, Kanagawa 239-0847, Japan
keita.nishimoto.wr@hco.ntt.co.jp

Abstract: We implement our abstraction method for provisioning and controlling, via OpenFlow, the disaggregated PON-OLT that features separation of hardware module and softwarized OLT functions, and demonstrate its operation by utilizing open source controllers ONOS / VOLTHA. © 2020 The Author(s).
OCIS codes: (060.4510) Optical communications, (060.4250) Networks

1. Introduction

To date, Fiber To The Home (FTTH) continues to be provided by the Passive Optical Network (PON). However, recent research is focusing on applying the PON to various fields (e.g. Passive Optical LAN [1] and Smart Wind farms [2]). The following two requirements should be satisfied to support various environments; (1) Space-saving and light weight architecture that enables small-start deployment, (2) Flexibility with regard to updating functions and adapting to various fields. To meet these requirements, we recently proposed the new PON system architecture called Mini-PON [3, 4]. The key feature of Mini-PON is that the PON-OLT is composed of softwarized and minimum OLT functions (OLT Software) and hardware module (H/W module) that is pluggable into Aggregation Service Gateway (ASG) like an OpenFlow switch (Fig. 1 (a)). OLT Software runs on a commodity server called OLT Compute in a central office (CO), while the H/W modules are distributed in various sites (e.g. factories, campus, etc.) distant from CO. This module minimizes the space needed for OLT implementation and drastically reduces the provisioning workload. Moreover, implementing OLT functions as software will enable the operators to easily update the functions to suit the latest environment, and also enable small start deployment as long as those functions are minimized. Regarding the module-type OLT architecture, Boyd et al. proposed a module-type OLT that is pluggable into an L2 switch [5]. Our architecture is different from theirs in the sense that we aim at enhancing flexibility by increasing the softwarized components, while they intended to implement most OLT functions inside the module except for a management function (i.e. partially disaggregated OLT architecture). Ruffini et al. demonstrated the vDBA concept that disaggregates the PON-DBA (Dynamic Bandwidth Allocation) function from OLT hardware into a general-purpose server [6]. However, they have not yet achieved the modularization of H/W and its abstraction and control; the main advances found in this paper.

As well as progress in module-type OLT devices and OLT function disaggregation, recent studies have targeted the control of optical access networks by the Software-Defined Network (SDN) protocol (e.g. OpenFlow) [7, 8]. The goal is to enable an OpenFlow controller to manage PON devices by abstracting PON-specific devices (i.e. OLT/ONUs) into a logical OpenFlow switch. OpenFlow is supported by open source communities, and many open source OpenFlow controllers have been developed. We can thus expect that a PON system that supports OpenFlow would make network management/control more cost-efficient, especially for the small scale start of a new service.

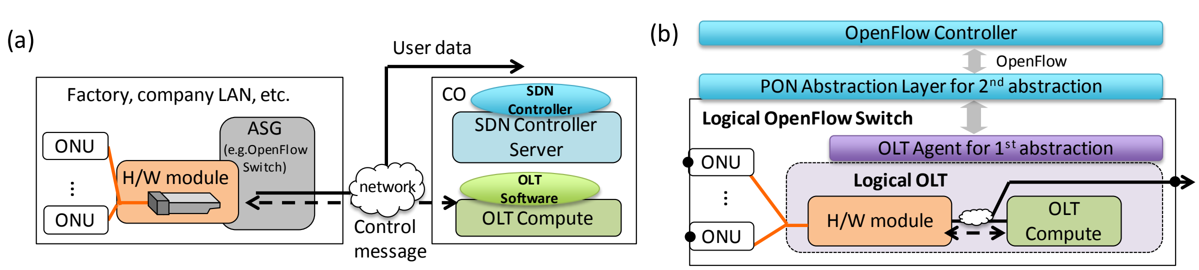

Fig. 1. (a) Mini-PON Architecture proposed in [3] and (b) our proposed "two-stage abstraction" method as Mini-PON C-plane.

We move these existing PON abstraction studies forward, and propose an OLT/ONU abstraction method aiming at OpenFlow control of disaggregated OLT architectures including our Mini-PON. We newly implement its control/management architecture by utilizing an open source OpenFlow controller and PON Abstraction Layer on our Mini-PON prototype that we developed in our previous studies mainly for measuring D-Plane and DBA performance [3, 4], and demonstrate that the OpenFlow controller can provision it. We believe this is the first study of an OpenFlow controller provisioning service on a disaggregated modular OLT device, and thus is an important milestone in expanding the application range of PON.

2. Mini-PON architecture

As we explained in the Introduction, Mini-PON features micro H/W modules and a disaggregated architecture. We implement most OLT functions as software running on a commodity server called OLT Compute; the functions include those whose update will be demanded by new service requirements. We implement the remaining functions in the module; these functions are difficult to implement as software, or not frequently updated. The module is pluggable into commodity L2/OpenFlow switch ports, and can be sited remotely from the CO. This disaggregation reduces the space needed to implement various services, and helps operators/service providers to change the functions flexibly. Specifically, the PON-PHY and GATE/REPORT message handling functions are implemented in the module while DBA and OAM (Operation Administration and Management) functions run on the server. Please note that DBA function and GATE/REPORT message processing are disaggregated and linked via a standard-based API (TR-402/403[9]) for enhanced function modularity.

3. Proposal

The main technical issue for PON control via OpenFlow is the abstraction of its tree topology. For example, Lee et al. proposed a method for abstracting an OLT and ONUs as a single logical switch that enabled OpenFlow control [7]. Furthermore, Open Networking Foundation [8] implemented a PON Abstraction Layer called VOLTHA based on the abstraction method proposed by Lee et al. [7]. VOLTHA is software that abstracts PON devices into a logical OpenFlow switch and exposes its interface to an OpenFlow controller above it. We go beyond these existing studies, which assumed a single chassis-type or box-type OLT device, by proposing "two-stage abstraction" method for disaggregation-type OLT architectures including Mini-PON, where OLT functions are physically separated between hardware and software. In our proposal, the H/W module and OLT Compute are abstracted as a single logical OLT device (1st abstraction), and the PON Abstraction Layer creates a logical OpenFlow device by abstracting the logical OLT device and ONUs (2nd abstraction) to allow access by an OpenFlow controller (Fig. 1(b)).

The following two features are required for implementing our "two-stage abstraction". The first one is path-control for the control messages exchanged between H/W module and OLT Compute which are physically separated. In our architecture, OpenFlow controller logically connects these separate components by assigning VLAN-IDs to each control message type, shown by colored arrows in Fig. 2 (a). The second feature is to let H/W module and OLT Compute cooperatively work and to abstract these components into a single logical OLT device after control message path establishment. We newly implement this as the feature called Mini-PON OLT Agent. Mini-PON OLT Agent consists of Manager, Module Adapter, and Compute Adapter. Manager mainly plays the role of issuing the commands (i.e. activation, deactivation, flow-entry install, flow-entry remove) sent from the PON Abstraction Layer to Module Adapter and Compute Adapter. Module Adapter translates the commands from Manager into H/W module-specific control messages, and sends each out to the module, while Compute Adapter gives commands to OLT Compute.

From the view point of the provisioning sequence, our architecture offers a three-phase workflow to operators as shown in Fig. 2 (a); (1) Path configuration in OpenFlow switch: OpenFlow controller configures data paths and control message paths in OpenFlow switch. (2) OLT/ONU activation: Operator sends an activation instruction to PON Abstraction Layer. Then, PON Abstraction Layer gives the appropriate command to Manager in Mini-PON OLT Agent. It initializes H/W module via Module Adapter, and starts up OLT Software process on OLT Compute after ensuring that the module is ready. Manager checks the activation states of OLT/ONUs and further notifies PON Abstraction Layer of their completion. PON Abstraction Layer starts to recognize each OLT/ONU after receiving this notification and then creates a logical OpenFlow switch for access by OpenFlow controller. (3) Path configuration in H/W module: OpenFlow controller installs flow entries into the logical OpenFlow switch created by PON Abstraction Layer. PON Abstraction Layer decomposes the flow entries into each OLT/ONU, and sends the OLT entries to Mini-PON OLT Agent. Agent extracts the path information (e.g. VLAN-ID translation configuration) from the entries and sets the configuration in the H/W module via Module Adapter.

4. Evaluation results

Using the components explained above, we newly implemented Mini-PON OLT Agent on our GE-PON-based Mini-PON prototype (i.e. H/W module, OLT Compute) [3, 4]. We adopted VOLTHA 1.5 as PON Abstraction Layer and ONOS 1.13.3 [10] as OpenFlow controller. Fig. 2 (b) shows the CLI outputs of VOLTHA (upper) and ONOS (lower) when we provisioned a single H/W module with 4 ONUs. The red line in the upper figure indicates a logical OLT device that is successfully abstracted from H/W module and OLT Software. Furthermore, you can find a logical OpenFlow switch in ONOS CLI (the lower figure), which is composed of OLT and ONUs. These figures demonstrate that the two-stage abstraction worked well and that provisioning succeeded.

Next, we show the feasibility of flow control by OpenFlow controller. We passed upstream traffic (128Byte length/100Mbps) across the network, see Fig. 2 (c). While the packets were flowing, we let ONOS remove the flow entry that added VLAN tag (VLAN-ID: 1001) to the upstream packets and installed a new flow entry that added VLAN tag (VLAN-ID: 1002) just after that removal. Fig. 2 (d) shows the time-sequence of packet traffic with each VLAN-ID. You can see that VLAN-ID was switched successfully despite the packet loss occurred caused by the time-gap between the flow install and remove. This proves that Mini-PON was successful in allowing the OpenFlow controller to conduct path control.

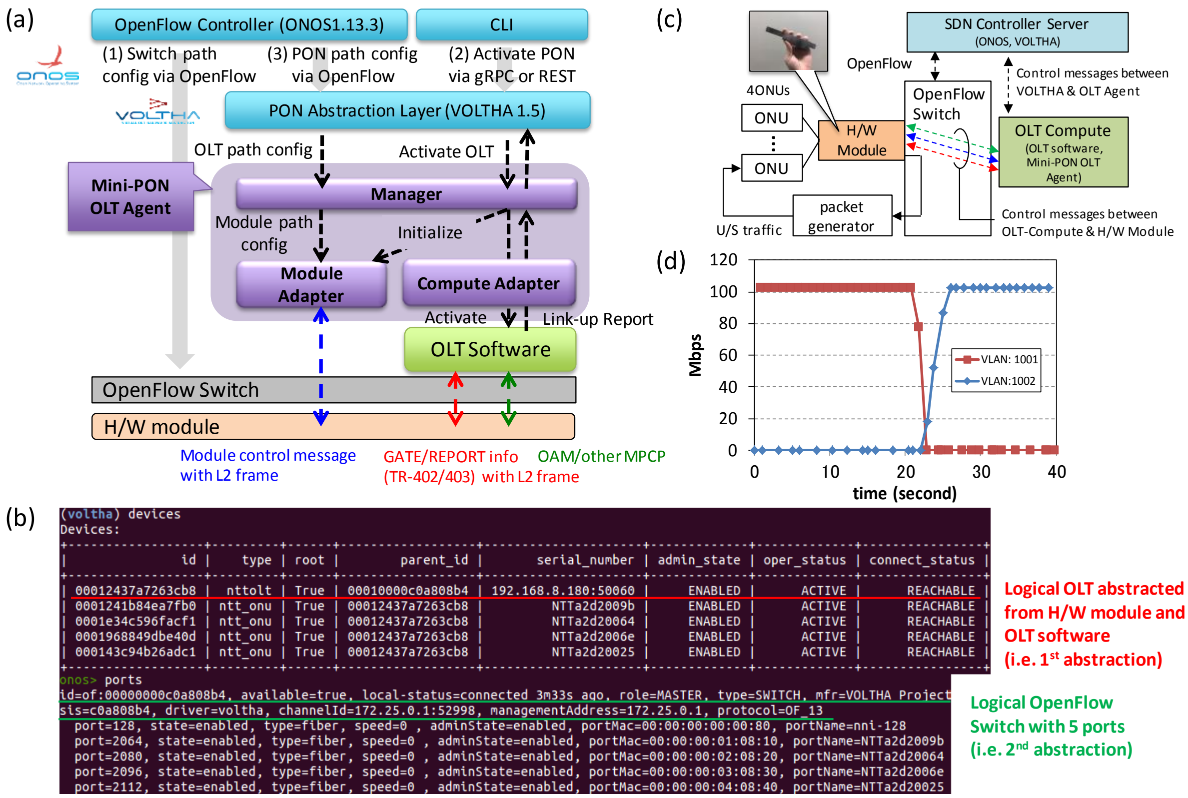

Fig. 2. (a) Implementation architecture of two-stage abstraction, (b) VOLTHA/ONOS outputs, (c) Experimental set-up, (d) Traffic changed by flow entry remove/install.

5. Conclusion

We demonstrated our proposed "two-stage abstraction" enabled open source controllers to provision and control a new PON architecture that features the separation of H/W module and softwarized OLT functions via OpenFlow.

6. Reference

[1] P. Parol, M. Pawloski, “Towards networks of the future: SDN paradigm introduction to PON networking for business applications,” Proc. 2013 Federated Conference on Computer Science and Information Systems, pp. 829–836 (2013).

[2] M. A. Ahmed, Y. C. Kim, “Communication Network Architectures for Smart-Wind Power Farm,” Energies, 7 (6), pp. 3900-3921 (2014).

[3] K. Nishimoto, et al., “Mini-PON: Softwarized module-type PON architecture enhanced by SEBA,” Demo in ONF Connect 2018 (2018).

[4] T. Tochino, et al., “First demonstration of Mini-PON by using standard-based DBA-API and software components on commodity server apart from modular OLT,” Proc. ECOC 2019, TuA3 (2019).

[5] E. Boyd, et al., “REMOTE PON NETWORK PERFORMANCE,” 2015 Spring Technical Forum (2015).

[6] M. Ruffini, F. Syne, “Moving the Network to the Cloud: The Cloud Central Office Revolution and Its Implications for the Optical Layer,” JLT, 37 (7), pp. 106-1716 (2019).

[7] S. W. Lee et al., “Design and Implementation of a GPON-Based Virtual OpenFlow-Enabled SDN Switch,” JOLT, 34 (10), pp. 2552-2561 (2016).

[8] E. Walter, “OFC 2018: AT&T's PON & Edge Compute Vision,” Proc. OFC 2018, pp. 1-21 (2018).

[9] TR-402/TR-403, “Functional Model for PON Abstraction Interface / PON Abstraction Interface for Time-Critical Applications,” (2018).

[10] P. Berde, P. et al., “ONOS: towards an open, distributed SDN OS,” Proc. HotSDN’14, pp. 1-6 (2014).