Implementation of software-based EPON-OLT and performance evaluation

Keita Nishimoto^{a)}, Masashi Tadokoro, Takeaki Mochida, Takashi Yamada, Toshikiyo Tanaka, Akiyuki Takeda, and Takashi Inoue

*NTT Access Network Service Systems Laboratories, NTT Corporation,
1-1 Hikarinooka, Yokosuka-shi, Kanagawa 239-0847, Japan*

a) nishimoto.keita@lab.ntt.co.jp

Abstract: In order to meet the diversifying demands, we propose the architecture of the software-based virtual EPON-OLT, where all of OLT functions except PHY-Layer are implemented as software running on an x86 server. The EPON-OLT requires not only high throughput, low latency but also accurate controls for the upstream transmission timing of ONUs. In order to satisfy these conditions, we introduced two methods for the virtual OLT: (1) Separation of D-Plane/C-Plane and (2) Smoothing RTT. Evaluation experiments show wire-speed throughput and low latency are achieved when the packet length is short, but simultaneously, a technical issue with long packet transmission.

Keywords: EPON, virtual OLT, SDN, NFV, Virtual access network

Classification: Network System

References

- [1] ETSI, “Network functions virtualization white paper,” 2012.
- [2] L. Peterson, A. Al-Shabibi, T. Anshutz, S. Baker, A. Bavier, S. Das, J. Hart, G. Palukar, and W. Snow, “Central office re-architected as a datacenter,” *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 96–101, 2016. DOI:10.1109/MCOM.2016.7588276
- [3] K. J. Kerpez, J. M. Cioffi, G. Ginis, M. Goldberg, S. Galli, and P. Silverman, “Software-defined access networks,” *IEEE Commun. Mag.*, vol. 52, no. 9, pp. 152–159, 2014. DOI:10.1109/MCOM.2014.6894466
- [4] A. Thyagaturu, A. Mercian, M. P. McGarry, M. Reisslein, and W. Kellerer, “Software Defined Optical Access Networks (SDOANs): A comprehensive survey,” arXiv:1511.04376v3, 2016.
- [5] K. Nishimoto, M. Tadokoro, T. Mochida, A. Takeda, T. Tanaka, and T. Inoue, “Virtualization of EPON OLT functions and collision suppression techniques for multi-point MAC control,” Proc. of OFC 2016, W3F.1, 2016.
- [6] Intel, “Intel® open network platform server (Release 1.2),” 2014.
- [7] C. Lee, C. Park, K. Jang, S. Moon, and D. Han, “Accurate latency-based congestion feedback for datacenters,” Proc. of USENIX ATC 15, pp. 403–415, 2015.

1 Introduction

One of the most prevalent ways of realizing the optical access network is the PON (Passive Optical Network). However, the flexibility of existing PONs will not well satisfy the diversifying demands from subscribers and service providers. In recent years, NFV (Network Function Virtualization), has attracted the attention of telecommunication carriers as a new approach to adding flexibility and greater cost-efficiency to carrier networks. NFV aims to decouple the network functions from dedicated hardware by implementing the functions as programs running on general-purpose servers (e.g. x86 servers) [1].

In this paper, we focus on the EPON (Ethernet PON), and aim to virtualize the EPON OLT (Optical Line Terminal) which will be required to offer greater flexibility because it is located nearest to the subscribers among the central office equipment and thus greatly influences the communication quality of subscribers. Currently, numerous studies related to “virtual” or “software-defined” PON and OLT are under way (e.g. CORD [2], SDAN [3] and a recent review [4]). However, most existing studies assume the dedicated hardware-based virtual OLT. Unlike these studies, we try to virtualize EPON OLT functions on an easy-to-use generic OS and low-cost, general-purpose servers in order to fully utilize the merits of virtualization. Our proposal is expected to enable rapid service introduction without hardware dependencies, and the reduction of the OLT equipment cost.

One of the big problems with network virtualization is the performance shortfall, in particular to packet transmission, because the software processing is generally slower than dedicated hardware processing. Moreover, unlike the other communication equipment, the EPON OLT demands a high level of timing accuracy for packet transmission/reception processing, so that it can control the upstream transmission timing of ONUs (Optical Network Unit) which share an optical fiber by TDMA, using MPCP (Multi-Point Control Protocol). Thus, it is necessary to prototype the software-based virtual OLT (vOLT) and evaluate its performance to confirm feasibility. In this paper, we present the concrete software architecture of vOLT, which was not shown in our previous work [5], and report evaluation results.

2 Architecture

Fig. 1 shows the architecture of the prototype. An x86 server is used as the platform, and software-implemented OLT functions run on the OS (Linux). We prepare Network Interface Cards (NIC) for the PON port (PON-NIC) and Service Node Interface port (SNI-NIC). In addition, we attach an external module with PON-NIC to translate the PON Ethernet frames into the normal Ethernet frames.

The performance targets of vOLT are as follows:

- ✓ Throughput: wire-speed (1 Gbps)
- ✓ Downstream latency: less than 100 μ s (In this paper, we do not focus on the upstream latency because it greatly depends on the DBA (Dynamic Bandwidth Allocation) algorithm which is not examined in this paper).

We will explain two technical methods adopted by our prototype in order to achieve these goals, and report the improvement effects of those.

Fig. 1. Software architecture and the figure of the prototype.

2.1 Separation of the data-plane and control-plane

In the dedicated hardware-based EPON OLT, the D-Plane functions (e.g. frame forwarding, VLAN-ID translation) and the C-Plane functions (e.g. DBA, Multi-Point Mac Control (MPMC), OAM) are tightly coupled. We separate their functions and run them as software processes independent of each other in order to reduce implementation dependency and accelerate performance by running these processes in parallel. The D-Plane process differentiates the Control messages from forwarding user-data frames, and apportions them between the C-Plane process and the NIC through ring buffers in the shared memory. The functions of C-Plane receive the control messages sent from ONUs through the D-Plane and send messages after processing.

The separation of D/C-Plane enables us to adopt acceleration technologies to the D-Plane regardless of whether they are hardware or software based. We select DPDK (Data Plane Development Kit) as the D-Plane which has been reported to realize high throughput, low latency [6] and low processing jitter [7] in forwarding packets.

In order to confirm the impact of the D/C-plane separation, we compared the performance with the previous prototype in which the D-Plane and C-plane were not separated, and uses a raw socket instead of DPDK.

We constructed a PON to connect to the x86 server running the prototype and an ONU (Fig. 2.(a)-(left)), and measured the performance. We added 72 Byte (we assume the double VLAN tagged frame) frames to the PON in one direction and incremented the frame rate by 10 Kpps and measured 10 secs $\times$ 10 trials at each rate. In order to measure the highest performance of the prototype, we regard the throughput as the maximum received packets per second with frame loss rate $<1\%$

in all 10 trials. Fig. 2(b) compares the throughput of both stream directions and the average latency of the downstream traffic. This shows the prototype's drastic improvement in both throughput and latency.

Fig. 2. Improvements by (1) D/C-Plane separation and (2) RTT smoothing.

2.2 Smoothing Round Trip Time (RTT)

In the EPON, the OLT transmits a GATE message to each ONU to grant the bandwidth, and the ONU sends back the REPORT message to request the bandwidth. In order to control the upstream transmission of ONUs precisely, the OLT synchronizes the ONUs via the GATE message which includes master clock information. Moreover, it measures the Round Trip Time of each ONU at every reception of the REPORT message. These process needs accurate timing to allow completion of the transmission and reception of MPCP messages (GATE/REPORT) in the OLT right on time without large jitter. However, vOLT uses software running on a generic OS for flexible frame handling. This causes large processing time jitter due to the frame queueing and IRQ (Interrupt ReQuest) processing, and prevents the MPCP protocol from working correctly. In our previous work [5], we speculated that upstream signal collision occurs due to two factors; (A) the time synchronization deviation between ONUs caused by the transmission jitter of GATE messages, and (B) the fluctuation in measured RTT caused by the transmission/reception processing of GATE/REPORT messages. Factor (A) is expected to be restrained by using DPDK. We mitigate the problem of factor (B) by smoothing RTT. In this method, the RTT for each ONU is measured and stored at every REPORT reception timing, and OLT uses the average of the past M values (M : memory length).

In order to analyze the smoothing effect, we prepared 4 ONUs connected to 1 vOLT (Fig. 2(a)), and show in Fig. 2(c)-(top) the time-series data of measured RTT and (bottom) the frequency of the REPORT loss derived from signal collision in the cases of (c)-(left) Original MPCP and (c)-(right) MPCP with smoothing RTT ($M = 1000$); 100 Mbps upstream traffic (72 Byte) was loaded on each ONU. The red line in Fig. 2(c) represents the start timing of the inflow of the traffic. We can observe large deviation of RTT in the left case (without smoothing), and frequent REPORT loss. In particular, successive REPORT losses occurred around the 6000th step just after the start of the traffic inflow. On the other hand, our smoothing proposal succeeded in suppressing the signal collisions which occurred at the initial stage.

3 Performance evaluations

We evaluated the throughput and the average latency of the upstream/downstream traffic, when 4 ONUs were connected to 1 vOLT (Fig. 3(c)). The evaluation results in Fig. 3(a) show vOLT achieved almost wire-rate transmission in both cases of upstream/downstream when the packet length was short. However, the performance in the upstream declined with long packet flows.

We consider that the main reason for the performance decline is the increase in reception/transmission jitter of MPCP messages caused by the long packet transmission. The data-forwarding part of vOLT processes MPCP messages with top priority. However, if the message arrives at the data-forwarding part while a user data packet is being processed, it has to wait until processing is finished. As

(c) Network setting for evaluation experiments.

Fig. 3. (a), (b) Performance evaluation results and (c) experimental setting.

Fig. 3(c) shows, the time necessary for packet transmission in vOLT is proportional to packet length, probably due to the memory copy step of the packets in the data-forwarding part. This means long user-data packets increase the maximum waiting time of MPCP messages such that the PON link fails due excessive jitter in MPCP message transmission/reception.

4 Conclusion

In this paper, we proposed the implementation architecture of vOLT and showed its improvements in throughput and latency due to (1) Separation of D-plane and C-plane and (2) Smoothing RTT. Evaluation experiments showed two key goals were achieved: wire-speed and less than 100 μ s latency in downstream traffic for short packets. However, they also disclosed the issue of a performance decline in long packet transmission due to an increase in the transmission/reception jitter of MPCP messages. This is a key problem for stabilizing vOLT performance, and will be the target of future research.