Evolution of access network sharing towards software based passive optical network: can virtual OLTs run at full line rate?

Marco Ruffini(1), David Coyle(2), Jasvinder Singh(2), Rory Sexton(2),
Brendan Ryan(2), Robin Giller(2), Michael O'Hanlon(2), Frank Slyne(1)

***Abstract*—This paper summarises how access network sharing has evolved from physical layer unbundling to fully virtualised architectures. We address one of the crucial question in access network virtualisation: whether fully virtualised PONs can run at line rate over general purpose processing units. This paper provides an experimental insight into this query, showing that it is possible to run XGS-PON at line rate, if additional hardware acceleration is provided, e.g., for cryptographic functions.**

***Index Terms*—DBA, virtualization, PON, NFV, SDN, access**

I. Introduction

Networks are fundamentally built on the notion of sharing. Starting from the concept of statistical multiplexing of different users and applications, over time, through deregulation and increase in competition, the concept of sharing was also implemented across network providers, which would share infrastructure by operating multi-tenancy solutions.

The past 25 years have seen, for example, access networks sharing evolving through multiple steps. The first sharing technology, the Local Loop Unbundling (LLU), focused on physical layer access, as Other Licensed Operators (OLOs) needed to get physical access to end user lines at each central office where service was provided. Although this removes the high cost part due to cabling, still required presence at each central office. Thus other virtual sharing options started to appear, in order to reduce cost for new entrants, so that OLOs could provide services to end users, by having their customers' data aggregated at regional Points of Presence (PoPs). Different technologies were developed, namely Bitsrtream, Virtual Unbundling Local Access (VULA) and next generation bitstream, offering different trade-offs between cost and ability to control the infrastructure. For example, LLU allows the highest level of control as new entrants operate their own physical layer equipment, but it requires new entrants to provide equipment locally, at each CO they operate, which is expensive. Bitstream instead enable entrants to use most of the equipment from incumbent operators, but have no control on the service quality. VULA and NGA bitstream have tried to provide some level of mitigation, by providing additional QoS differentiation. The work in [1] provides a survey of the evolution of such sharing mechanisms.

In the meantime, server virtualisition started to be applied in the computing domain, showing the potential that such technology could bring to infrastructure sharing. Thus, as the concept of function virtualisation made its way in the networking world, we has seen the development of a completely new model for sharing access network resources. The Central Office Rearchitected as a Data Centre (CORD) [2], adopts the concept of network functions virtualisation and scales it up to implement all the main functions typically found in a central office. The code implementation of this open source project, coordinated by the Open Networking Foundation (ONF), focused on three complementary use cases: residential broadband, mostly delivered through Passive Optical Networks, Mobile Broadband, making use of Software Defined Radio (SDR) technology, and enterprise, focusing on metro transport. Such use cases are continuously evolving, and the reader can refer to [3] for the latest developments.

II. Deep PON virtualistion

Concepts such as CORD were developed to provide full control to the OLOs, now better referred to as Virtual Network Operators (VNOs), for delivery of broadband services. Taking PONs as an example, virtualisation enabled dynamic sharing of resources across VNOs, which could take a share of the network performance and deliver arbitrary amounts of committed and peak data rate to its users. It should be noticed however, that in CORD the virtualisation is applied to the management layers of the system, i.e., VLAN configuration and setting of QoS parameters such as priority queues and Dynamic Bandwidth Allocation (DBA) assured and non-assured bandwidth values. The MAC layer however remains in hardware, meaning that this part is not virtualised and there is one common DBA mechanism that handles the scheduling for all VNOs. The main reason behind this choice was the computational cost of implementing the MAC in software (consider that this work originated in 2015). However, this mechanism is not applicable to future PON use cases that require low latency. Today for example, PONs are being increasingly considered for supporting Cloud-RAN and other low latency services, where controlling packet scheduling becomes essential to meet strict latency requirements. For example, this was experienced in the cooperative DBA [4] approach (standardised in [5]), which enables exchange of information between the C-RAN Distributed Unit (DU) and the OLT DBA mechanism, so that upstream scheduling can be coordinated in advance and the system can achieve the strict latency requirements needed for transporting the LLS interface. This, for example, cannot be

[1] CONNECT Centre, Trinity College Dublin marco.ruffini at tcd.ie

[2] Intel Corporation Ireland.

978-3-903176-21-8 © 2020 IFIP

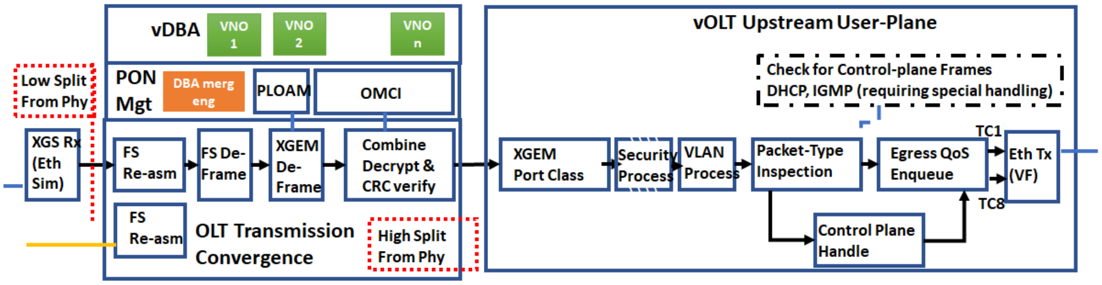

Fig. 1. vDBA and vOLT implementation over high-split and low-split architectures

achieved in CORD, as its virtual OLT does not have control of the DBA mechanism.

The idea of a software DBA, was proposed in the FASA architecture [6], allowing the DBA mechanism to be modified depending on the type of service it needs to support. However, this architecture was not developed with multi-tenancy as a primary goal. Our solution, initially presented in [7] and more recently consolidated in [8], proposes the concept of virtual DBA, where multiple DBA instances are created and operated independently by different virtual operators. Their outcomes are then merged together into a physical bandwidth map. This solution was standardized in [9], [10]. Its implementation was also demonstrated in [11]. This solution delivers both the full control of the scheduler (deep virtualisation) and the possibility of multi-tenancy. The virtual DBA solution also supports cooperative DBA and provides the further benefit of true multi-tenancy, as virtual operators can run different DBAs, to suit different services, in parallel in the same shared PON infrastructure [12]. For example, the vDBA enables the same physical PON to be divided into one slice offering C-RAN services through cooperative DBA, another slice serving residential broadband and another slice operating DBA for low latency applications. In PON, lower efficiency is typically a trade-off for low latency. Our vDBA system allows this lower efficiency to be restricted only to the low latency slice.

One key question on PON virtualisation is the achievable performance. This was indeed one of the main reasons CORD steered away from a software PON MAC. The question we address in this paper is whether our virtual PON can deliver line rate performance, while running as software over a general purpose Intel processor.

In this paper we first present some high-level implementation details and then show the performance we have achieved in our implementation, for two different levels of PON softwarisation, namely a high split and a low split.

III. High level implementation description

In our implementation, the vDBA is collocated with the Intel vOLT system (Fig. 1) running on a 16-core 2.3 Ghz Intel© Xeon© D-2100 processor server with 32 GB DDR4 DRAM.

The vOLT application is based on the DPDK packet-processing framework, which provides XGS-PON MAC processing in software that is compliant with XGS-PON standard specified in ITU-T G.9807.1 and BBF TR-156. When the DBA is architected using standard Linux operating network interfaces, DBRU packets are transited through a packet driver running in kernel space, and passed to the vDBA application running in user space. A number of issues arise with this approach, which leads to slow packet processing. Firstly, the hardware interface may generate an interrupt to the kernel for each packet processed in the kernel network stack. Secondly, the packet must be copied from a data structure in kernel space to a data structure in user space, which requires CPU processing. Thirdly, when the vDBA application is blocked awaiting the input of each packet, physical resources such as memory and locks are exclusively locked. Our vDBA application processes many tens of thousands of small DBRU packets every second, resulting in a large number of interrupts, packet copying and locking/unlocking of resources. This leads to a threshing of Layer 1 and 2 cache and memory and a high latency in packet processing, on a system otherwise busy with the calculation of DBA Bandwidth Maps. In order to reduce I/O virtualisation overhead and accelerate packet I/O, we use the DPDK Data Plane acceleration toolkit as well as improve packet processing performance within the vDBA. In particular, we use a user space device driver to provide fast access to the network interface card, which enables the vDBA within each VNO to send and receive packets from the network efficiently.

The vOLT supports two modes of operation, shown in Fig. 1. These modes are a high-split mode, for which OLT MAC transmission processing is offloaded to a PON hardware acceleration unit and a low-split mode for which OLT MAC transmission-convergence processing is done entirely in software. When the vOLT runs in low split mode, the XGTC, XGEM framing and User-Plane all run in software. This presents a higher load on the CPU than high split where the XG-PON transmission convergence layer (XGTC) and XG-PON encapsulation method (XGEM) framing is done through hardware accelleration and the User-Plane in software. Our virtual DBA scheduling mechanism runs in software for both the high-split and low-split mode.

IV. Performance analysis of the virtual PON

We have thus carried out experiments to assess the performance of the software OLT, comparing the high and low level split implementations, for varying traffic load. In our experiment, we configured one PON with 64 ONU's. There are 8 T-Conts and XGEM-ports off each ONU. There are 8 Traffic Class queues covering Management, Voice, Video and High Speed internet. We generated a range of packet sizes of 84 Bytes (15%), 256B (10%), 1280B (75%). Fig. 2 show the results of our performance tests, highlighting both the high level (yellow curve for downstream and gray for upstream) and low level (red for downstream and blue for upstream) splits. We varied input traffic generation at 20%, 40%, 60%, 80% and 100% of line rate levels for both high and low splits. For the High-Split scenario, we can see that the virtual PON can support full line rate of 10 Gb/s for all traffic loads, both in the upstream and downstream. The difference between upstream throughput and goodput (application) is of 200 about Mbps, which is the rate consumed by the DBRU traffic and burst overheads. The performance is however lower for the low-split scenario. Downstream traffic peaks at 4.9 Gbps for 100% offered traffic. Upstream traffic levels off at 3.3 Gbps for 60% offered traffic. The reason is that in low split, the cores need to carry out all the functions of the High Split, plus all the MAC framing functions. Furthermore, we are running the worst-case scenario, with all functionalities, such as FEC, CRC, upstream fragmentation turned on. Such performance will dramatically improve in the next generation of the Intel© Xeon© processor, as the Icelake generation will provide acceleration for cryptographic functions, in addition to other micro-architecture improvements.

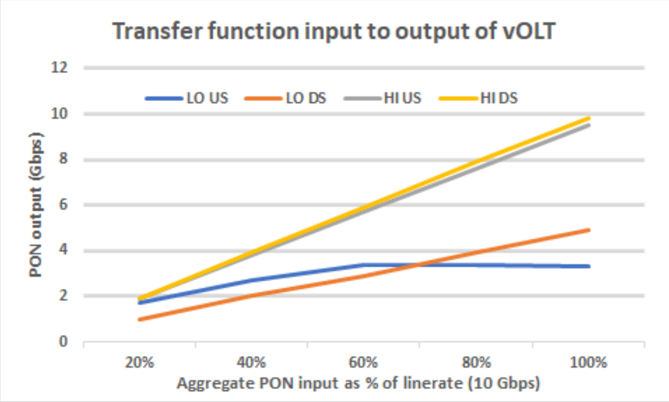

Fig. 2. Line rate performance of the vOLT for high and low split

Fig. 3 shows the processor utilisation on a 16-core CPU. Core 2 and 3 are assign to the Upstream and the Downstream of the vOLT respectively. Core 10 and 11 are assigned to the ONU emulation and packet generation. Cores 13, 14 and 15 are assigned to the merging engine, and the virtual DBAs for VNO's 0 and 1 respectively. The other cores run the Linux OS and other tools. 3 shows the CPU utilisation for the low split mode at full offered rate. One interesting observation we made, looking at additional utilisation graphs (not reported here) is that processors utilisation remains close to 100% for the vOLT and vONU processes, independently of the load. The reason is that our applications are developed using poll-mode drivers, which has advantages and disadvantages. Poll-mode means that the vOLT and vDBA applications run continuously regardless of the offered load. So the CPU utilisation is no different between traffic levels. While it may appear wasteful of CPU resources, particularly at low packet rates, poll-mode makes applications very responsive. This is different from traditional blocking I/O where the application waits for I/O, for example for a packet to arrive. If there are no packets waiting, the CPU utilisation for that thread goes to zero.

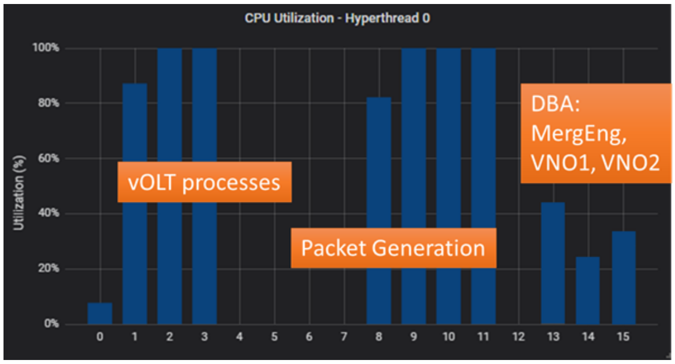

Fig. 3. CPU utilisation of the different VNFs

V. Conclusion

In this paper, we have analysed the line rate performance of a fully virtualised DBA and OLT implementation for an XGS-PON system. Using OpenSource frameworks such as the Linux Foundation DPDK, it is possible to make use of general purpose processors for running PONs, which can reduce substantial the cost of network infrastructure. Our performance showed that high level split, with external hardware acceleration, can run at full line rate. Low level split are not yet able to reach line rate performance, although this is due to improve with the upcoming Intel Icelake generation.

Acknowledgment

Financial support from SFI grants 14/IA/252 (O'SHARE) and 13/RC/2077 is gratefully acknowledged.

References

[1] N. Afraz et al., Evolution of Access Network Sharing and Its Role in 5G Networks. Applied Sciences. 2019; 9(21):4566
[2] L. Peterson et al. Central office re-architected as a data center. IEEE Commun. Mag. 2016, 54, 96–101
[3] Open Networking Foundation https://www.opennetworking.org/
[4] T. Tashiro, et al., A Novel DBA Scheme for TDM-PON based Mobile Fronthaul. OFC 2014, paper Tu3F.3.
[5] ITU-T, G989.3 Amd 2. 40-Gigabit-capable passive optical networks (NG-PON2): Transmission convergence layer specification Amendment 2. Nov., 2018.
[6] J. Kani et al., Flexible Access System Architecture (FASA) to Support Diverse Requirements and Agile Service Creation, in Journal of Lightwave Technology, vol. 36, no. 8, pp. 1510-1515, 15 April, 2018.
[7] A. Elrasad, Nima Afraz and Marco Ruffini. Virtual Dynamic Bandwidth Allocation Enabling True PON Multi-Tenancy. OFC 2017, paper M3I.3.
[8] M. Ruffini, et al., The Virtual DBA: Virtualizing Passive Optical Networks to Enable Multi-Service Operation in True Multi-Tenant Environments. JOCN, No.4, Vol.12, April 2020.
[9] BBF, TR-402 Functional Model for PON Abstraction Interface. Oct. 2018.
[10] BBF TR-370i2 Fixed Access Network Sharing - Architecture and Nodal Requirements, issue 2. Publication planned in 2020.
[11] F, Slyne, et al., Experimental Demonstration of multiple Disaggregated OLTs with Virtualised Multi Tenant DBA, over General Purpose Processor. OFC 2020.
[12] A. Ahmad, et al., Capacity sharing approaches in multi-tenant, multi-service PONs for low-latency fronthaul applications based on cooperative-DBA. OFC 2020.