# Frame Level Sharing for DBA Virtualization in Multi-Tenant PONs

Amr Elrasad, *Member, IEEE* and Marco Ruffini, *Senior Member, IEEE*

***Abstract*—The worldwide installation of Fiber-to-the-premises (FTTP) access network solutions is hindered by the high upfront cost of deploying ubiquitous fiber infrastructure. While passive optical networks can provide lower cost compared to point-to-point solutions, their total cost of ownership is still high for most operators to justify a mass scale deployment. Sharing passive optical network (PON) infrastructure has thus been proposed as a solution for network operators to reduce the cost of running FTTP services. In addition, the ability for operators to offer business services (including for example mobile backhaul) in addition to residential services, is crucial to increase the overall PON network revenue. However running services with highly diverse requirements over a physical infrastructure shared among multiple operators (which we now refer to as virtual network operators -VNOs) requires VNOs to have a tight control over PON capacity scheduling.**

**In this paper, we introduce a novel upstream PON capacity sharing algorithm called Frame Level Sharing (FLS). FLS is based on the idea of virtual Dynamic Bandwidth Assignment (vDBA), and allows sharing the upstream frame among multiple VNOs to maximize bandwidth utilization, minimize latency, and provide a high level of service isolation among the VNOs sharing the PON. Our simulation results show that FLS outperforms other benchmark algorithms proposed in the literature.**

## I. Introduction

Passive optical networks (PONs) are considered as one of the prominent access network solutions for delivering fiber to the home, due to the high capacity and coverage they can provide. Meanwhile, the high Capital expenditure (CAPEX) required for PON deployment has been an obstacle to large-scale adoption, especially in rural areas with a lower number of users and bandwidth demand. To this point, multiple solutions have been proposed in the past to improve the business case of access fiber deployments, stemming from changes in the overall network architecture [1], to development of cost-effective transceivers for multi-wavelength PONs [2]. A complementary approach to economic sustainability is to increase the revenue generated by the PON by increasing the number and types of services that can be supported [3], for example including mobile backhaul [4], fronthaul [5] and enterprise services, in addition to residential applications. Therefore, a scenario in which all the aforementioned services can coexist and operate on the same PON infrastructure is pivotal in increasing the utilization of the infrastructure, thus generating new revenue streams. However, current network sharing methods do not give strict control of capacity assignment to the operators running services over the shared network (which we refer to as Virtual Network Operators - VNOs), often generating latency values that make it impossible to support a number of business services. A set of suitable PON sharing-oriented solutions are thus required to enable coexistence of multiple VNOs with diverse service requirements.

In this paper, we present a novel PON sharing architecture called Frame Level Sharing (FLS) that is applicable to the concept of virtual Dynamic Bandwidth Assignment (vDBA) [6]. The idea behind vDBA is to give each VNO full control over the capacity scheduling algorithm associated to its virtual PON slice, so that for example it can guarantee strict latency and jitter requirement to selected applications, in addition to Peak and Committed Information Rates (PIR and CIR), while maintaining isolation between multiple VNOs.

The rest of the paper is organized as follows. Section II introduces the related research work on PON sharing, focusing in particular on the Slice Scheduler (SS) architecture proposed in [7], which will be used as a comparison metric for our proposed Frame Level Sharing (FLS) architecture. FLS is then described in detail in section III. The performance evaluation of our FLS algorithm against the SS algorithm, carried out through a C++ based simulator, is presented in section IV. Finally, conclusions are drawn in section V.

## II. Related Work

In this section, we present some of the relevant state of the art on PON sharing, focusing on shared access at different layers, such as IP Layer, Medium Access Control Layer, and Physical Layer.

In [8], the authors proposed a higher level (IP layer) PON sharing. They relay on introducing fixed access network as a service through the introduction of virtual private networks (VPNs). The high level sharing however does not allow virtual operators to control the PON scheduling functions, like dynamic bandwidth allocation (DBA), thus they do not have tight control over capacity assignment and virtually no control over latency and jitter.

In [7], the authors presented an approach to share the XG-PON Transmission Convergence (XGTC) frames among multiple network operators. We refer to their architecture as Slice Scheduler (SS), which is reported in Fig. 1. The SS architecture introduces a new layer, the Slice Scheduler, on top of the Transmission Convergence (TC) layer. This new layer acts as a circuit switch that allocates the entire PON capacity to one VNO for the duration of the upstream frame. The authors proposed two switching mechanisms, namely static and dynamic. In the static approach, the switching decision is based on the ratio of the minimum committed service rate of each VNO to the total PON capacity. In the dynamic approach, the decision is based on the effective transmission rate which is related to the bandwidth demand of each VNO. This means

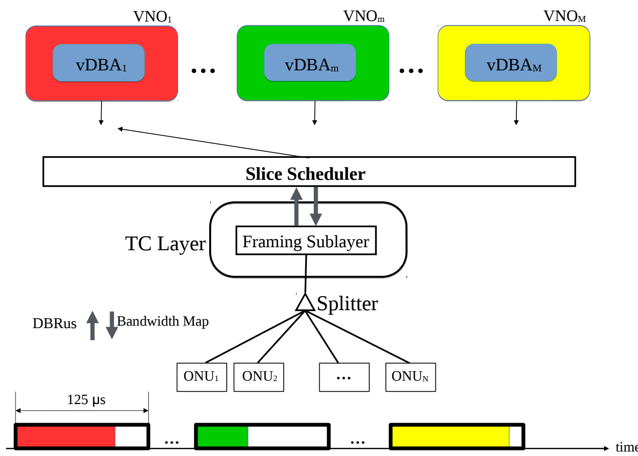



Fig. 1: Slice scheduler architecture, assigning different frames to different VNOs.

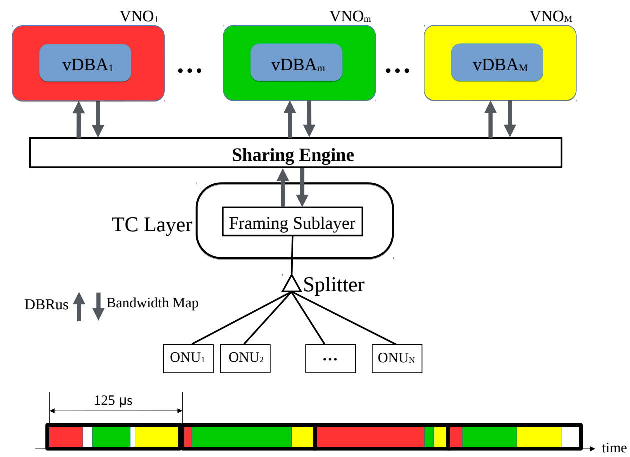



Fig. 2: Frame level sharing architecture, sharing frames among VNOs.

that overloaded VNOs will be allocated more whole upstream frames compared to underloaded NOs.

We found two main drawbacks in this SS architecture. First, the minimum achievable latency of VNOs is related to the number of running VNOs. For example, a VNO asking for 1 Gbps will be allocated an upstream frame every 10 frames on average. Consequently, the minimum latency of assured bandwidth [9] for that VNO will be 15 frame duration (1.5 $\times$ frame duration) [10], which can be excessive for some services (especially when considering low latency 5G applications). Second, the dynamic switching approach leads to bandwidth starvation and increased delay of the underloaded VNOs (see section IV for details). Consequently, this leads to poor isolation among VNOs.

Another form of PON sharing is physical layer sharing, with which the current technology aims at assigning different wavelength to different VNOs, for example in an NG-PON2 implementation. Although this guarantees maximum isolation between VNOs, it creates static wavelength allocation, which is inefficient, as shown in [11].

## III. Frame Level Sharing

The contribution of this work is to propose the Frame Level Sharing (FLS) architecture, shown in Fig. 2, to facilitate the coexistence of multiple VNOs in the PON. The proposed architecture falls under the category of medium access control layer sharing and includes a new layer, the sharing engine, to be placed on top of the PON transmission convergence (TC) layer. It should be noticed that for our work we normally refer to the XGS-PON standard, although the work can be applied to other types of ITU-T PON standards. In our FLS model, the bandwidth requests (DBRus) from the Optical Network Units (ONUs) are relayed by the TC layer to the vDBA instance of the corresponding VNO. Such virtual instance could physically run, for example, on a server in the central office, also owned by the PON infrastructure provider, following a model similar to that proposed by the Central Office Re-architected as a Datacentre project - CORD [12]. With this information each VNO can calculate a virtual Bandwidth Map (vBWMap), achieving full control over the capacity scheduling to the ONUs it serves.

The sharing engine layer shown in the figure has two main tasks. First, it is responsible for handling the communications to each vDBA instance. It passes the upstream buffer reports (DBRus) to the vDBA and it receives the corresponding virtual bandwidth maps from each VNO. Second, It performs a full analysis on all the received vBWMaps, merging them into one physical bandwidth map. This analysis includes both grant sizing approach and grant scheduling algorithm within the next upstream frame. This operation is critical to solve the contention between conflicting vBWMap allocations, and within the context of XGS-PON, we have defined two merging policies described in the following subsections.

### *A. No Capacity Sharing Policy*

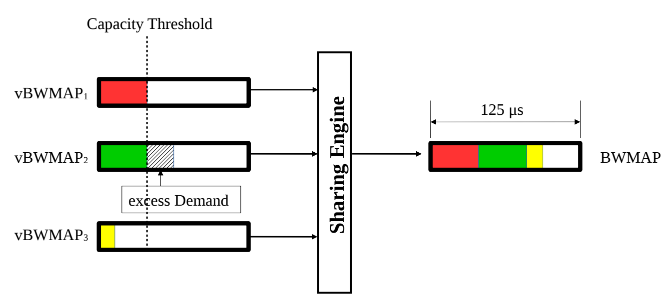



Fig. 3: FLS no capacity sharing policy.

In this policy, each running vDBA is aware of its allocated share of the upstream capacity, $C_i$. Thus, the cumulative grant size of the produced virtual bandwidth can not under any circumstances exceed the corresponding VNO share of the upstream frame. This policy helps reducing the sharing engine task complexity. The sharing engine shall keep the size of the received bandwidth grants as they are. It will only perform simple a scheduling algorithm to ensure that the complete bandwidth map has no overlapped grants. An illustration of this policy is shown in Fig. 3, where the second VNO can not allocate its excess demand bandwidth although there is enough bandwidth capacity because the third VNO is underloaded. This simple policy does not allow unused capacity of underloaded VNOs to be shared with the other overloaded VNOs. However, It ensures full isolation among VNOs. Unlike SS static mechanism, each VNO can be polled on each upstream frame to help meeting low latency requirements (see section IV).

### *B. Capacity Sharing Policy*

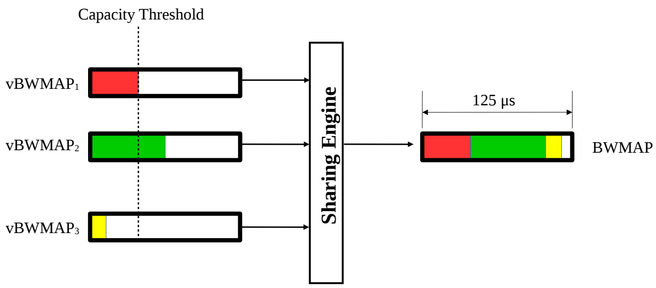



Fig. 4: FLS capacity sharing policy.

The capacity sharing policy, shown in Fig. 4, compromises between bandwidth utilization efficiency and VNOs isolation. The policy works as follows. Each vDBA acts as if it owns the whole PON capacity and produces the corresponding virtual bandwidth maps. Consequently, these vBWMaps can have cumulative grant size as large as the whole upstream frame. The sharing engine layer task is more sophisticated in order to process the virtual bandwidth maps from all VNOs and produce the final bandwidth map. Regarding grant sizing, the sharing engine acts as follows:

1) If the cumulative size of all bandwidth grants can be accommodated within the upstream frame, non of the virtual bandwidth map grant sizes is reduced.
2) If the cumulative size of all bandwidth grants is too big to be accommodated in one upstream frame, the bandwidth grants of overloaded VNOs are to be reduced in order to be fitted in the next upstream frame. Generally speaking, the bandwidth grants reduction process cuts the non guaranteed bandwidth grants starting from the most delay tolerant T-CONT and going higher if necessary. Within the context of T-CONTs defined in [9], the sharing engine layer starts reducing best effort traffic grants first. If it is still not enough, non-assured traffic bandwidth grants are also to be reduced.

Although in this paper we carry out an analysis based on a single-wavelength system, the concept can be easily extended to multi-wavelength systems, where for example to total PON capacity across multiple wavelengths can be virtualized into slices with fine granularity of capacity (for example virtual slices do not need to operate as 10G PONs, but could be dynamically assigned for example as 4G PONs or 18G PONs, as required). In addition, the idea can also be easily adapted to applications requiring very low latency, like fronthaul. For example the BBU could embed the DBA and generate the bandwidth map without waiting for DBRu messages, thus enabling tight synchronization between BBU and OLT as proposed in [13].

## IV. Performance Evaluation

We developed a C++ XGS-PON simulator (e.g., using symmetric 10G upstream/downstream rates) and used it to simulate one OLT and 60 ONUs with maximum physical distance of 40 Km. The upstream capacity was set to 9.95328 bps, according to the standard. The Ethernet frame size for the packet load generator ranges from 64 to 1518 bytes with trimodal size distribution, as reported in [14]. We employed self-similar traffic with long range dependence (LRD) and Hurst parameter 0.8. The ONUs are divided equally among VNOs and all VNOs employ the GIANT [9] DBA algorithm with three T-CONTs, namely: assured, non-assured and best effort. We considered service intervals of 4, 8, and 8 frames respectively. The ONU buffer size is set to 3 MB. The offered load is uniformly distributed among ONUs and T-CONTs. We assume the total PON assured traffic capacity is divided homogeneously among VNOs, but they are allowed to exceed this figure with non-assured and best effort traffic.

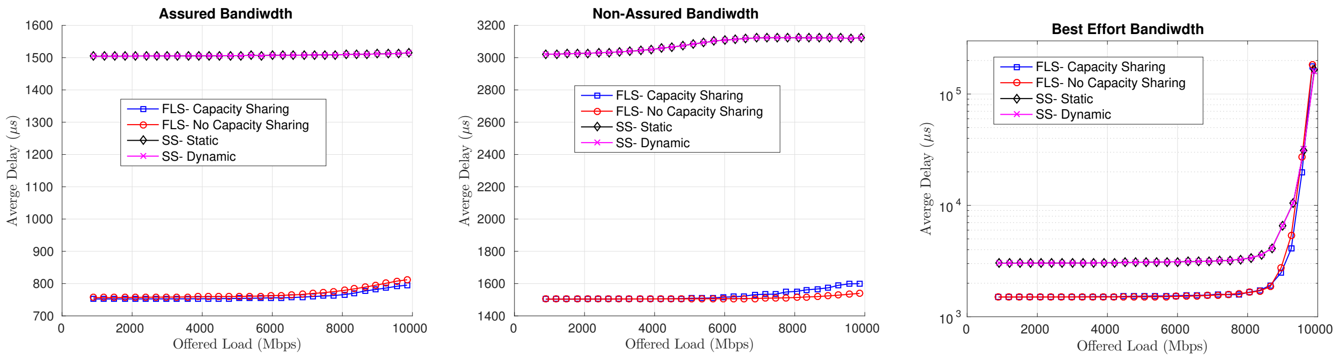
Regarding the number of VNOs and offered load distribution, we consider three simulation scenarios as follows:

- Scenario 1: we consider two VNOs with offered load divided equally among them.
- Scenario 2: we consider five VNOs with offered load divided equally among them.
- Scenario 3: we consider two VNOs with offered divided on a 1:2 ratio among them.

The performance of our FLS algorithm is tested against the SS framework [7], discussed in section II. In order to achieve a fair comparison between FLS and SS capacity sharing policies we use same number of VNOs and same service intervals (both mechanism are based on the GIANT DBA) and set the maximum service rate to the full XGS-PON capacity. The forgetting factor was set to 0.125 [7], while the minimum committed service rate was set equal to the share of the total PON capacity for each VNO. Our main performance metrics is the average packet delay, while we also investigate the frame loss rate.

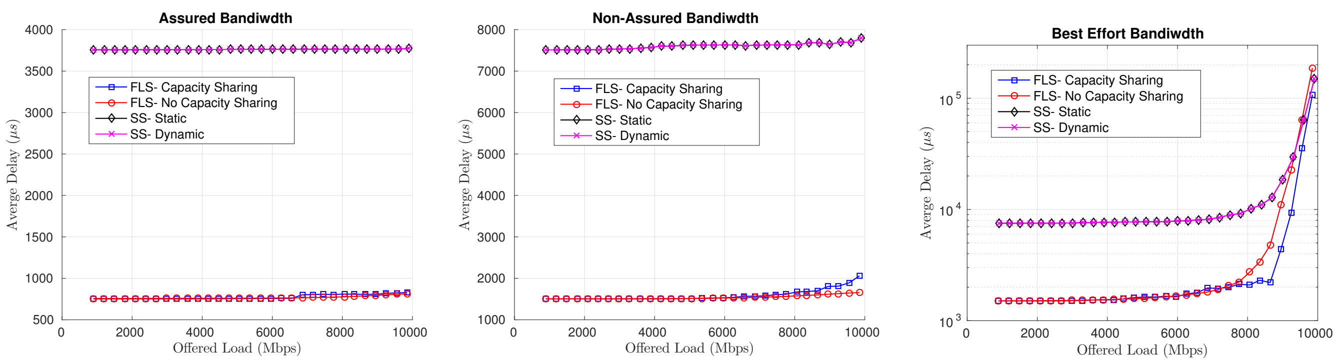
### *A. Scenario 1*

The average packet delay for our proposed FLS and the benchmark SS is shown in Fig. 5. It can be noted that both static and dynamic SS switching mechanisms have the same performance for assured, non-assured, and best effort traffic. Regarding FLS, both capacity sharing and no capacity sharing have very close performance. Capacity sharing has lower delay than no capacity sharing at high load for best effort traffic. This situation is reversed for non assured traffic. Comparing both SS and FLS, we find that FLS delay is significantly lower than SS delay by 50% for assured and non assured traffic. This statement is still true for best effort traffic for offered load below 9 Gbps. The minimum achieved delay by SS matches our note in Section II. Since the service interval




(a) (b) (c)

Fig. 5: Average delay (scenario 1): (a) Assured bandwidth (b) Non-assured bandwidth (c) Best effort.




(a) (b) (c)

Fig. 6: Average delay (scenario 2): (a) Assured bandwidth (b) Non-assured bandwidth (c) Best effort.

is equal to 4 for assured traffic and there are two VNOs, then the assured T-CONTs are polled every 8 frames. Hence the minimum average delay is 12 frames (1500 $\mu s$). The reported results in the original SS paper [7] show the same behavior. On the other hand, FLS allows assured T-CONTs to be polled every 4 frames. Hence FLS achieves 50% lower delay. The frame loss rate is similar in both SS and FLS, thus we do not report its plot.

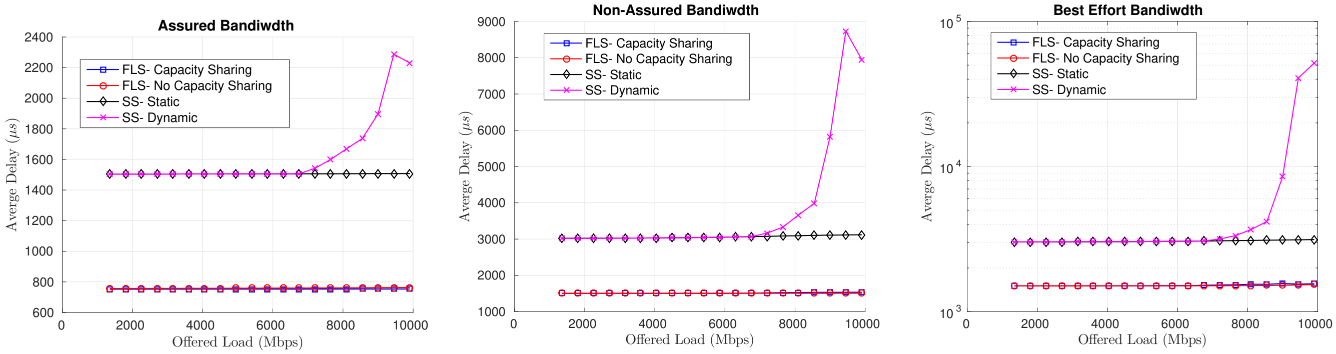
### *B. Scenario 2*

In scenario 2, the number of VNOs is set to 5. There are three interesting points to note. First, in FLS for best effort traffic the capacity sharing policy shows significant lower delay at high load compared to no capacity sharing. Second, the minimum achieved latency for FLS is still the same as in scenario 1. This shows that FLS is more resilient to the number of VNOs compared to SS. Third, the minimum achieved latency of SS is increased by a factor of 2.5, since, as explained in the subsection above, it is proportional to the number of VNOs in the system. This shows that SS framework performance is highly dependent on the number of VNOs.

### *C. Scenario 3*

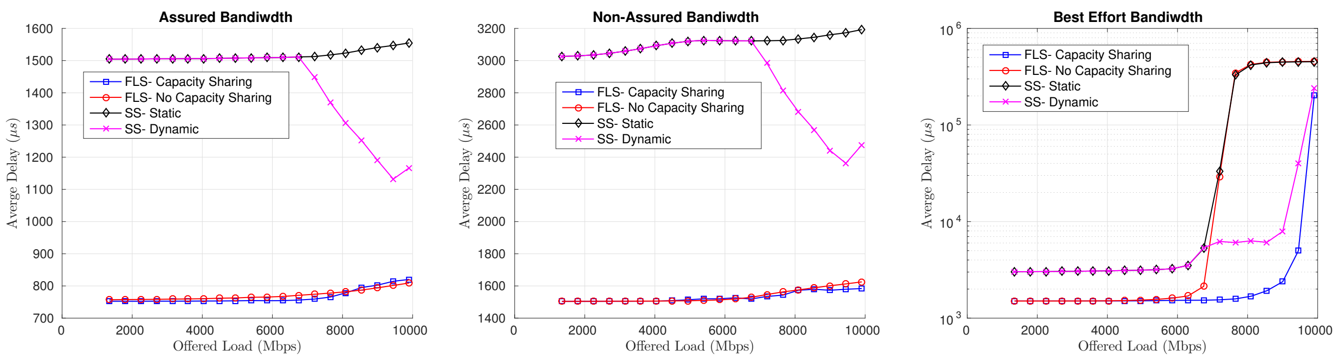
In Scenario 3, the number of VNOs is set to 2, but the offered load of one VNO is twice that of the other VNO. The delay performance of the low loaded operator is shown in Fig. 7, while the high loaded one is shown in Fig. 8. Comparing assured bandwidth performance for both operators, we see that FLS achieve higher isolation than SS, as we notice that the assured bandwidth delay of the lower loaded operator is almost constant over the load range for both the capacity and no-capacity sharing policies. On the other hand for SS, the dynamic switching mechanism achieves increasing delay for the lower loaded operator and decreasing delay for the higher loaded operator at high offered load. This is because as the offered load increases, the SS layer (dynamic) assigns more upstream frames to the higher loaded operator. This leads to reduced delay for the higher loaded operator and increased delay for the lower loaded one. Regarding Best effort traffic, FLS capacity sharing policy achieves significant lower delay compared to no-capacity sharing policy and both SS switching mechanisms.

The frame loss rate is reported in Fig. 9. For the lower loaded operator, the SS dynamic approach increases the frame loss rate by a small amount. For higher loaded operator, the SS dynamic approach and FLS capacity sharing policy are more stable than the SS static approach and FLS no-capacity sharing policy. However, the FLS capacity sharing approach has the advantage of not raising either the lower loaded operator frame loss rate nor the average delay, thus providing again good



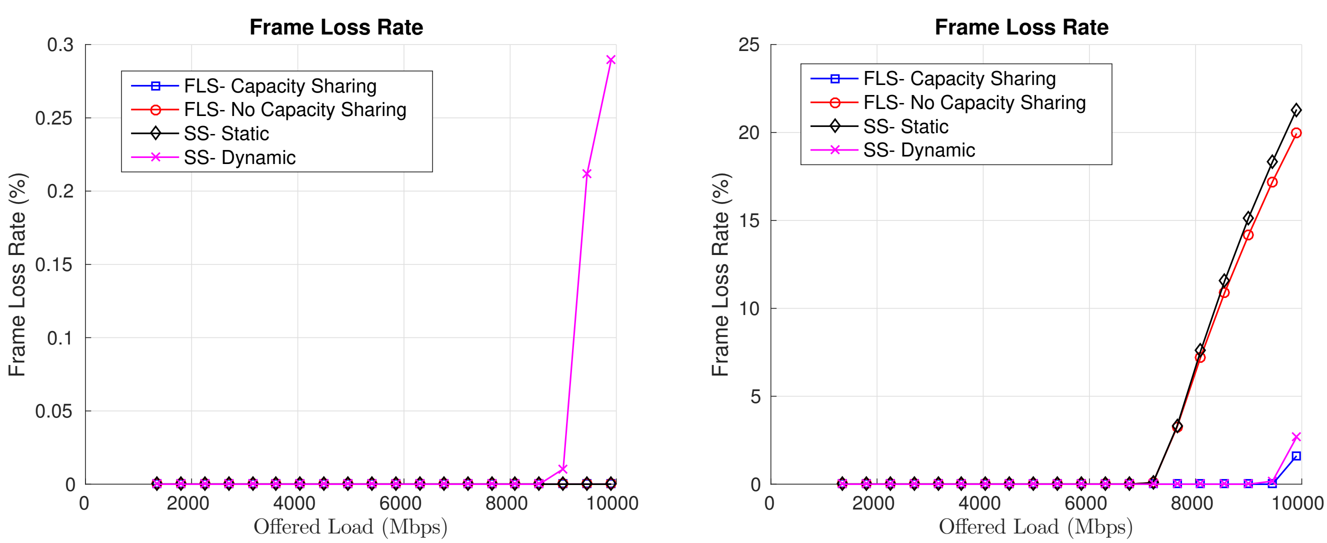

(a) (b) (c)

Fig. 7: Average delay (scenario 3, low loaded VNO): (a) Assured bandwidth (b) Non-assured bandwidth (c) Best effort.




(a) (b) (c)

Fig. 8: Average delay (scenario 3, high loaded VNO): (a) Assured bandwidth (b) Non-assured bandwidth (c) Best effort.




(a) (b)

Fig. 9: Scenario 3: Frame loss rate (a) Low loaded VNO (b) High loaded VNO.

isolation between the two VNOs.

## V. Conclusion

In this work, we proposed a novel virtualized PON sharing architecture called Frame Level Sharing (FLS). FLS introduces the concept of virtualization by migrating and virtualizing the DBA function from the physical OLT (owned by the infrastructure provider) to a virtual PON slice controlled by the virtual network operator. FLS is designed to achieve upstream frame level sharing among VNOs while maintaining service isolation among them, by introducing a new sharing engine layer on top of the TC layer. The sharing engine is responsible for merging the received virtual bandwidth maps into the physical bandwidth map to be transmitted along with the downstream frame. Simulation results in balanced load scenarios shows that FLS achieves less delay compared to a benchmark scheme (the Slice Scheduler) found in literature, and a low dependency on the number of VNOs sharing the PON. In addition, even for non-balanced load scenario, FLS achieves excellent service isolation among VNOs.

## Acknowledgment

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) and is co-funded under the European Regional Development Fund under Grants Number 14/IA/2527 (OSHARE) and 13/RC/2077 (CONNECT).

## References

[1] M. Ruffini, N. Doran, M. Achouche, N. Parsons, T. Pfeiffer, X. Yin, H. Rohde, M. Schiano, P. Ossieur, B. O'Sullivan *et al.*, "Discus: End-to-end network design for ubiquitous high speed broadband services," in *Transparent Optical Networks (ICTON), 2013 15th International Conference on*. IEEE, 2013, pp. 1–5.

[2] N. Cheng, J. Gao, C. Xu, B. Gao, D. Liu, L. Wang, X. Wu, X. Zhou, H. Lin, and F. Effenberger, "Flexible twdm pon system with pluggable optical transceiver modules," *Optics express*, vol. 22, no. 2, pp. 2078–2091, 2014.

[3] M. Ruffini, "Multi-dimensional network convergence in future 5g networks," *Journal of Lightwave Technology*, vol. 35, no. 3, pp. 535–549, 2017.

[4] P. Alvarez, N. Marchetti, and M. Ruffini, "Evaluating dynamic bandwidth allocation of virtualized passive optical networks over mobile traffic traces," *Journal of Optical Communications and Networking*, vol. 8, no. 3, pp. 129–136, 2016.

[5] P. Chanclou, A. Pizzinat, F. Le Clech, T.-L. Reedeker, Y. Lagadec, F. Saliou, B. Le Guyader, L. Guillo, Q. Deniel, S. Gosselin *et al.*, "Optical fiber solution for mobile fronthaul to achieve cloud radio access network," in *Future Network and Mobile Summit (FutureNetworkSummit), 2013*. IEEE, 2013, pp. 1–11.

[6] A. Elrasad, N. Afraz, and M. Ruffini, "Virtual dynamic bandwidth allocation enabling true PON multi-tenancy," in *Optical Fiber Communication Conference (OFC)*, 2017.

[7] C. Li, W. Guo, W. Wang, W. Hu, and M. Xia, "Bandwidth resource sharing on the xg-pon transmission convergence layer in a multi-operator scenario," *Journal of Optical Communications and Networking*, vol. 8, no. 11, pp. 835–843, 2016.

[8] B. Cornaglia, G. Young, and A. Marchetta, "Fixed access network sharing," *Optical Fiber Technology*, vol. 26, pp. 2–11, 2015.

[9] H.-C. Leligou, C. Linardakis, K. Kanonakis, J. D. Angelopoulos, and T. Orphanoudakis, "Efficient medium arbitration of FSAN-compliant GPONs," *international journal of communication systems*, vol. 19, no. 5, pp. 603–617, 2006.

[10] A. Elrasad and B. Shihada, "Parallel void thread in long-reach ethernet passive optical networks," *Journal of Optical Communications and Networking*, vol. 7, no. 7, pp. 656–668, 2015.

[11] M. Ruffini and D. B. Payne, "Business and ownership model case studies for next generation FTTH deployment," Feb 2016, White Paper. [Online]. Available: http://img.lightreading.com/downloads/Business-and-ownership-model-case-studies-for-next-generation-FTTH-deployment.pdf

[12] Open Cord project. [Online]. Available: http://opencord.org/

[13] T. Tashiro, S. Kuwano, J. Terada, T. Kawamura, N. Tanaka, S. Shigematsu, and N. Yoshimoto, "A novel dba scheme for tdm-pon based mobile fronthaul," in *Optical Fiber Communications Conference and Exhibition (OFC), 2014*. IEEE, 2014, pp. 1–3.

[14] D. Sala and A. Gummalla, PON functional requirements: services and performance, in IEEE 802.3ah Meeting in Portland OR, July 2001. [Online]. Available: http://www.ieee802.org/3/efm/public/jul01/presentations/sala_1_0701.pdf.