A Reinforcement Learning-based Dynamic Bandwidth Allocation for XGS-PON Networks

Abdullah Quran[1], Sebastian Troia[1], Omran Ayoub[2], Nicola Di Cicco[1], Massimo Tornatore[1]
[1]*Politecnico di Milano, Milan, Italy* [2]*University of Applied Sciences of Southern Switzerland, Switzerland*
{abdullah.quran@mail.polimi.it}

***Abstract*—Time-division-multiplexing passive optical networks (TDM-PONs), with their massive deployment worldwide, are considered a fundamental technology for supporting not only traditional Internet broadband services, but also for new emerging 5G latency-sensitive services, such as Ultra-Reliable and Low Latency Communications (URLLC). Traditional dynamic bandwidth allocation (DBA) mechanisms, currently used to allocate network resources in TDM-PONs, are not suited to meet the requirements of these new services with strict latency requirements, as they use a polling mechanism which can result in a high queuing delay and ultimately violate URLLC latency requirements. In this work, we propose a new predictive-based DBA mechanism for Gigabit Symmetrical PON (XGS-PON) that allows to reduce the latency to fulfill requirements of emerging latency-sensitive services. Our solution employs reinforcement learning (RL) to predict the ingress buffer occupancy of ONUs in the next DBA cycle. Results show that the proposed RL method outperforms traditional DBA approaches in terms of upstream delay while maintaining similar frame loss ratio.**

***Index Terms*—Passive Optical Network, Dynamic Bandwidth Allocation, Reinforcement Learning**

I. Introduction

URLLC services, such as vehicle-to-vehicle communications, health care and tactile Internet applications, feature very strict low-latency requirements [1], which pose critical challenges in the design of 5G, and beyond, networks. Time division multiplexing passive optical networks (TDM-PONs) (and in particular, the Gigabit Symmetrical PON (XGS-PONs) considered in this work), given their massive deployment worldwide, are being considered as a candidate solution for cost-effectively providing traffic aggregation in low-latency 5G networks [2]. In XGS-PON, an Optical Line Terminal (OLT) is connected to multiple Optical Network Units (ONUs), and an algorithm for dynamic bandwidth allocation (DBA) [3]–[5] is run inside the OLT to allocate bandwidth resources to upcoming service traffic requests from the ONUs. More specifically, conventional DBA mechanisms adopt a polling-based approach to gather information regarding the ONUs service traffic requests stored in their buffer. First, the OLT asks the ONU to report its buffer occupancy. After that, the OLT makes use of the buffer occupancy reports to allocate network bandwidth efficiently. However, polling-based DBA mechanisms can incur in high queuing delay, especially for service traffic bursts that may arrive just after the ONU report is sent. As such, currently XGS-PONs cannot fully meet the Quality of Service (QoS) of low-latency services.

In the literature, novel DBAs that feature advanced reservation-based mechanisms to support low-latency traffic flows (e.g., flows associated to fronthaul traffic) [6] have been discussed but they require redesigning the mobile radio network. Moreover, other predictive-based DBAs mechanisms have been proposed that are based on statistical modelling and supervised learning based models [7], [8]. However, they rely on historical data collection mechanisms to train the models. The training method is offline, so they need to be validated and tested before being used. In this work, we propose a novel reinforcement learning (RL)-based DBA mechanism that leverages ONU buffer occupation reports in an online fashion to predict future traffic requests. The OLT utilizes these predictions to run the DBA algorithm beforehand reducing the delay experienced by conventional DBA approaches, thus meeting the requirements of URLLC services. Our approach is based on online training, i.e. our models learn while they operate eliminating the need of offline training. Online training offers a significant advantage in both speed and feasibility when deploying real XGS-PON-based networks, as gathering large training sets for offline training may not be viable. We compare the proposed solution with 3 different DBA mechanisms for XGS-PON: 1) GigaPON Access Network (GIANT) [3] DBA; 2) Improved Bandwidth Utilization (IBU) [4] DBA; and 3) a custom predictive-based DBA based on Long Short-Term Memory (LSTM) inspired by the method in [7]. To mimic real-world traffic, we used a self-similar traffic generator. Our preliminary results show that, when network capacity is not fully saturated, the proposed RL-DBA mechanism outperforms Traditional methods in terms of latency, while not significantly increasing the frame loss ratio.

II. System Model and Problem Formulation

We consider an XGS-PON consisting of an OLT and N ONUs. We assume three types of Transmission Containers (T-CONTs) (i.e., T-CONT 2, T-CONT 3, and T-CONT 4 traffic, while we ignore T-CONT 1 as it has a static bandwidth allocation regardless of network state) with different QoS requirements including priority and data rate as defined in the XGS-PON standard [9]. The upstream link is shared among ONUs and traffic transmission is scheduled using TDM. The goal of the DBA algorithm is to guarantee the latency requirements defined in the service level agreements (SLAs). Each ONU buffer stores the bursts of multiple end-users belonging to the same T-CONT class. ONUs have a finite buffer size to store incoming bursts. Arriving bursts to a full buffer are dropped.

978-3-903176-44-7

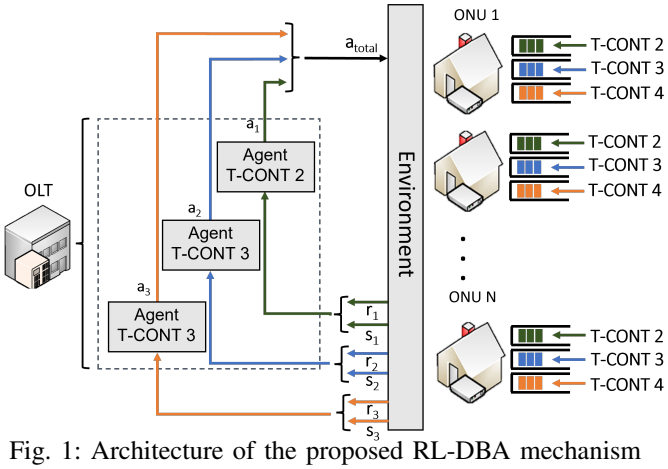


Fig. 1: Architecture of the proposed RL-DBA mechanism

III. RL-BASED DYNAMIC BANDWIDTH ALLOCATION

Figure 1 shows the schematic of the proposed RL-DBA. The RL agents are implemented as part of the OLT. The RL state encodes information about the ONUs buffer occupancy. At every step, buffer occupancy is used by the RL agents to predict the subsequent buffer occupancy of each ONU. The predicted occupancy value is then used by the algorithm to generate the final allocation bandwidth map (BWmap). The RL action represents the bandwidth allocation decision taken by the OLT every DBA cycle. After taking an action, the agent observe a reward that reflects the accuracy of the predicted values of ONUs buffers occupancy.

As shown in Figure 1, we use a separate RL agent to predict the traffic of each T-CONT. The state of the system (S_{total}) is composed of the state of three agents (i.e., S_{s1}, S_{s2}, and S_{s3}), and the total action (a_{total}) consists of actions taken by each agent (i.e., a_1, a_2, and a_3). Three separate rewards (i.e., r_1, r_2, and r_3) are received by the three agents. The rationale behind using a separate RL agent for each T-CONT is threefold. First, the traffic of each T-CONT is independent of the other T-CONTs. Second, multiple agents operating on the same environment have a much shorter convergence time on optimal solutions than using just one. Third, the quality of the actions of the single agents is far superior to that of the single agent because each of them has the sole objective of optimizing the performance of the single T-CONT. This work focuses on showing the performance of a RL solution based on multiple agents, while comparisons against a single centralized agent are left as future work.

Equation 1 defines the total state (S_{total}) of the RL model, which comprises the states of each RL sub-agent used for each T-CONT type, where S_{s_1}, S_{s_2}, and S_{s_3} represent the sub-state of T-CONT-2, T-CONT-3, and T-CONT-4 respectively. Equation 2 shows the sub-state of each T-CONT. The dimensionality of a sub-state is equal to the number of ONUs. Equation 3 shows the state of ONU i (S_{ONU_i}) at time step t_n, where r_{n-1} is the buffer occupancy of ONU_i at the previous time step t_{n-1}, and b_{n-1} is the size of the burst that arrived at the ingress buffer of the ONU_i during the previous timestep.

$$S_{total} = \begin{bmatrix} S_{s_1}, & S_{s_2}, & S_{s_3} \end{bmatrix} \tag{1}$$

$$S_{s_j} = \begin{bmatrix} S_{ONU_1}, & S_{ONU_2}, & \dots & S_{ONU_N} \end{bmatrix} \tag{2}$$

$$S_{ONU_i} = \begin{bmatrix} r_{n-1}, & b_{n-1} \end{bmatrix} \tag{3}$$

Equation 4 defines the total action a_{total} taken by the RL agent. The a_{total} consists of the predicted buffer occupancy of each ONU for various T-CONTs. The a_{total} is then used in the DBA algorithm to make the final allocation decision and generate the BWmap. g_{ij} represents the predicted value for the buffer occupancy of T-CONT j for ONU_i.

$$a_{total} = \begin{bmatrix} g_{11} & g_{12} & \dots & g_{1N} \\ g_{21} & g_{22} & \dots & g_{2N} \\ g_{31} & g_{32} & \dots & g_{3N} \end{bmatrix} \tag{4}$$

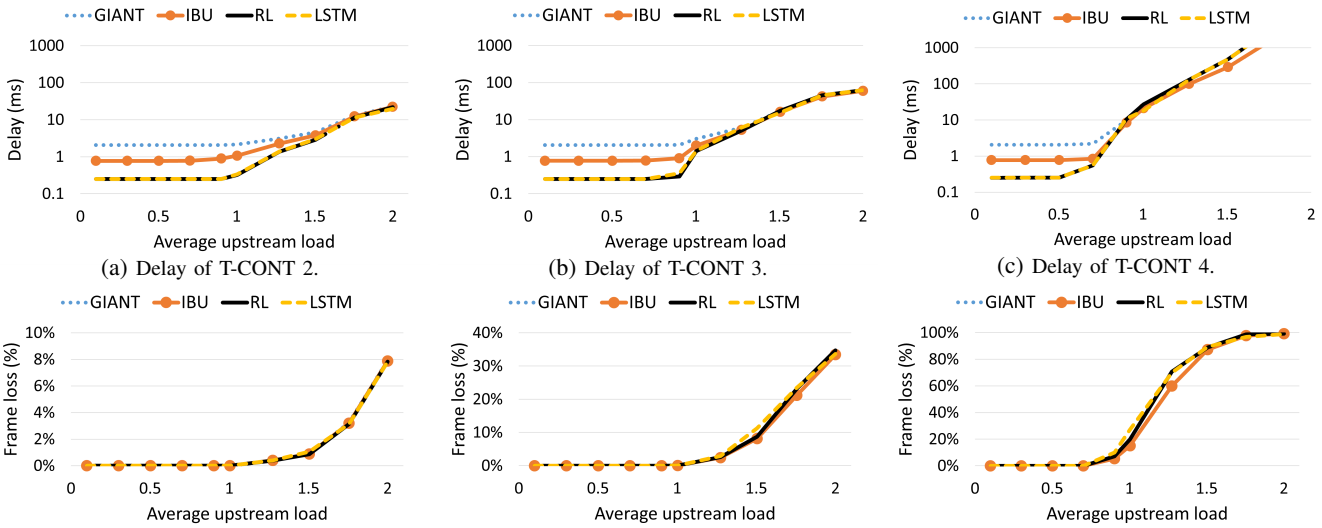
Each sub-agent receives a reward after taking an action. The goal of our RL system is to reduce the delay of XGS-PON traffic by predicting the ONUs buffer occupancy. Equation 5 shows the formula to compute the reward R_j for T-CONT-j. We adopt an exponential reward that varies between 0 and 1. The values of the base parameter and β are chosen experimentally to enable both a gradual move toward the objective reward and to give a relatively high reward value when entering the terminal state, i.e., the state in which the predicted value equals the actual buffer value. r_i represents the actual buffer occupancy of ONU_i at time t, and g_i is the predicted occupancy for the same ONU at the same time. The reward value observed by the agent depends on the accuracy of the predicted value. The closer the predicted value g_i to the real buffer occupancy r_i, the higher the reward.

$$R_j = \sum_{i \in S_{s_j}} 1.1^{-\beta |r_i - g_i|} \tag{5}$$

The RL model was trained using the Proximal Policy Optimization (PPO) algorithm [10] with a recurrent policy (i.e., LSTM). PPO is among the state-of-the-art learning algorithms for environments with continuous action spaces and continuous rewards. LSTM was adopted because of its ability to capture long-range dependencies in the self-similar traffic [11]. We trained the RL agent to predict the buffer occupancy of a T-CONT in the next service interval (SI) by exploiting the previously received buffer occupancy reports. Then, the bandwidth map BWmap is generated based on the predicted values. We trained the RL using traffic generated based on a self-similar model. Then, the model was tested on self-similar traffic with different burst sizes and inter-arrival rates.

IV. ILLUSTRATIVE NUMERICAL RESULTS

We compare our proposed RL-DBA mechanism to other three approaches, GIANT [3], IBU [4], and an offline supervised learning-based DBA using LSTM (henceforth LSTM) inspired from [7]. GIANT serves as a baseline due to its simple design. IBU is a refined version with more advanced functions. LSTM is a predictive-based algorithm based on offline supervised learning. Comparison is carried on in terms of average upstream delay and frame loss ratio. We conducted our experiments in a simulation environment that consists of an OLT and 8 ONUs with a network capacity of 10 Gbps



(a) Delay of T-CONT 2. (b) Delay of T-CONT 3. (c) Delay of T-CONT 4.

(d) Frame loss of T-CONT 2. (e) Frame loss of T-CONT 3. (f) Frame loss of T-CONT 4.

Fig. 2: Average upstream delay and frame loss ratio of all traffic classes

in upstream and downstream links. The traffic follows a self-similar shape generated by a Poisson Pareto Burst Process (PPBP). The simulation environment was built in Python using SimPy which is a discrete event simulation package and the RL environment was developed using OpenAI Gym toolkit.

Figures 2a, 2b, and 2c show the upstream delay for T-CONTs 2, 3, and 4 respectively. RL and LSTM are both predictive-based mechanisms and reach similar performance, both outperforming GIANT and IBU. RL and LSTM reduce T-CONT 2 delay by up to 67% and 88% compared to IBU and GIANT when the load is below 100% of the network capacity. RL and LSTM achieve this performance by bringing the queuing delay near to zero values as both mechanisms allocate bandwidth for packets beforehand (note that, for low traffic, the delay of RL is dominated by the propagation delay which is 250 μs). For T-CONT 3, the proposed RL method reduces the delay of IBU by 70% when the load is below 80% of the network capacity. For T-CONT 4, RL and LSTM methods reduces the delay by 32% compared to IBU. However, as the average load increases above 75%, T-CONT 4's excess bandwidth decreases as the network is used to serve traffic from other, higher-priority, T-CONTs.

Figures 2d, 2e, 2f show the frame loss ratio for T-CONTs 2, 3, and 4 respectively. For T-CONT 2, we can see that all the mechanisms have no frame loss when the load is below the network bottleneck and similar frame loss ratio when the load is higher than 150% of the network capacity. For T-CONT 3 and T-CONT 4, for higher upstream loads, our RL approach (and, similarly, LSTM approach) results in slight increase of 3% and 9% in frame loss ratio compared to GIANT and IBU. This is because RL overprovisioned bandwidth for T-CONT 2 traffic due to prediction errors, resulting in wasted bandwidth that could have been allocated to T-CONT 3 and T-CONT 4. Nonetheless, the significant reduction in the average upstream delay outweighs this increase in frame loss ratio.

V. Conclusion

This paper investigates the use of RL-based predictive DBA methods to reduce the latency in XGS-PON with the goal of accommodating latency-sensitive services. Our solution employs an RL-based DBA algorithm to predict the buffer occupancy of ONUs in the next DBA cycle. Results show that the proposed RL method outperforms other approaches in terms of upstream delay while maintaining similar frame loss ratio. Despite the similar performance of RL and LSTM methods on the upstream delay, they differ on the principle of operation. LSTM training is executed offline and requires large training data set to train the model while RL training operates online requiring no generation of training data set.

References

[1] Y. Ji *et al*, "5G flexible optical transport networks with large-capacity, low-latency and high-efficiency," *China Communications*, vol. 16, no. 5, pp. 19–32, 2019.

[2] H. Uzawa *et al*, "First demonstration of bandwidth-allocation scheme for network-slicing-based TDM-PON toward 5G and IoT era," in *Optical Fiber Communication Conference (OFC)*, 2019.

[3] L. Helen *et al*, "Efficient medium arbitration of fsan-compliant gpons: Research articles," *International Journal of Communication Systems*, vol. 19, pp. 603 – 617, 06 2006.

[4] R. A. B. *et al*, "Improved dynamic bandwidth allocation algorithm for XGPON," *J. Opt. Commun. Netw.*, vol. 9, no. 1, pp. 87–97, Jan 2017.

[5] M. Han *et al*, "Development of efficient dynamic bandwidth allocation algorithm for XGPON," *ETRI Journal*, vol. 35, 02 2013.

[6] J.-i. Kani, J. Terada, K.-I. Suzuki, and A. Otaka, "Solutions for future mobile fronthaul and access-network convergence," *Journal of Lightwave Technology*, vol. 35, no. 3, pp. 527–534, 2017.

[7] A. M. Mikaeil *et al*, "Traffic-estimation-based low-latency XGS-PON mobile front-haul for small-cell C-RAN based on an adaptive learning neural network," *Applied Sciences*, vol. 8, no. 7, 2018.

[8] K. A. Memon *et al*, "Demand forecasting DBA algorithm for reducing packet delay with efficient bandwidth allocation in XG-PON," *Electronics*, vol. 8, no. 2, 2019.

[9] ITU, "G.9807.1: 10-gigabit-capable symmetric passive optical network (XGS-PON)," Tech. Rep., 2016.

[10] J. S. *et al*, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, Nov. 1997.