AMERICAN UNIVERSITY OF BEIRUT

Predictive Resource Management using Deep Learning in Next Generation Passive Optical Networks

by
John Abied Mitri Hatem

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Science
to the Department of Computer Science
of the Faculty of Arts and Sciences
at the American University of Beirut

Beirut, Lebanon
December 2018

AMERICAN UNIVERSITY OF BEIRUT

Predictive Resource Management using Deep Learning in Next Generation Passive Optical Networks

by

John Abied Mitri Hatem

Approved by:

Dr. Ahmad R. Dhaini, Assistant Professor — Advisor

Computer Science

Dr. Shady Elbassuoni, Assistant Professor — Member of Committee

Computer Science

Dr. Haidar Safa, Professor — Member of Committee

Computer Science

Dr. Fatima K. Abu Salem, Associate Professor — Member of Committee

Computer Science

Date of thesis defense: December 21, 2018

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Hatem (Last) John Abied (First) Mitri (Middle)

☑ Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☑ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☐ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One ___ year from the date of submission of my thesis, dissertation or project.**
Two ___ years from the date of submission of my thesis , dissertation or project.
Three ___ years from the date of submission of my thesis , dissertation or project.

Signature

23-Jan-2019

Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

Acknowledgements

First and foremost, I wish to express my sincere thanks to my thesis supervisors. Prof. Ahmad Dhaini, for his invaluable and generous advice, expert and detailed guidance in research and writing, the illuminating discussions which steered the research, and support on every step of this work. Prof. Shady Elbassouni for his insightful guidance, engaging observations, kindness and continuous support throughout this work, and the vital knowledge I acquired through his Machine Learning graduate course offering. Prof. Dhaini's and Prof. Elbassuoni's enthusiasm and dedication to excellence in research have extremely contributed to this thesis. It was a great opportunity to study and work under their supervision.

I would also like to thank Prof. Haidar Safa for his astute advice, support throughout my time at AUB, and his Advanced Computer Networks graduate course offering which got me interested in the field of networks.

Next, I would like to thank Prof. Haidar Safa and Prof. Fatima Abu Salem for serving on my thesis examining committee, and for their constructive comments and suggestions.

My profound gratitude to my grandmother and grandfather who thought me to never give up and persevere. My sincerest thanks to my parents and brother for their unfailing support and endless giving love throughout my life.

My warmest thanks and love go to Rev. Soghomon and Esther Kilaghbian for their non-stop support, encouragement, friendship, prayers, and love.

My deepest gratitude and love go to Arda Kerbabian for her never-ending love and precious support. I also thank Elias Sahyouni, Adon Nouman, George Shammas and Asadour Mncherian for their friendship and support, and the whole NEST community for the friendly and loving environment.

I am grateful to the Department of Computer Science at the American University of Beirut for their educational environment and support, and for chairperson El Hajj for his advise. I am also thankful for my fellow graduate students for their friendship.

Last but not least, I thank God for his Grace, and for providing me with the strength, will, and wisdom to achieve my goals in life.

An Abstract of the Thesis of

John Abied Mitri Hatem for Master of Science
Major: Computer Science

Title: Predictive Resource Management using Deep Learning in Next Generation Passive Optical Networks

Over the last decade, Passive Optical Network (PON) has emerged as the best solution for the bottleneck problem in the *first-mile*, making it an ideal candidate for next-generation broadband access networks. Meanwhile, machine learning, and more specifically deep learning, has been regarded as a star technology for solving complex classification and prediction problems. Recent advances in hardware and cloud technologies offer all the necessary capabilities for employing deep learning to enhance PON's performance. In PON systems, to allocate bandwidth for the end-users, the Optical Line Terminal (OLT) polls the Optical Network Units (ONU) in a cyclic manner using control messages to enable Dynamic Bandwidth Allocation (DBA) in the upstream direction. In this thesis, we propose a novel DBA approach, thus-called Deep DBA, that employs deep learning to predict the bandwidth demand of end-users so that the overhead due to the request-grant mechanism in PON is reduced, thereby increasing the bandwidth utilization. More specifically, we employ a Long Short-Term Memory recurrent neural network that predicts the bandwidth demands of ONUs for several future cycles by peep-holing only a few previous cycles. Consequently, the OLT does not need to poll the ONUs during the predicted cycles, thereby reducing the overhead of control messages and idle times in the network. The gain achieved through Deep-DBA enables to provision more users and/or services on the same network while ensuring fairness among ONUs and supporting quality of service. Extensive simulations highlight the merits of the new DBA approach and offer insights for this new line of research. Results show that with Deep-DBA, the control message overhead and total overhead in the upstream direction are reduced by up to 70% compared to existing schemes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

According to Cisco's Visual Network Index forecast, the global Internet traffic, which was $26,600$ Gbps in year 2016, will reach a whopping $105,800$ Gbps by year 2021 [4]. The ZettaByte era has already begun with dramatically increasing Internet Protocol (IP) traffic, which is expected to reach 3.3 ZB per year, by year 2021. Consequently, Information and Communication Technology (ICT), which has a direct impact on most current and emerging technologies, has been already one of the fastest growing industries. Accommodating this unparalleled increase of Internet traffic demand is one of the biggest challenges that key players in this technology aim to address.

1.1 The *Bottleneck* Problem

The *first-mile* (also called the *last-mile* in some references) connects the service providers at the Central Offices (COs) to the different subscribers (e.g., businesses, residential, etc.), and is referred to as the "subscriber access network" or the "local loop". The network infrastructure at the first-mile mainly consists of the Digital Subscriber Line (DSL) and Cable Modem (CM) technologies. DSL uses the conventional twisted pair as medium, and the commonly used Asymmetric DSL (ADSL) technology can only provide few Mbps of bandwidth with a range of 5.5 km. Even the newer variations of DSL, i.e., very high-speed DSL (VDSL), provides only up to 50 Mbps for even a shorter range of 1.5 km. CM networks use coax cables as medium, and can provide only up to 40 Mbps for distances up to 25 km. This widely deployed network infrastructure at the first-mile reached its bandwidth limits and created what is known as the *first-mile bottleneck*. This is the result of the unprecedented growth in Internet traffic demands caused by the quick development of electronic devices (e.g., mobile devices, personal computers, smart TVs, smart homes, smart cities, etc.). These support an ever growing list of services (e.g., video conferencing, video-on-demand, Voice-over-IP (VoIP), interactive games, Tactile services, Internet of Things (IoT), etc.) combined with the advancements in the backbone network where systems with speeds of 1 Tbps and more, are easily being deployed [3, 5]. Table 1.1 presents the bandwidth requirements of the mostly used applications.

Table 1.1: Bandwidth requirements for different applications [3].

Application	Bandwidth	Latency
VoIP	64 kbps	200 ms
Video conferencing	2 Mbps	200 ms
File sharing	3 Mbps	1 s
SDTV 4.5	Mbps/ch	10 s
Interactive gaming	5 Mbps	200 ms
Telemedicine	8 Mbps	50 ms
Real-time video	10 Mbps	200 ms
Video on demand	10 Mbps/ch	10 s
HDTV	10 Mbps/ch	10 s
Network-hosted software	25 Mbps	200 ms

1.2 The Solution: Passive Optical Network

Over the last two decades, Passive Optical Networks (PONs) have been taken as the best solution to the access bottleneck problem in the first-mile and to deploy the FTTx (Fiber-To-The-x, where x stands for home, curb, building, office, etc.) technology. With PONs, the customer premises can be further apart from the service provider's central office with distances up to 100 km using Long-Range PONs (LR-PONs) [6]. Also, PONs provide much higher bandwidth ensuring the delivery of more than 1 Gbps speeds. Furthermore, PONs have passive components, such as optical fiber and splitters, instead of active ones, thereby relieving network operators from installing multiplexers and demultiplexers in addition to maintaining and provide power to them. This reduces the complexity of the network and is more cost effective. Moreover, PON equipment has low power consumption and can be easily upgraded to have additional wavelengths and/or

higher bit-rates [5].

Several variants of PON exist; however, Ethernet PON (EPON) has acquired considerable attention from the industry and the research community, due to its cost effectiveness, high bandwidth capacity, and ability to efficiently support Quality of Service (QoS) [5].

1.3 Thesis Motivation and Contributions

Compared to other access networks, PONs provide much higher bandwidth and are far more energy efficient. However, to embrace this explosive growth of demands, there is still much room for improvement. Data transmission in PON is subject to an overhead caused by several factors such as the deployed Dynamic Bandwidth Allocation (DBA) scheme, and the control messages exchanged between the different components of PON. This overhead limits the network's utilization and performance [7, 8].

Next Generation PONs (NG-PONs) are expected to support an increasing number of users, reach for longer distances (e.g., LR-PONs), provide much higher data rates, and meet the QoS requirements of various bandwidth-intensive and delay-stringed application such as mobile fronthaul, Tactile services [9], etc. Hence, it is expected that NG-PONs would increase in complexity to accommodate such various requirements; thus, "efficiently customized" PONs can become the new trend. That is, traditional "one-size-fits-all" approaches will not work,

and a certain setup of patterns which works fine for one network may prove to be inefficient for another network of the same kind [10].

To address the aforementioned challenges, we exploit the rapidly progressing field of Artificial Intelligence (AI), and more specifically Machine Learning (ML) as a promising tool to address some of the PON challenges. With ML, given a large enough set of training data, using machine learning, systems can learn from this data, via exploiting hidden and unexpected patterns by leveraging complex mathematical and statistical tools. This trained model can then perform automatized intellectual tasks and infer solutions. In recent years, ML applications, which proved to be efficient and successful, are invading numerous industries (e.g., healthcare, manufacturing, oil and gas, finance, etc.) in an accelerated rate. This success is due to the colossal amount of data available, which is bound to increase with new emerging networks like IoT where billions of devices are connected in smart cities. Furthermore, the recent advances in Graphics Processing Unit (GPU) technology and Cloud Computing provide the processing and storing capabilities needed for training the ML models. For inference, the trained model can be deployed in low capability devices, e.g., smartphones [10].

Even though some research has been done in the past few years focusing on the application of ML in optical networks in general [11, 12] and PONs in particular [13, 14, 15, 16, 17], this field is still in its baby stages [11]. For instance, ML can be used in network failure detection by learning from historical monitored network component information to detect and identify fault location and infer

specific failure types. This can be extended to build autonomic networks which are capable of self-configuring, self-healing, and self-optimizing. Also, using ML, systems can learn from collected historical Internet traffic to predict and classify traffic flows which helps network service providers to reduce over provisioning and guarantee best QoS. Furthermore, ML applications in PON can have significant impact in path computation, topology design, energy saving, and more. [10, 11].

Finally, the applicability of ML is further promoted by important advancements in networking technology. These include software-defined networks and software-defined optical networks [18], which enable network programmability, and AI-enabled optical network on chip. These promise to be a huge improvement over legacy electronic network on chip technology in terms of computation power and energy consumption [12, 10].

In this thesis, we study the causes of transmission overhead in PON, and propose a deep learning based PON DBA scheme to reduce the PON overhead, thereby increasing the network bandwidth utilization, while ensuring QoS. Hence, more users and/or services can be provisioned on the same network. One salient feature of the proposed DBA, is that it works with any current or future machine learning models and DBA algorithms.

1.4 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we present an overview of EPON and review Internet traffic prediction and the deep learning model used in our proposed scheme followed by up-to-date related work. In Chapter 3, we describe the proposed deep learning based PON system model. In Chapter 4 we present our simulation results and finally we conclude in Chapter 5 and discuss our future work.

Chapter 2

Background and Related Work

In this chapter, we provide a detailed overview of EPON, including the transmission overhead, and a review of the different DBA algorithms. We also discuss related works in Internet traffic prediction and predictive scheduling schemes in PON, and present an overview of Long Short-Term Memory (LSTM) recurrent neural network architecture, which is the model used for the proposed scheme.

2.1 Ethernet Passive Optical Network

EPON, like any other type of PON, is a point-to-multipoint optical network with no active elements on the transmission path between the source and destination. As illustrated in Fig 2.1, the most popular architecture of EPON is the tree topology, rooted by the Optical Line Terminal (OLT) located at the CO, and connects via a passive 1:N optical splitter a set of Optical Network Units (ONUs)

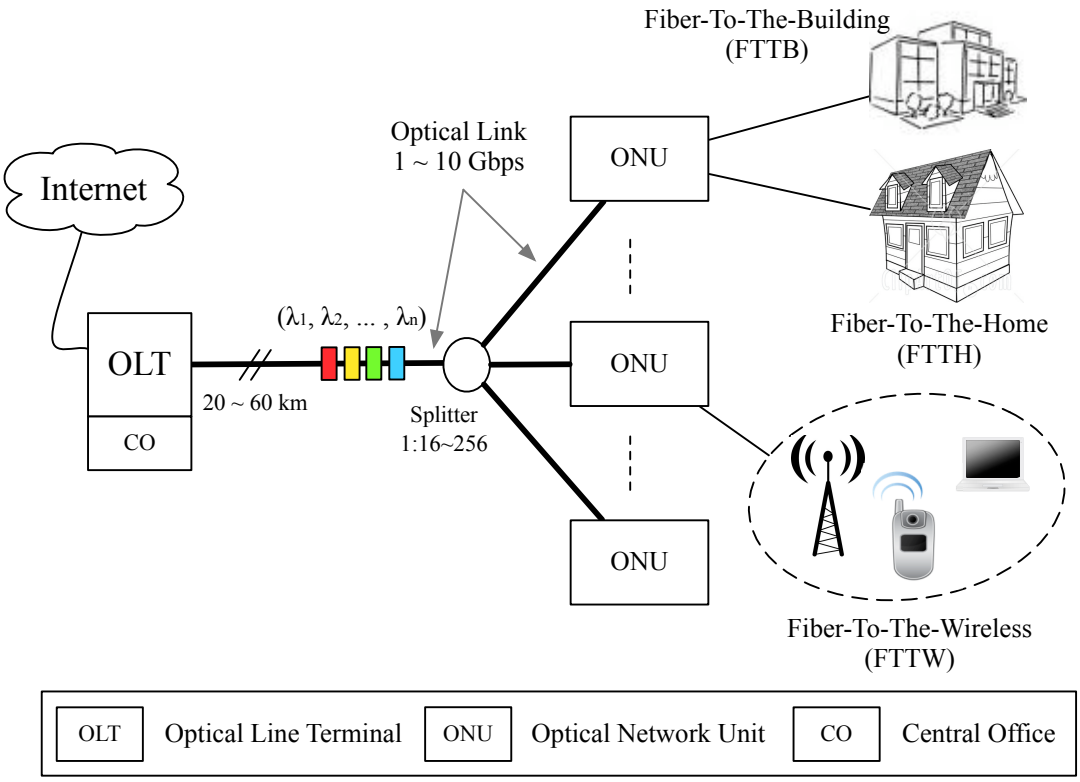


Figure 2.1: EPON architecture [1].

located at the end-users' premises, representing the leaves of the tree topology. The number of supported ONUs can be between 2 to 128. PON is connected to the backbone network, i.e., Metropolitan Area Network or Wide Area Network, via the OLT. The infrastructure that interconnects the OLT and the ONUs is defined as the Optical Distribution Network [3, 19].

The transmission in the downstream direction (i.e., from the OLT to the ONU) is performed by the OLT via broadcasting the data to the ONUs using a 1490 nm wavelength. Consequently, based on the Medium Access Control (MAC) address, each ONU will be able to identify the data that is destined to it. In the upstream direction (i.e., from the ONU to the OLT), ONUs can send data to the

OLT but not to each other due to the directional properties of the passive combiner. Hence, this setup is similar to a point-to-point architecture, even though the ONUs share the upstream transmission medium; thus, they are in the same collision domain. Consequently, data sent simultaneously by two or more ONUs will result in a collision, which is hard to detect [5]. Since it is difficult to implement Carrier Sense Multiple Access with Collision Detection, other solutions such as Wavelength Division Multiplexing (WDM) and Time Division Multiplexing (TDM) are employed. Using WDM, each ONU's upstream transmission would be on a different wavelength, at the expense of high network cost. TDM, on the other hand, is seen as a more attractive solution since it requires a single wavelength, which is highly cost-effective, at the expense of limited bandwidth capacity. With TDM, each ONU is granted a *non-overlapping* time slot for data transmission by the OLT. Hence, TDM is implemented on a single wavelength (1310 nm) for upstream transmission. The EPON standard does not specify a DBA scheme; instead, the decision is left to the network operator.

Today, 1G-EPON is widely deployed, and the deployment of 10G-EPON, which was standardized in 2009, has already begun in some countries [20]. Furthermore, in response to the high bandwidth demand, the IEEE P802.3ca task force has already spent efforts to standardize NG-PONs. To be more effective and to comply with the high deployment rates of legacy PONs, the task group has been working on standardizing three upcoming generations instead of one, namely the 25/50/100G-EPON, which are planned to be standardized by year

2019 [21, 20].

2.1.1 Multi-Point Control Protocol

In all EPON systems, an arbitrating mechanism is employed, so-called the Multi-Point Control Protocol (MPCP), which represents the signaling protocol that enables communication between the OLT and the ONUs. MPCP resides in the MAC control layer and operates in two modes relying on the use of six control messages which have a fixed size of 64B [3, 5, 22].

1. *Auto-discovery mode*: using REGISTER, REGISTER_REQ, REGISTER_ACK, and discovery_GATE control messages, MPCP ensures the discovery of newly connected ONUs in the network, collects relevant information like the MAC address and Round Trip Time (RTT), and registers these ONUs.

2. *Normal mode*: As illustrated in Fig. 2.2 using REPORT and GATE control messages, MPCP facilitates the medium access control and arbitrates the transmission of multiple ONUs over the shared upstream media. An ONU would send a REPORT message to the OLT, informing it about its bandwidth demands, and then the OLT, using a DBA algorithm, would respond with a GATE message granting the ONU a time slot that does not overlap with the transmission time slot of other ONUs. Hence, MPCP is independent from the used DBA strategies and schemes which are left to the vendors to specify.

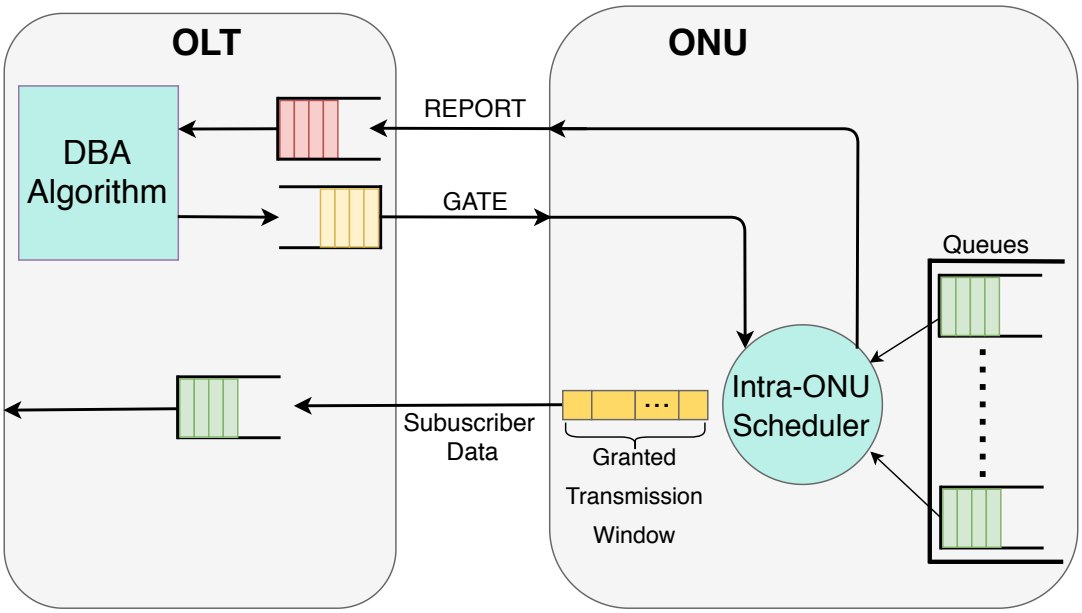


Figure 2.2: MPCP Operations [2].

As illustrated in Fig. 2.3, a GATE message can include up to 4 transmission window grants indicated by a mandatory 1B field. Each grant, indicated by a 6B field, consists of a grant start time (4B) and a grant length (2B). Furthermore, in addition to the requested data, the grant time slot should make room to include the REPORT message that the ONU sends to the OLT. A REPORT message can include 0 to 13 queue sets indicated in a 1B field. Each queue set can support 8 ONU upstream queues indicated by a 1B report bitmap, and the buffer occupancies, i.e., the bandwidth requirements, of each queue is reported in a 2B field. Hence, the number of queue sets depends on the number of queues reported in each set. Including multiple queue sets in the REPORT message helps avoid wasting bandwidth and can be used to prioritize traffic and support differentiated services [3].

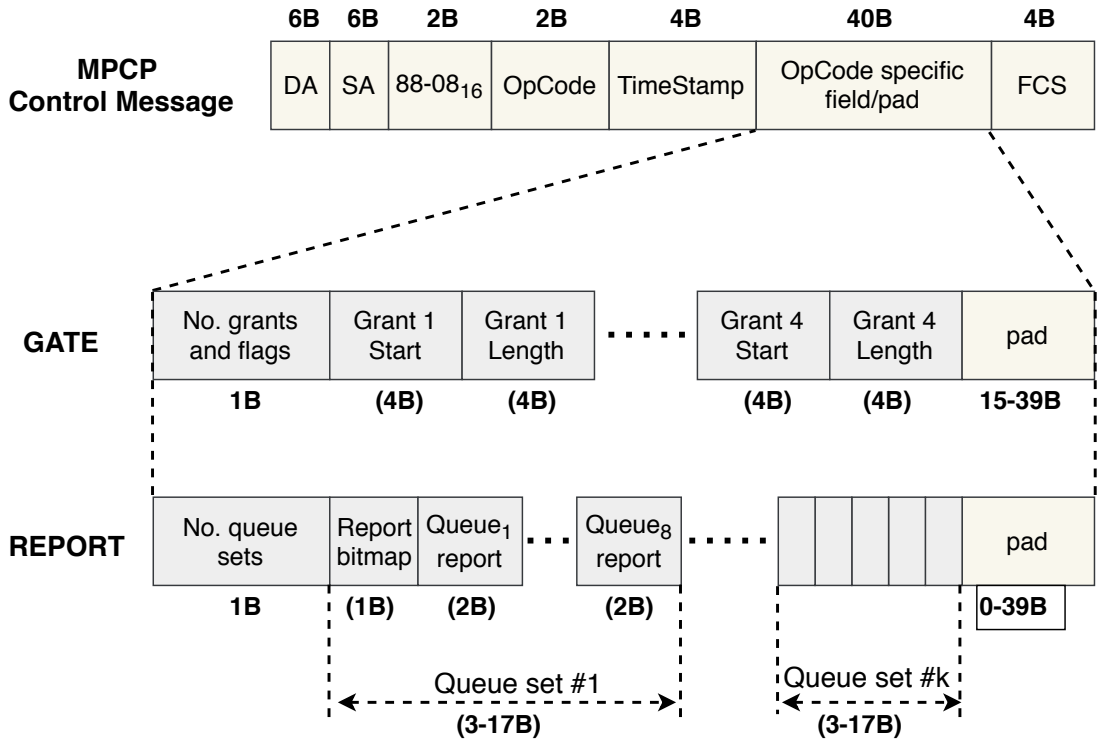


Figure 2.3: EPON GATE and REPORT message formats [3].

2.1.2 Polling Strategies

In every transmission cycle, the OLT requires the instantaneous bandwidth demands of each ONU in order to make allocation decisions. To this end, polling strategies are introduced to poll multiple ONUs.

Fig. 2.4a illustrates the *online* scheduling, introduced in [23], also referred to as "Interleaved Polling". With online scheduling, the OLT makes grant decisions on-the-fly based on individual ONU REPORT messages and sends the GATE message before the transmission end of previously polled ONU(s). This is feasible since the upstream and downstream transmissions are assigned different wavelengths. Note that, a short guard time between two successive ONU

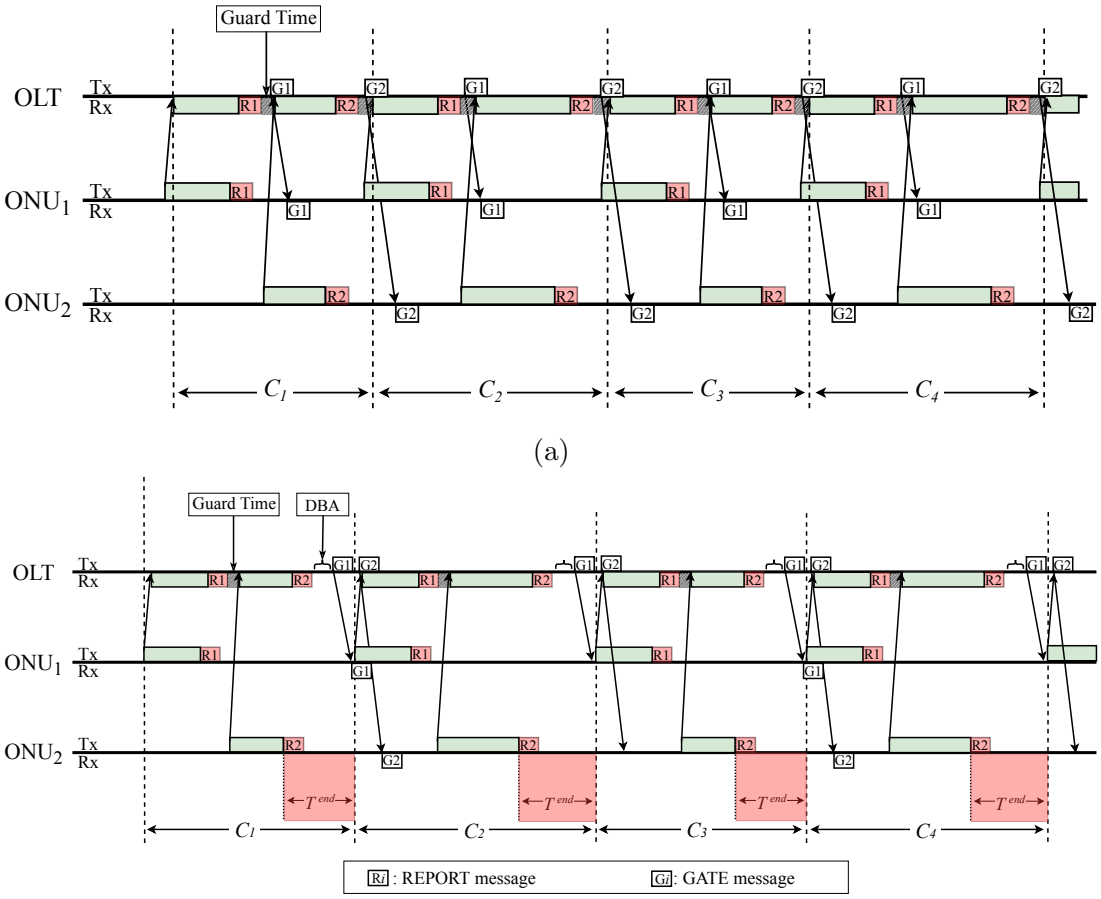


Figure 2.4: Polling Strategies: a) Online scheduling, b) Offline scheduling.

transmissions is required by the OLT's receiver. Results in [23] show that online scheduling significantly increases the bandwidth utilization and decreases the average delay. However with this strategy, the OLT would lack a holistic view of all ONU demands while making granting decisions. Hence, ensuring fairness among ONUs is very difficult, and supporting QoS is not easily attainable [19, 22, 2].

This problem is resolved by using *offline* scheduling. As illustrated in Fig. 2.4b, the OLT waits until REPORT messages from all ONUs are received and then performs bandwidth allocation at the end of the cycle with a complete

knowledge of the bandwidth demands of all ONUs. This allows the OLT to make more intelligent granting decisions ensuring fairness among ONUs and supporting QoS. However, this comes at the expense of upstream bandwidth utilization by forcing an idle period (also referred to as "walk time") at the end of the cycle, T^{end}. This idle period comprises the propagation delay of the REPORT message from the last polled ONU to the OLT, the DBA calculation time, and the time needed for the GATE message to be received by the first polled ONU [19, 22, 2].

2.1.3 Dynamic Bandwidth Allocation

Internet traffic in the access network is characterized by being bursty in nature. Even on low loads, a network traffic burst might cause the time slots for a certain ONU to overflow while the time slots for other ONUs are not fully utilized. Therefore, the bandwidth requirements of an ONU might vary widely even within short periods of time. This makes static bandwidth allocation a not-so-efficient solution [24]. On the other hand, in order to adapt to the instantaneous bandwidth demands of multiple ONUs, a DBA algorithm can be deployed at the OLT to provide statistical multiplexing among ONUs, and is shown to be more efficient [2]. Fig. 2.5 presents a taxonomy for DBA algorithms. As illustrated, DBA schemes can be divided into three main categories.

2.1.3.1 Grant Sizing

In the **fixed** grant sizing scheme, ONUs are granted a transmission window of a fixed size for every cycle. This can be represented by the function: $G_i = G_i^{max}$, where G_i is the grant size of ONU i and G_i^{max} is the maximum limit of a grant size. Result show that this scheme degrades network performance increasing bandwidth waste and average packet delay [23, 24].

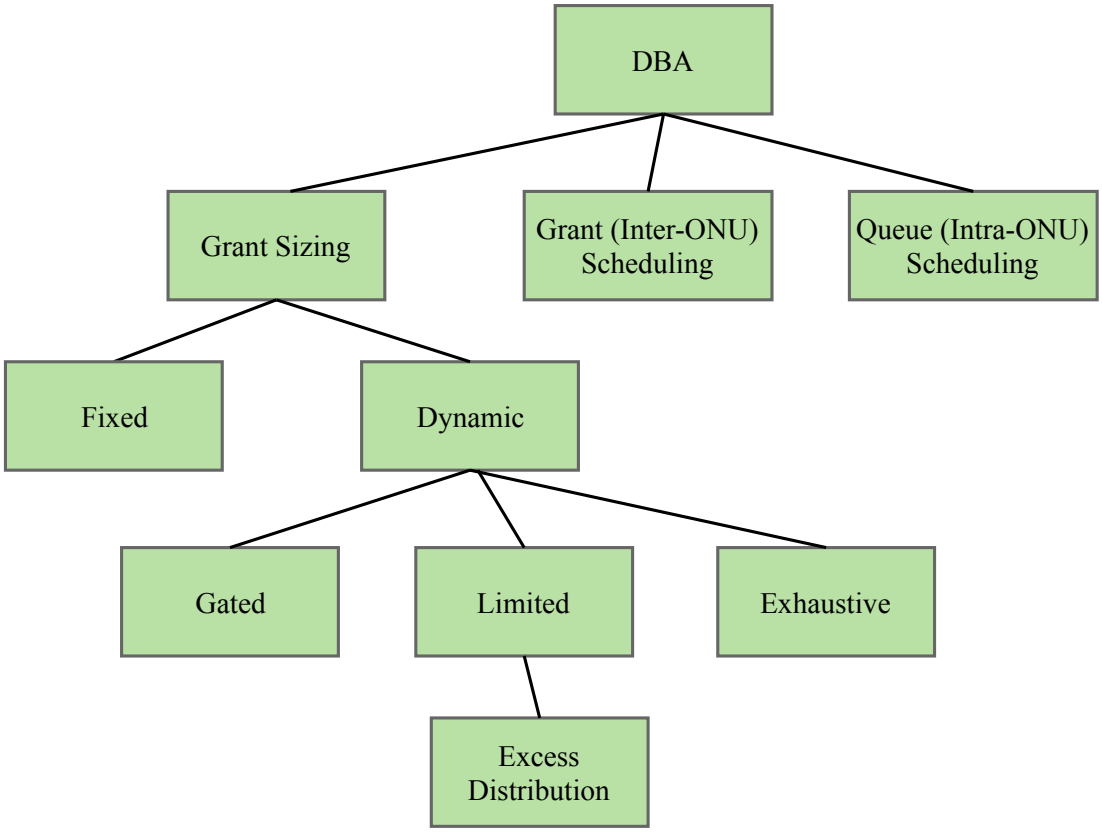


Figure 2.5: Dynamic Bandwidth Allocation Taxonomy [2].

In the **gated** grant sizing scheme, the size of the granted transmission slot for a given ONU is equal to the queue size reported by that ONU in the previous polling cycle. The function representing this scheme would be: $G_i = R_i$, where R_i is the

requested queue size included in the most recently received REPORT message. Simulation results show that the Gated scheme lowers the average packet delay in the network. However, the upstream channel might be monopolized by a single ONU for a long period of time. Therefore, the gated scheme fails to ensure fairness among ONUs [25].

In the **limited** grant sizing scheme, the size of the granted transmission slot for a given ONU is equal to the reported queue size, but limited by the maximum grant size for that ONU. Hence, the function for the limited scheme would be: $G_i = min(R_i, G_i^{max})$. Limiting the grant size to a maximum value G_i^{max} prevents upstream channel monopolization by a single ONU. Results show that the average packet delay is similar to that of the gated scheme [23]. Note that the network performance is greatly impacted by the value of the G_i^{max}. Setting the G_i^{max} to be very large will increase the delay for all packets; whereas setting G_i^{max} to be very small will increase the number of exchanged control overhead, thereby reducing the upstream channel utilization [22]. Moreover, since an Ethernet frame cannot be fragmented, the limited scheme might cause bandwidth waste in the case when $G_i^{max} < R_i$ and the head-of-line (HOL) Ethernet frame is larger than the remainder of the granted slot. Hence, the remained portion is wasted and the HOL frame is transmitted in the next transmission time slot [2].

The **limited with excess distribution** scheme is a variation of the limited scheme, in which the "excess" or unallocated bandwidth is exploited to improve the network performance. In this scheme, ONUs are divided into two groups:

underloaded and overloaded ONUs. An ONU is considered underloaded if $R_i \leq G_i^{max}$ whereas an ONU is considered overloaded if $R_i > G_i^{max}$. In this scheme, when all REPORT messages are received, the total excess bandwidth is calculated as $E_{total} = \sum_{i \in u}(G_i^{max} - R_i)$, where u is the set of underloaded ONUs. Afterwards, different schemes can be used to divide E_{total} between all overloaded ONUs.

In the **exhaustive** grant sizing scheme, prediction techniques are used to estimate the amount of accumulated bandwidth during the time between sending of the REPORT message and the start of the granted transmission window. Predicting the ONU's queue size at the time of transmission can lower the queuing delays. However, predictions might not be very accurate due to the bursty nature of Internet traffic, which may downgrade the network performance. Prediction schemes are further explored in Section 2.2.2.

2.1.3.2 Grant (Inter-ONU) Scheduling

Inter-ONU scheduling is concerned with scheduling grants in the upstream channel to the ONUs and is performed at the OLT. The simplest way to perform Inter-ONU scheduling is in a round robin manner. However, in order to change the grant order, the OLT must wait until it receives all the REPORT messages from all ONUs. This requires using the offline scheduling scheme. Several Inter-ONU scheduling schemes were proposed in the literature such as the Longest Queue First (LQF) and the Earliest Packet First (EPF). In LQF, the ONUs with the largest transmission slots are set to transmit their data first. In EPF,

the ONUs are ordered to transmit based on the time of the arrival of the HOL Ethernet frame [2].

2.1.3.3 Queue (Intra-ONU) Scheduling

Scheduling the transmission of the different queues at an ONU within the granted transmission window is controlled by Intra-ONU scheduling. Packets arriving from the registered subscribers to the ONU are first classified by means of a packet-based classifier. Next, before placing packets in the queues, the ONU decides if a given packet should be admitted depending on the traffic policing mechanism. Typically, there are two types of Intra-ONU scheduling: strict and non-strict priority scheduling. In the first scheme, low-priority queues are scheduled only if higher priority queues have no traffic left for transmission. This causes starvation for low-priority traffic. Non-strict priority scheduling algorithms tackle the starvation problem by allowing all queues in the ONU to have access to the upstream channel while respecting their priority [2, 26].

2.1.4 EPON Efficiency

Network throughput is the amount of application-level user data the network carries through within a unit of time. Network throughput efficiency, which is also called network *utilization*, is the ratio of the maximum measured throughput to the network bit rate [7].

EPON systems suffer from transmission overhead which lowers the utiliza-

tion. This overhead has two main components associated with encapsulation and scheduling [7]. The overhead caused by scheduling consists of control message overhead, guard time overhead, discovery overhead, etc., and is influenced by several factors such as the maximum polling cycle time, the number of connected ONUs, the distance between the OLT and ONUs, and the employed DBA.

For example, assuming there is one GATE message and one REPORT message between the OLT and each ONU in every cycle, we can calculate the control messages overhead as follows [7, 8]:

$$O_{Control} = \frac{S \times N}{T}. \tag{2.1}$$

where S is the size of the control message including the Ethernet preamble and inter-packet gap (84 bytes), T is the maximum cycle length (in seconds), and N is the number of ONUs. Thus, for an EPON system of 128 ONUs and 1ms cycle time, in the upstream direction, REPORT messages would cause about 86 Mbps of overhead. Furthermore, as illustrated in Section 2.1.2, there is a trade-off between using online and offline schedulers. Offline schedulers support QoS and enable fairness among ONUs at the expense of decreased channel utilization due to forcing an idle period at the end of a polling cycle T^{end}. With online schedulers, T^{end} is eliminated, but ensuring fairness among ONUs and supporting QoS is not easily attainable as the OLT would lack a holistic view of all the ONU demands.

Finally, an extra idle time may occur if the GATE message arrival time to

ONU i is later than the transmission start time of that ONU; in this case, the transmission start time will be delayed until the GATE message is received by the ONU. This occurs at light network loads where the ONUs would typically have negligible bandwidth requests and the guard time between successive transmissions is less than the propagation delay of the GATE message.

Our proposed deep learning based PON system model addresses the control message and idle time overheads and provides a balance between offline and online scheduling schemes by increasing the bandwidth utilization in the upstream direction while maintaining the QoS support and fairness among ONUs.

2.2 Internet Traffic Prediction

Internet traffic is considered a time series which is a sequence of observations $(x_1, x_2, ..., x_t)$ recorded at a specific time t [27]. Moreover, Internet traffic has the characteristics of self-similarity, long-range dependence, and highly non-linear nature [28, 29]. To predict Internet traffic, several linear prediction techniques, such as AutoRegressive Moving Average, Autoregressive Integrated Moving Average, AutoRegressive AutoRegressive and HoltWinter algorithm, were proposed [30, 31, 32]. These methods can only learn the linear correlation structure present in the time series, but they do not learn the non-linear patterns. For non-linear prediction, some of the models used were Radial Basis Function and Support Vector Machine, which had lower error rates [33].

Recently, based on the non-linearity of Internet traffic, Neural Networks (NNs) have been employed for traffic prediction. NNs are widely used since they can approximate any linear or non-linear patterns in an accurate manner even though the underlying data relationships are unknown. In [27], the authors have shown that by using neural networks, better prediction outcome in terms of accuracy and response time can be obtained.

In [34], the authors compare two different NN approaches. The first one is a simple Multi-Layer Perceptron (MLP) and the second one is a deep learning Stacked AutoEncoder (SAE). The MLP has one input layer, two hidden layers, and one output layer with a feed forward flow. Furthermore, the sigmoid activation function was used. On the other hand, for the SAE model, a deep learning neural network was constructed with multiple layers of AutoEncoders. Here, the output of each layer is connected to the input of the next layer. The activation function used for the SAE model was also the sigmoid function. Furthermore, for both models, before the training, the data is normalized and the loss function used is the Root Mean Squared Error (RMSE). The reported results show that both models are capable of having accurate predictions. However, the MLP model outperforms the SAE model, even when more layers are added to the SAE model. In addition, the SAE model is more computationally costly during training than the MLP model. Hence, the authors conclude that the MLP model is more advantageous.

In [33], the authors present three architectures using Deep Belief Networks.

The NN is created with an input layer, four hidden layers, and an output layer of one neuron representing the prediction Internet traffic in time t + 1. In the first model, the four hidden layers are of sizes 300, 200, 100, and 10, respectively. In the second model, the hidden layers are of sizes 300, 300, 300, and 300, respectively. In the third model, the hidden layers are of sizes 300, 200, 10, and 3, respectively. The sigmoid activation function is used, and the performance measure of each of the three models was calculated in terms of Mean Squared Error (MSE) and RMSE. Results show that the first model outperforms the other two models by having more accurate predictions.

Nevertheless, traditional Feed Forward Neural Network (FFNN) are not able to handle historical data and are only limited to modeling the data within fixed-size window. In contrast, RNNs take into consideration past-seen input, by containing cycles that carry the activations from previous time steps back to the network, to make predictions in the current time step. This makes RNNs a good candidate for *sequence-to-sequence* predictions. However, conventional RNN models suffer from vanishing and/or exploding gradient problems, which limits the RNN's capability to model long-term dependencies starting from 5 discrete time steps between relevant input and output signals [35]. Consequently, an elegant RNN architecture called LSTM has been designed to address these foregoing issues [36], and it was demonstrated to outperform the traditional RNN for many applications [35, 37, 38, 39]. In Section 2.2.1, we review the architecture of the LSTM recurrent neural network and recent works that used LSTM in In-

ternet traffic prediction. In Section 2.2.2, we discuss related work in predictive scheduling in PON.

2.2.1 Long Short-Term Memory

The architecture of LSTM is composed of memory blocks. As illustrated in Fig. 2.6, each block contains memory cells with self connections and multiplicative units called gates. An LSTM network takes as input a sequence $x = \{x_1, x_2, ..., x_T\}$, and outputs a sequence $h = \{h_1, h_2, ..., h_T\}$ by using the following equations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.2}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.3}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{2.4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.5}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{2.6}$$

$$h_t = o_t \odot tanh(C_t) \tag{2.7}$$

where W denotes the weight matrices, b denotes the bias vectors, and the terms f_t, i_t, and o_t are denoted as the forget gate, input gate, output gate at time t respectively. These gates control the information flow in the block. Finally, h_t represents the hidden state and C_t represents the cell state of the memory cell at

time t.

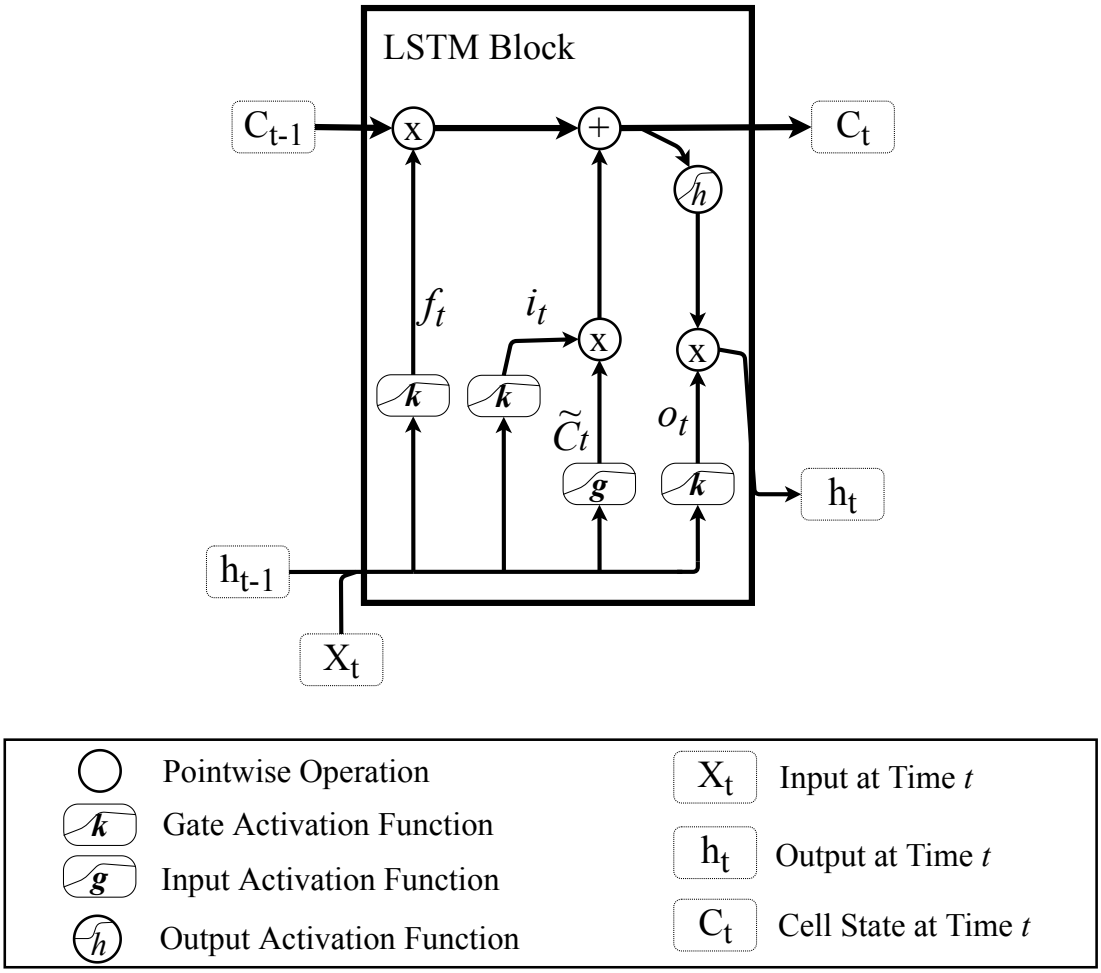


Figure 2.6: LSTM Memory Block.

Recently, several works have shown the effectiveness of LSTMs in predicting Internet traffic. In [37], a deep LSTM architecture was used to learn the network traffic and predict the future traffic matrix of a network. Different number of hidden nodes (200, 300, ..., 700) is used, and a different number of hidden layers (between 1 and 6) is employed. Here, MSE is used to estimate the prediction accuracy. Results show that the LSTM model significantly outperforms traditional

linear methods and FFNNs. Furthermore, it was shown that the MSE drops as the number of hidden layers and the number of hidden nodes increase. A similar study in conducted in [29] with similar results.

In [38], the authors evaluate the performance of different Recurrent Neural Network (RNN) architectures in predicting network traffic. Experiments are performed on FFNN, RNN, Gated Recurrent Unit, Identity RNN, and LSTM with different number of layers and hidden units. Results show that LSTM has the best performance compared to other architectures.

2.2.2 Predictive Scheduling in PON

Given the scope of our work, we focus on predictive schemes in PON. For instance, in [40], Kramer et al. propose a scheme so-called Constant-Bit-Rate Credit used with the Interleaved Polling with Adaptive Cycle Time (IPACT) DBA to reduce the queuing delay of low-priority packets in intra-ONU scheduling. Here, given the nature of Constant Bit Rate (CBR) traffic, prediction with some accuracy on how many high priority packets will arrive at the ONU, was performed. However, this method would not be accurate for bursty Internet traffic.

In [41], for less delays and less data loss, the authors propose a Limited Sharing with Traffic Prediction (LSTP) algorithm. In LSTP, before sending a REPORT message, the ONU uses the amount of data arrived in previous cycles

to predict/estimate the amount of data that will arrive during the waiting time based on the self similarity of Internet traffic [28]. The prediction is done at the ONU side, and the predicted amount is then added to the bandwidth request sent to the OLT.

To improve bandwidth utilization and QoS, the authors in [42] propose an Early DBA scheme with excessive bandwidth allocation. An unstable degree list of ONUs is defined based on the variance of historical traffic of each ONU, and prediction is done for different classes of service (i.e., Expedited Forwarding (EF), Assured Forwarding (AF), Best Effort (BE)). Similarly, the authors in [43], propose a class-based traffic prediction scheme at the ONU using a simple linear predictor.

In [44], Hwang et al. proposed a generic QoS-aware interleaved DBA. The cycle is divided into two equally-sized subcycles, each containing half of the ONUs. Here, one group of ONUs would be sending data in the upstream direction; meanwhile, the OLT would be calculating the grant times for the second group of ONUs. Furthermore, the excessive bandwidth allocation would be combined with a prediction method that supports differentiated traffic characteristics (i.e., EF, AF, BE). The predicted EF traffic is calculated by multiplying the past EF traffic request by the waiting time, divided by the cycle time. The AF and BE traffic predictions are calculated by comparing the difference between the current request and the mean value of the requests in the past 10 cycles.

In [45], an adaptive DBA algorithm that supports multi-services over EPON

is proposed, such that prediction is used only for EF traffic since it is assumed to be the most sensitive for delay.

The authors in [46] propose the IPACT with Grant Estimation (IPACT-GE) scheme. After receiving a GATE message, the ONU calculates the traffic arrival rate and uses the obtained value to estimate the amount of packets arriving in the next polling cycle. This estimation is added to the request size and the total amount is then sent to the OLT via a REPORT message.

To properly react to real time changes in the Service Level Agreements (SLAs) in PONs, the authors in [47] presented a SLA Proportional-Integral-Derivative (SPID) controller to provide better QoS to users. The SPID is enhanced with an online neural network to tune the parameters of the SPID controller. The input layer of the NN has three neurons representing the three past error readings of the SPID. Similarly, the output layer has three neurons representing the three parameters used to tune the SPID controller. Since the NN is online, it will learn from previous values in real time and will modify its weights accordingly.

To optimize upstream bandwidth allocation in PONs, the authors in [48] proposed to dynamically (re)allocate SLA parameters, which are represented by the Committed Information Rate (CIR), which is the guaranteed bandwidth provided to the user, the Excess Information Rate (EIR), which is an additional bandwidth that may be provided to the user, and the Peak Information Rate (PIR), which is the maximum bandwidth that can be assigned to a user, based on the user profile. Namely, using K-means clustering, users are classified into three different groups:

heavy, *light*, and *flexible* for specific periods of the day. Subsequently, excess bandwidth is allocated to the EIR of heavy users to improve their QoS. The limitation of this scheme is that the majority of users were classified as *flexible*. This work is then extended in [13], such that user groups are further classified based on the bandwidth usage during weekdays and weekends. Furthermore, a "Grey Forecasting Model" is employed to predict the future bandwidth demand trend of users in the flexible group, that is, whether they will shift or not to the heavy or light groups to have a more balanced distribution of the excess bandwidth.

To predict the additional packets that may arrive during the polling period, the authors of [14] proposed a data mining forecasting DBA, so-called DAMA, which employs an enhanced k-nearest neighbor (k-NN) algorithm. Results show that predicting the additional bandwidth improves the network performance in terms of latency and jitter.

In [15], the authors proposed an Artificial Neural Network (ANN) decision-making model to predict the bandwidth demand of an ONU. The ANN model is trained to predict uplink latency under different network scenarios so as to dynamically allocate bandwidth to meet low latency requirements.

To support Tactile services, the authors of [16] employed a Bayesian estimation to approximate the packet inter-arrival time for Poisson-distributed Tactile traffic in a WDM-PON. For Pareto-distributed traffic, the authors used a maximum-likelihood sequence estimation to approximate the *On* and *Off* durations. The estimations are performed at the ONU, and are then sent to the

OLT using a REPORT message. Consequently, the OLT evaluates the average bandwidth demand of each ONU and maintains the low latency constraint by dynamically varying the number of active wavelength channels.

Finally, the authors of [17] proposed a ML based Predictive DBA (MLP-DBA), where an ANN model is deployed at the OLT to identify the *On* and *Off* periods of bursty Internet traffic for the next polling cycle of every ONU. Based on this prediction, the bandwidth demand during the waiting time is evaluated. Consequently, if the sum of the requested bandwidth plus the predicted bandwidth is greater than the maximum bandwidth allowed for each ONU, an extra cycle is introduced by generating additional GATE messages for these ONUs at the beginning of the next polling cycle. This would offer lower latency and enable the support of Tactile services.

Chapter 3

Deep Learning-Based DBA

3.1 System Model

In legacy PON systems, in every polling cycle, each ONU sends to the OLT a REPORT message depicting its buffering queues' occupancies, which reflects the end-users' bandwidth demands. Consequently, the ONUs are granted time slots by the OLT in the next cycle using GATE messages; these time slots are sized depending on the DBA discipline.

As illustrated in Fig. 3.1, to reduce the overhead discussed in Section 2.1.4 (which can be significant depending on the DBA scheme, the polling strategy, i.e., online or offline, the number of connected ONUs, the distance between the OLT and ONUs, and the channel speed), we propose to employ a machine learning model at the OLT to predict the bandwidth demand of an ONU for the next Q cycles based on its demands in the past P cycles.

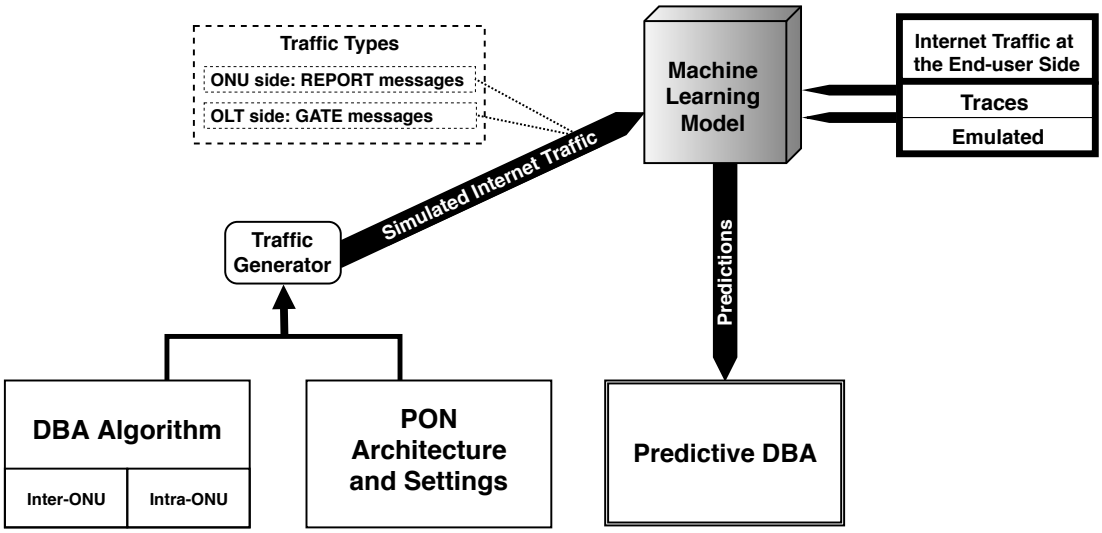


Figure 3.1: Proposed machine-learning based system model.

Here, bandwidth demands can be collected in three forms: 1) incoming traffic flows/streams at the end-user side; 2) REPORT messages in every polling cycle, which depict the bandwidth demands; and 3) GATE messages issued by the OLT in every polling cycle, which indirectly reflect the bandwidth demands of the ONUs. The latter two depend on the employed DBA algorithm and the network architecture and settings, as these affect the behavior of the network and thus the bandwidth included in the REPORT and GATE messages. Consequently, the machine learning model is trained on the collected data, and the obtained model is saved and embedded as a module in the DBA so as to perform predictive bandwidth allocation.

3.2 Bandwidth Demand Prediction using Deep Learning

Internet traffic in PON can be seen as time series, which corresponds to the bandwidth demand in every cycle. Hence, any machine learning model, which can handle *sequence-to-sequence* time series predictions can be used to perform predictive DBA. In this work, based on similar findings in related problems [49, 38, 35, 29] and extensive experimental results on our dataset, we choose to employ an RNN LSTM model.

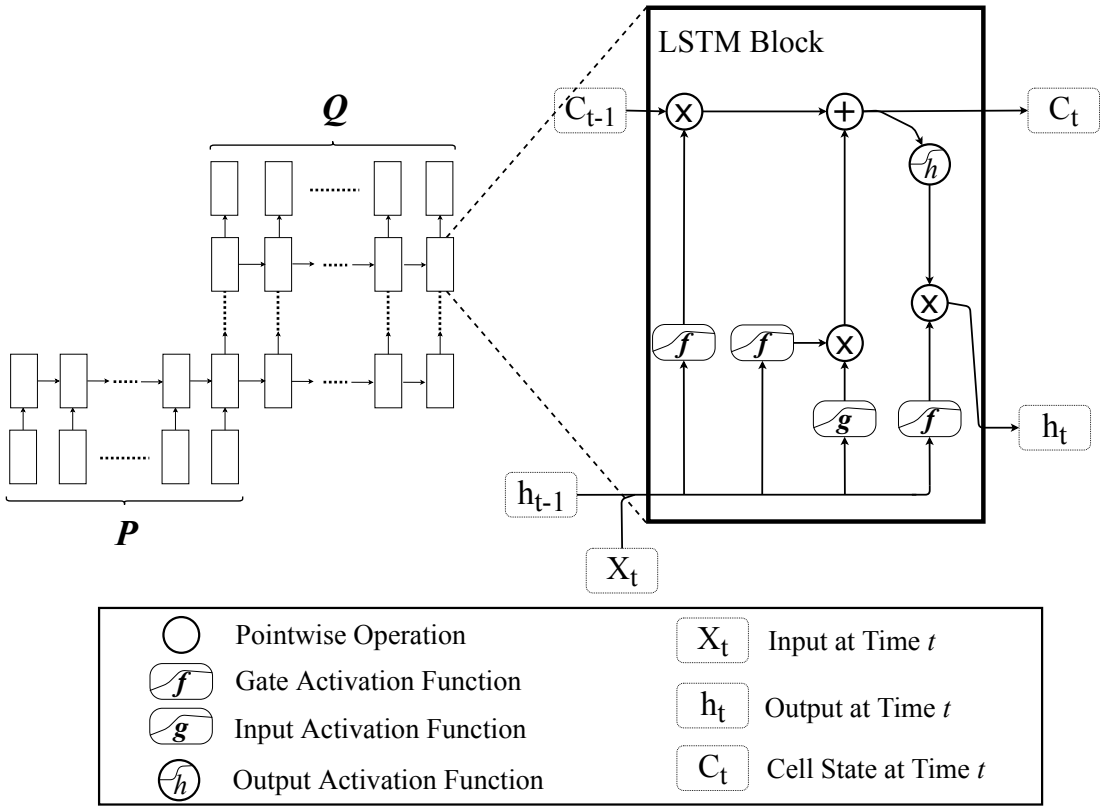


Figure 3.2: Employed LSTM RNN model.

Fig. 3.2 depicts the general LSTM architecture that we employ to predict future sequences from previous ones. As illustrated, the LSTM network is fed a sequence of P cycles as input, such that each cycle $p \in P$ includes the ONU's reported queue size. The output of the model would be the next sequence of Q cycles, where each cycle $q \in Q$ includes the predicted total queue size for this ONU in this cycle. This is an improvement over previous works since not only we are predicting the traffic in the next time step $t+1$ (next cycle), but we are also predicting the bandwidth demands for the upcoming $t+Q$ time steps.

3.3 Operation of Deep-DBA

The key principle behind Deep-DBA is to make use of the predictions made by a machine learning model using the past P REPORTs, so as to allocate bandwidth for the next Q cycles without requiring any further REPORT messages within those cycles. Thus, a Deep-DBA cycle would typically comprise two sets of cycles; namely the *reporting* cycles $\{p_1, p_2, \cdots, p_P\}$ and the *prediction* cycles $\{q_1, q_2, \cdots, q_Q\}$.

For simplicity and without loss of generality, we illustrate in Fig. 3.3 the operation of the proposed Deep-DBA for $P = 2$, and $Q = 8$. As observed, during the reporting cycles, every ONU sends a REPORT message requesting bandwidth based on its queue size (just like with regular DBA approaches). Consequently, the OLT runs the DBA algorithm (i.e., T_{DBA}) and responds with a GATE message

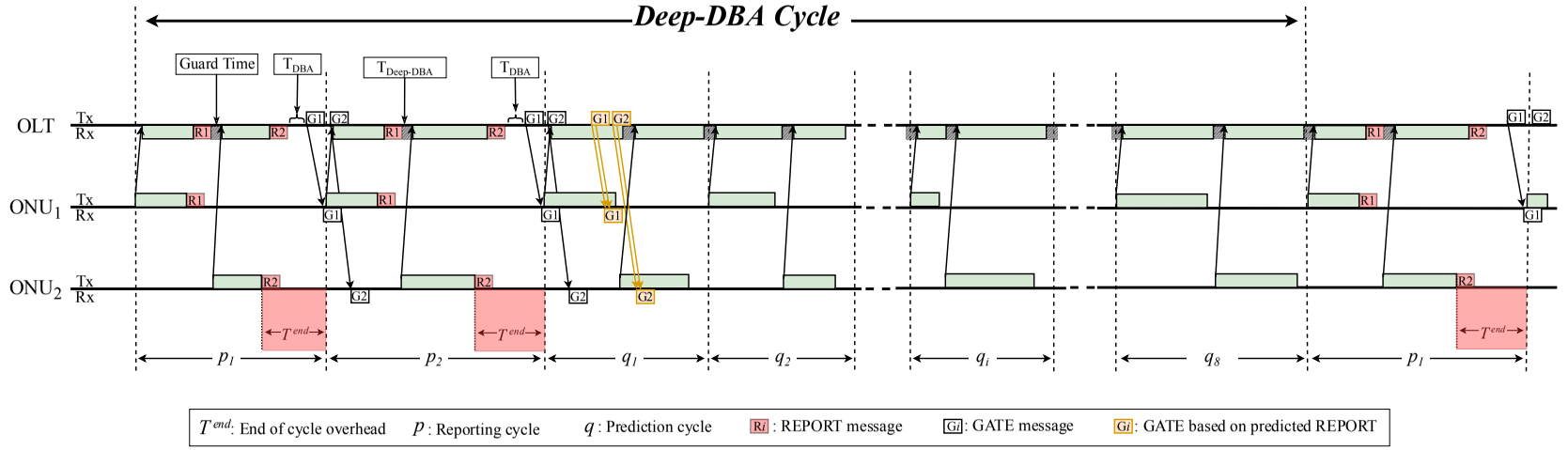


Figure 3.3: Operation of the proposed Deep-DBA scheme.

that includes a grant for the next cycle **only**, based on the employed scheduling discipline (i.e., Limited, Gated, etc.). However, the OLT keeps record of the foregoing request to be used for prediction. As such, during the last reporting cycle $p_{_P}$, as soon as the OLT receives the P^{th} REPORT message from an ONU, it uses the P saved requests of this ONU as input to the deep learning model, so as to predict its request sizes for the next Q cycles; thereby marking the start of DBA prediction time $T_{Deep-DBA}$.

When predictions are obtained by the deep learning model and the OLT has the predicted request sizes for all ONUs, these predicted request sizes are considered as if they are REPORT messages received from the ONUs for cycles $\{q_{_1}, q_{_2}, \cdots, q_{_Q}\}$. After the prediction, the OLT will apply the same DBA scheme used to grant transmission windows for each ONU for cycles $\{q_{_2}, \cdots, q_{_Q}, p_{_1}\}$ without requiring any further REPORT messages. This reduces the effective cycle time, and increases the network utilization. Subsequently, the OLT informs the ONUs of their transmission windows for cycles $\{q_{_2}, \cdots, q_{_Q}, p_{_1}\}$ using GATE messages sent during the first prediction cycle $q_{_1}$. This can be accomplished in three different ways:

1. The OLT sends $Q \times N$ GATE messages in a contiguous manner, where N is the number of ONUs.

2. The OLT incorporates 4 grants in one GATE message (which adheres to the default GATE message structure), so that $\frac{Q \times N}{4}$ GATEs are sent in a

contiguous manner.

3. The format of the GATE message is modified so that it can include Q grants, which enables the OLT to send N GATEs in a contiguous manner.

As observed in Fig. 3.3, the GATE message based on the predicted REPORT for the first ONU to start upstream transmission should arrive before the start of the second prediction cycle q_2. Hence, there is sufficient time to do the machine learning prediction which starts in the beginning of the last reporting cycle p_P, when the REPORT of the 1^{st} ONU arrives, and continues through the first prediction cycle q_1. Therefore, the time from the start of $T_{Deep-DBA}$ until the latest moment for the first GATE to reach its corresponding ONU is almost equal to the duration of 2 cycles, which is more than sufficient given the instantaneous output that is normally produced by a trained deep learning model [10].

In the next Deep-DBA cycle, during the first reporting cycle p_1, the ONUs start data transmission immediately; however, they also send REPORT messages at the end of their transmission window marking the start of the reporting cycles. The total number of cycles in one Deep-DBA cycle K, can be obtained as:

$$K = P + Q. \tag{3.1}$$

To highlight the merits of the offline Deep-DBA, we calculate its gain compared to regular offline and online DBA schemes. For the reader's convenience,

we summarize the notations used in Table 3.1.

Table 3.1: Summary of Notations.

Notation	Description
N	The number of ONUs
$T_{G_i^j}$ / $T_{R_i^j}$	Total delay for GATE / REPORT of ONU j in cycle i
T_G^{proc} / T_R^{proc}	GATE / REPORT processing time
T_G^{trans} / T_R^{trans}	GATE / REPORT transmission time
T_j^{prop}	Propagation time of ONU j's packet
T_{DBA} / $T_{Deep\text{-}DBA}$	DBA computation/prediction time
T_i^{end}	End-of-cycle overhead in cycle i
R_i	REPORTs overhead in cycle i
$O_{Deep\text{-}DBA}$	Total overhead using Deep-DBA
O_{REG}	Total overhead using a regular PON

The total delay caused by a GATE message destined for ONU j in cycle i equals to 2 processing delays, one at the OLT and the other at the ONU, in addition to the transmission and propagation delays.

$$T_{G_i^j} = (2 \times T_G^{proc}) + T_G^{trans} + T_j^{prop}. \tag{3.2}$$

Similarly, the total delay caused by a REPORT message from ONU j in cycle i is computed as follows:

$$T_{R_i^j} = (2 \times T_R^{proc}) + T_R^{trans} + T_j^{prop}. \tag{3.3}$$

As illustrated in Fig. 3.3, the reporting cycles with Deep-DBA are similar to the offline cycles of a legacy DBA (that is, an ONU sends a request message in

cycle $i-1$, and receives a grant from the OLT for cycle i). However with Deep-DBA, two salient overhead periods that occur in these cycles are eliminated in the prediction cycles, namely the end of the cycle idle time, T_i^{end}, and the REPORT messages transmission time. The overhead T_i^{end} comprises the time taken by the N^{th} REPORT message to be transmitted and processed, the time taken to compute the DBA, T_{DBA}, and the time taken by the first GATE message to be transmitted and processed. Thus, T_i^{end} can be computed as follows:

$$T_i^{end} = \begin{cases} T_{R_i^N} + T_{DBA} + T_{G_i^1} & i \in \{p_1, p_2, ..., p_P\} \\ 0 & i \in \{q_1, q_2, ..., q_Q\} \end{cases} \tag{3.4}$$

As such, the total overhead caused by REPORT messages during a Deep-DBA cycle i, R_i, would be obtained by:

$$R_i = \begin{cases} (N-1) \times T_R^{trans} & i \in \{p_1, p_2, ..., p_P\} \\ 0 & i \in \{q_1, q_2, ..., q_Q\} \end{cases} \tag{3.5}$$

Here, the transmission delay of the N^{th} ONU is accounted for in (3.4). Thus, the control overhead would only be incurred in the reporting cycles. Therefore, the total overhead using Deep-DBA can be computed as follows:

$$O_{Deep\text{-}DBA} = \sum_{i=1}^{P} (T_i^{end} + R_i). \tag{3.6}$$

Conversely, in a regular PON model, the total overhead using offline schedul-

ing would be computed as follows:

$$O_{REG} = \sum_{i=1}^{K} (T_i^{end} + R_i). \tag{3.7}$$

Consequently, the total gain G obtained via Deep-DBA can be estimated as follows:

$$G = O_{REG} - O_{Deep\text{-}DBA} = \sum_{i=1}^{Q} (T_i^{end} + R_i). \tag{3.8}$$

Chapter 4

Performance Evaluation

To validate the effectiveness of the proposed scheme, we generate training data and conduct extensive simulations using OMNET++ [50]. The simulation parameters, as in [51], are shown in Table 4.1. The 95% confidence interval of the simulation results gave $\approx 2\%$ variation, which is statistically insignificant; hence it is not shown in the figures.

Table 4.1: Simulation parameters.

Number of ONUs	16
Channel speed	1 *Gbps*
Link speed between ONU and user	65 *Mbps*
Distance from OLT to ONU	20 *km*
Guard time	1 μs
Processing time (T_R^{proc} and T_G^{proc})	10 *ns*
Maximum cycle time	2 *ms*
ONU buffer size	10 *MB*
$T_{DBA}, T_{Deep\text{-}DBA}$	≈ 0 (negligible)

4.1 Dataset

Without loss of generality, to validate the feasibility of the proposed Deep-DBA scheme, we generate traffic for the Gated and Limited scheduling disciplines, which are the most widely used legacy disciplines for predictive DBA schemes [2]. Namely, we implement a traffic generator at each ONU, which generates Poisson-distributed and Pareto-distributed traffics; the latter has 2000 alternating (i.e., ON/OFF periods) sources to emulate the long-range dependence and self-similarity of bursty Internet traffic [28]. Furthermore, different from previous works [17], we only make use of the request sizes, which are already included in the REPORT messages sent from the ONUs to the OLT, and no extra features are used to train the LSTM model so as not to add additional information in the REPORT messages. This also experimentally proved to be sufficient for predicting requested bandwidth without any added value for extra features.

Overall, our dataset consists of 8 million REPORT messages collected at all network loads, which is large enough to build robust LSTM models that generalize well. For different P-to-Q values, different datasets are prepared as shown in Fig. 4.1 to train, validate and test the corresponding P-to-Q LSTM model. For example, if a 2-to-2 model is employed, two REPORT messages are used as input, and the next two REPORT messages are used as output, and so on. As is custom in such settings, 80% of the dataset is used for training, 10% for validation, and 10% for testing.

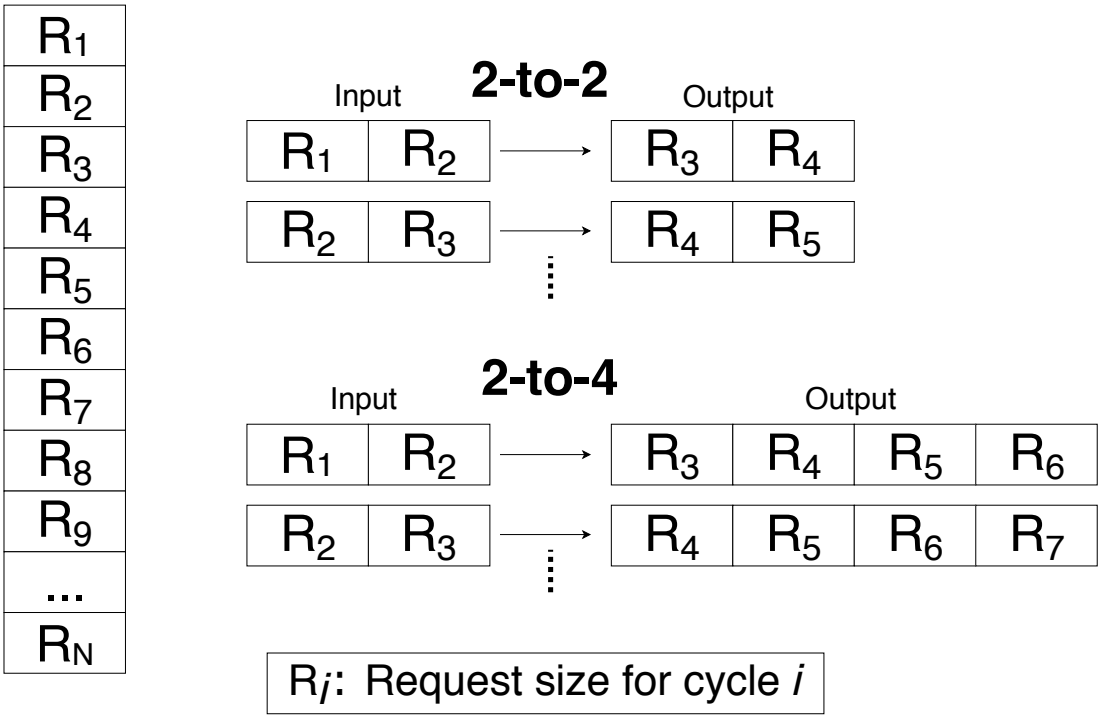


Figure 4.1: Dataset preparation.

4.2 Training the LSTM Model

To train the LSTM model, the dataset is normalized by dividing each request value by the maximum queue size. As a loss-function, we use the MSE between the predicted queue sizes and the actual queue sizes. The optimizer used to train our models is "AdaGrad" with learning rate 0.01. For different values of P and Q, a defined DBA scheme (i.e., Gated, Limited, etc.), PON architecture, and traffic distribution, the hyper-parameters of the LSTM network are tuned accordingly. The LSTM models has 2 to 3 hidden layers and training each model

took between 10 to 50 epochs. We have also considered both the Pareto and Poisson traffic distributions. However, since the results were very similar for both traffic types, we only report the ones for the Pareto distribution as it captures the bursty nature of Internet traffic [28]. We use the Tensorflow backend to build and train our LSTM models [52]. The training was performed on a machine with Intel XEON processor, Nvidia Quadro P2000 GPU card, and 64 GB of RAM. Training each LSTM model took on average about 4 to 8 hours.

4.3 Setting P-to-Q

To achieve the highest gain using Deep-DBA, P must be set as small as possible, and Q must be set as large as possible. However, the selection of these values must not be at the expense of high prediction error and poor network performance (in terms of packet latency and network throughput). Thus, we built different LSTM models for different P and Q values and compared the obtained MSE by running the models on the test set. For $P = 1$ (i.e., the smallest possible value for P), the LSTM models had significantly high errors; whereas for $P \geq 2$, the errors were adequate. Consequently, we varied the values of P and Q and measured the performance of each built model. As shown in Table 4.2a, we first built LSTM models with equal values of P and Q starting from 2. Results show that when P and Q are smallest, the lowest MSE is obtained. Next, to check the impact of increasing P, we built LSTM models with $Q = 2$ for different values of

Table 4.2: Mean Square Error with: a) $P = Q$, b) $P \geq Q$, c) $P \leq Q$.

P-to-Q	MSE
2-to-2	4.3×10^{-6}
4-to-4	8.8×10^{-6}
6-to-6	9.9×10^{-6}

(a)

P-to-Q	MSE
2-to-2	4.3×10^{-6}
4-to-2	5.7×10^{-6}
6-to-2	6.9×10^{-6}

(b)

P-to-Q	MSE
2-to-2	4.3×10^{-6}
2-to-4	4.8×10^{-6}
2-to-6	8.1×10^{-6}
2-to-8	8.8×10^{-6}
2-to-10	1.1×10^{-5}
2-to-12	1.3×10^{-5}
2-to-14	1.5×10^{-5}
2-to-16	1.7×10^{-5}
2-to-18	1.9×10^{-5}
2-to-20	2×10^{-5}

(c)

P. Results in Table 4.2b show that increasing P does not yield lower MSE and therefore will not have a positive impact on the performance. Finally, as shown in Table 4.2c, we set $P = 2$ and increase Q. As expected, the MSE increases as the value of Q increases.

Given these findings, we set $P = 2$, since increasing P would not offer lower MSE values. In addition, increasing P would entail increasing Q to even higher values, which in turn will cause higher prediction errors. For example, setting $P = 2$ and $Q = 8$ means for every $K = 10$ cycles, 8 cycles are prediction cycles, which sums up into 80% of all cycles being prediction cycles. Thus, to obtain the same percentage when $P = 4$, Q must be set to 16.

To choose the best value of Q, we compare the network performance under Deep-DBA with the Limited discipline for different values of Q. As shown in Fig. 4.2a, the throughput on high loads increases with increasing Q values, since

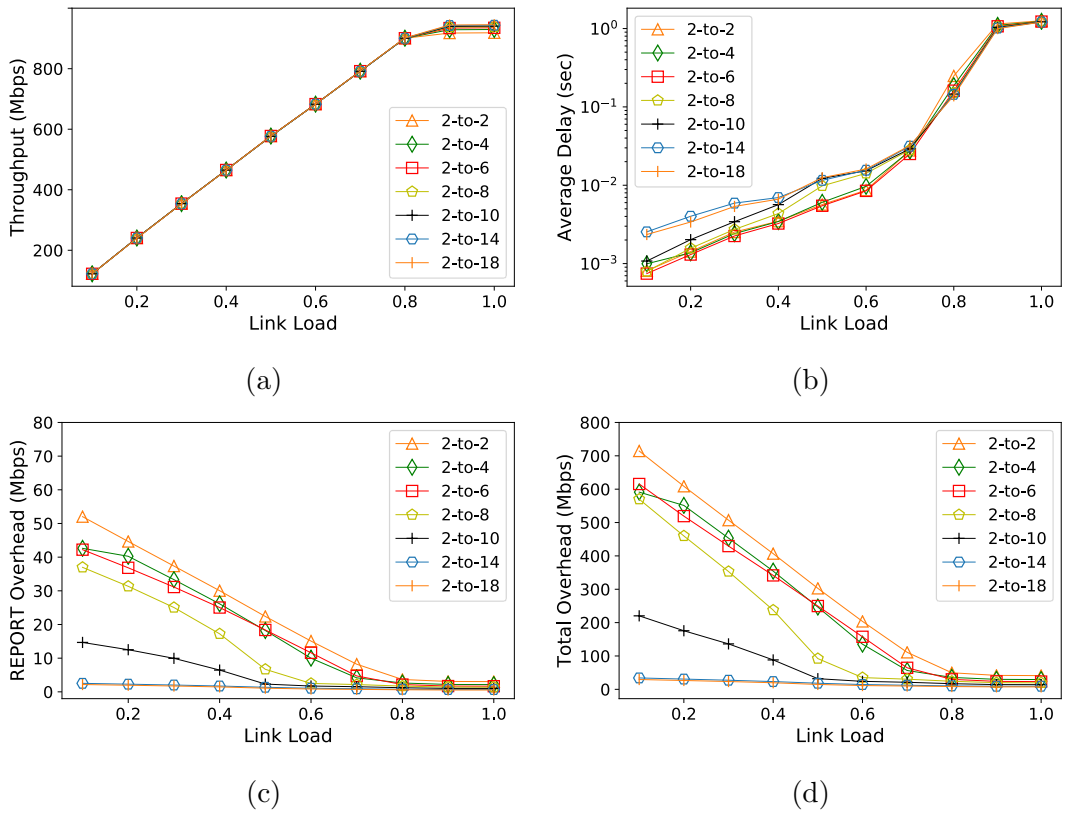


Figure 4.2: Comparison of different P-to-Q LSTM models: a) Throughput, b) Average delay, c) REPORT overhead, d) Total overhead.

increasing the number of prediction cycles will decrease both the REPORT and T_i^{end} overheads, leaving the gained bandwidth to be used by the ONUs. Fig. 4.2b shows small difference in delay for different models. However, the lowest delay is obtained for $Q \leq 6$, whereas the delay increases for higher values of Q. This is caused by the mis-predictions of these models; this behavior is related to the obtained MSE errors corresponding to each of these models. Fig. 4.2c and Fig. 4.2d highlight how both the REPORT and total overhead (i.e., $T_i^{end} + R_i$) decrease as the value of Q increases. However, for high values of Q, the low REPORT and total overhead bandwidth are due to the long cycle times caused

by over-predicting ONU bandwidth demands by the LSTM model. These over-predictions cause the OLT to grant larger transmission windows compared to what is actually needed by the ONUs, which results in wasted bandwidth. Thus, choosing the best Q value would be equivalent to maximizing the throughput, meanwhile minimizing the average delay and total wasted bandwidth (which comprises both the total overhead and prediction error wasted bandwidth). This is equivalent to maximizing the following objective function:

$$f(P,Q) = \frac{throughput(P,Q)}{delay(P,Q) \times waste(P,Q)}. \tag{4.1}$$

Therefore, to choose the best P-to-Q ratio, we normalize the different parameters, which are obtained from simulations, and plot $f(P,Q)$ in Fig. 4.3. Results show that the best P-to-Q under the Limited discipline is 2-to-6, with an MSE of 8.1×10^{-6}. In contrast, the best P-to-Q value under the Gated scheme is 2-to-2, with an MSE of 1.7×10^{-4}. Yet, it can be observed that the 2-to-4 model could also achieve a "good-enough" trade-off between an "acceptable" $f(P,Q)$ value and higher network utilization. We note that the MSE under the Gated scheme is higher compared to the Limited scheme due to the high fluctuations of queue sizes, especially at higher loads, making training of such models more difficult.

We validate the performance of the best P-to-Q models under the Limited and Gated schemes in Fig. 4.4 (i.e., with the 2-to-6, and 2-to-2 models, respectively), by comparing the predicted Internet traffic versus the actual Internet traffic.

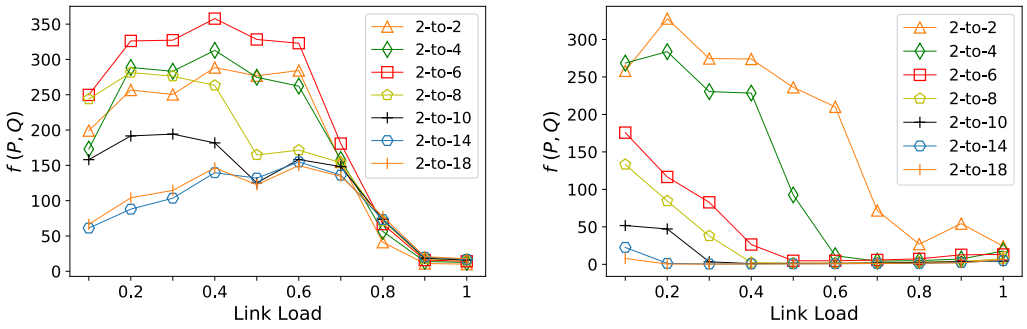
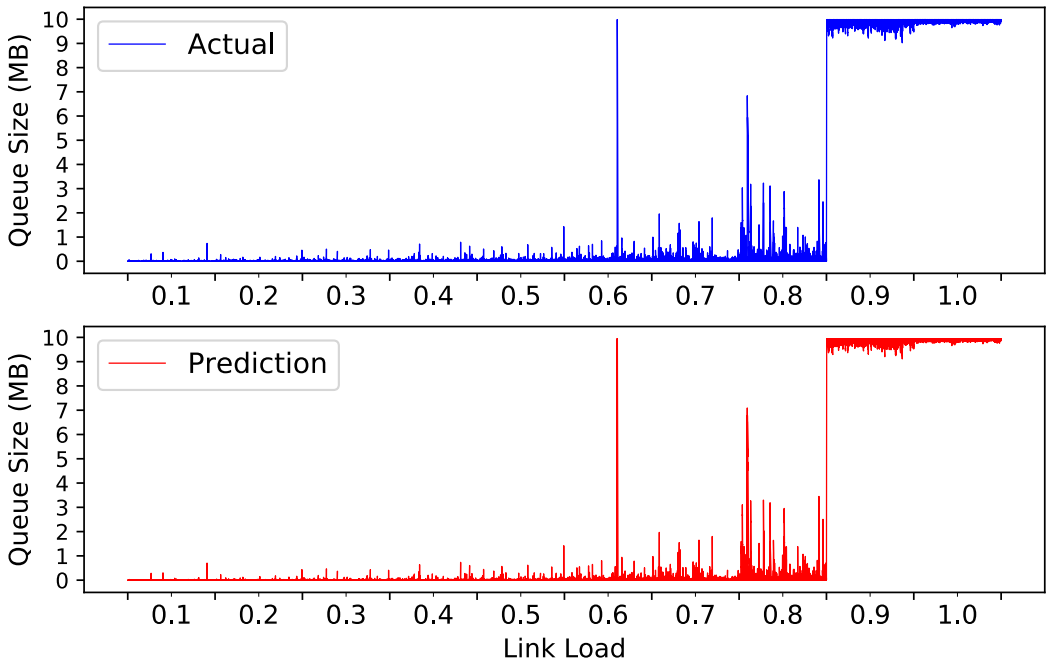


Figure 4.3: Choosing P-to-Q for: a) Limited scheme, b) Gated scheme.

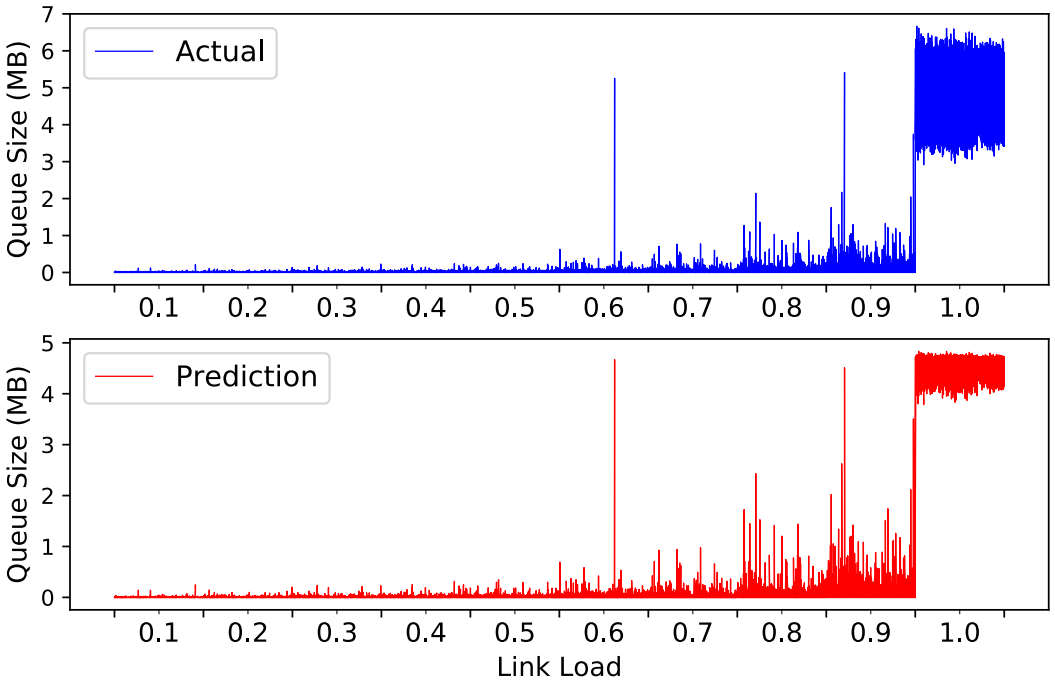
Here, "Link Load" corresponds to the load on the access link (i.e., between the user and the ONU). We observe that with the Limited scheme, the predicted traffic misses some very short bursts; however, it closely tracks the actual traffic overall. On the other hand, as expected, for the Gated scheme, the margin of mis-predicted queue size and incoming bursts is slightly larger (except at Load 1.0, which makes the system no longer in steady state). Moreover, we deduce that as Q increases and P decreases, the accuracy of the predictions decreases, and vice versa. Hence, there is a trade-off between the number of predicted cycles and the accuracy of predictions.

4.4 Comparison of Deep-DBA with other schemes

Fig. 4.5 compares the performance of the proposed Deep-DBA scheme under the Limited discipline (i.e., using the 2-to-6 LSTM network) with the prediction-based IPACT with Grant Estimation (IPACT-GE) scheme that predicts the size



(a)



(b)

Figure 4.4: Predicted vs. Actual bandwidth demand: a) Limited scheme (with 2-to-6), b) Gated scheme (with 2-to-2).

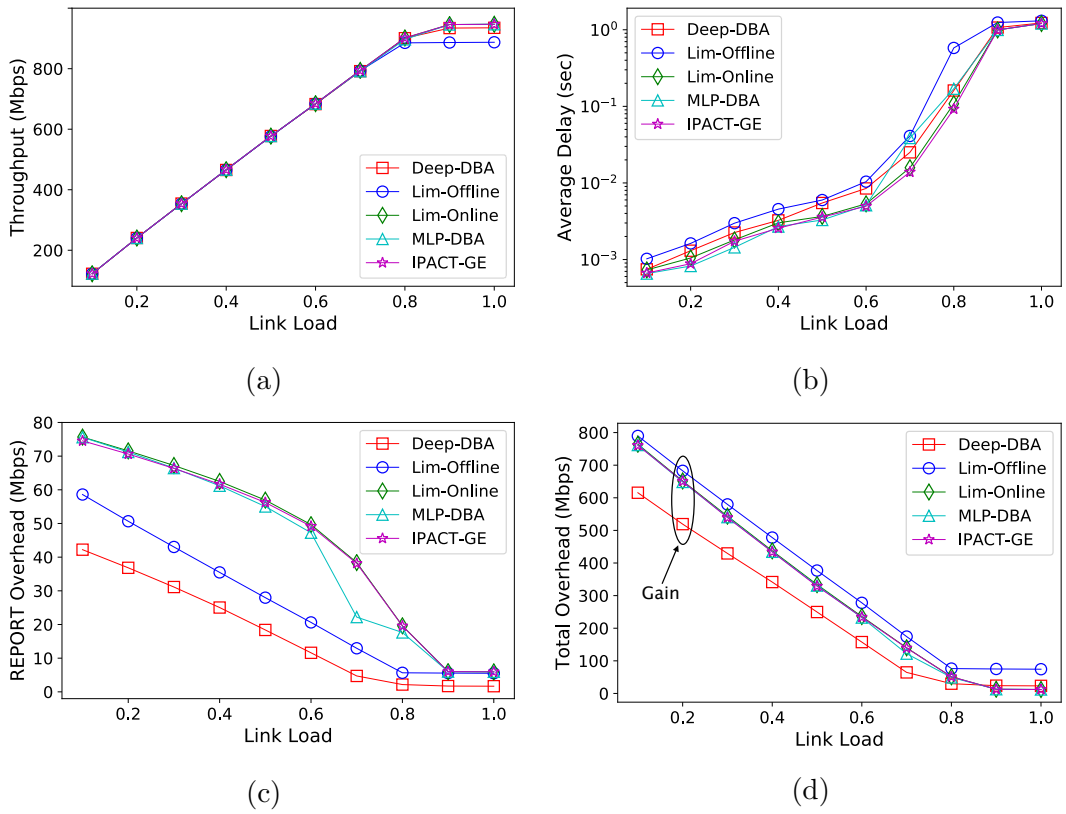


Figure 4.5: Comparison of schemes under the Limited discipline: a) Throughput, b) Average Delay, c) REPORT Overhead, d) Total Overhead

of incoming requests between two successive cycles [46], the legacy offline Limited (i.e., *Lim-Offline*) DBA scheme, the legacy online Limited (i.e., *Lim-Online*), and the most recent machine learning based predictive DBA (i.e., *MLP-DBA*) [17].

As shown in Fig. 4.5a, *Lim-Offline* exhibits the lowest throughput due to the control and T_i^{end} overheads. *Lim-Online*, *MLP-DBA*, and *IPACT-GE* exhibit higher throughput since they are online schemes and thus do not incur the T_i^{end} overhead. Deep-DBA exhibits increased throughput similar to the online schemes even though it is an offline scheme. This improvement is due to the reduction of the control and the T_i^{end} overheads.

As shown in Fig. 4.5b, the online schemes exhibit the lowest delay, which is expected since they have no T_i^{end} overhead, which reduces the idle and cycle times. *MLP-DBA* and *IPACT-GE* show slightly better results compared to the legacy *Lim-Online* since they use bandwidth prediction with the specific aim of decreasing the packet delay. The *Lim-Offline* scheme exhibits higher delays due to the overhead T_{REG} calculated in (3.7). On the other hand, even though Deep-DBA is an offline scheme, its performance is much better that the *Lim-Offline* scheme and is closer to the online schemes. This is due to the gain achieved as per (3.8). However, as previously mentioned, lower packet delays could be attained using Deep-DBA for different P-to-Q ratios. Nevertheless, these may either affect the prediction accuracy and/or may not achieve the most optimal bandwidth utilization.

The control overhead due to REPORT messages can be observed in Fig. 4.5c. The online schemes have a higher REPORT overhead than the offline scheme. This is because the online schemes do not have the T_i^{end} overhead, which results in a shorter cycle time compared to the offline schemes. Typically, at lower loads, the cycle time is shorter, which causes more control messages to be exchanged in short periods of time; as such the control overhead decreases as the load increases. However, we notice that Deep-DBA achieves the lowest control overhead over all loads (e.g., around 42 Mbps with Deep-DBA, compared to 60 Mbps for the *Lim-Offline* scheme, and around 75 Mbps for the online schemes). At higher loads, the cycle time is equal to the maximum cycle time; hence, the control overhead

reaches its lowest value for all schemes (e.g., 5.5 Mbps with IPACT-GE, offline Limited, and offline Limited-GE), whereas it is equal to 1.5 Mbps with Deep-DBA. This highlights the advantages of Deep-DBA, which enables the OLT and ONUs to only exchange control messages in reporting cycles every K cycles, as opposed to every cycle.

Fig. 4.5d shows the total overhead observed in the network (which is the control messaging overhead plus the cycle idle time) under all schemes. As noticed, even though online schemes are optimized to reduce the idle time, Deep-DBA is still able to achieve the lowest total overhead. This bandwidth gain can be used so that more users can be provisioned in the network, and better QoS support can be attained. This again highlights the merits of Deep-DBA over existing approaches.

In Fig. 4.6, we compare the performance of Deep-DBA under the Gated discipline (i.e., using the 2-to-2 and 2-to-4 LSTM models) with the offline Gated DBA scheme (*Gated-Offline*). Due to their nature, the prediction of IPACT-GE in [46] and MLP-DBA in [17] cannot be directly applied to an offline Gated scheme. As shown in Fig. 4.6a, the 2-to-2 Deep-DBA provides the same throughput as the offline Gated discipline, whereas the 2-to-4 provides lower throughput, which is captured by $f(P, Q)$ in (4.1).

In Fig. 4.6b, the average delay with Deep-DBA is a bit higher than with *Gated-Offline*. This is due to the "mis-predictions" of the LSTM model, which will make the ONU buffer more packets (thus, request more bandwidth), thereby

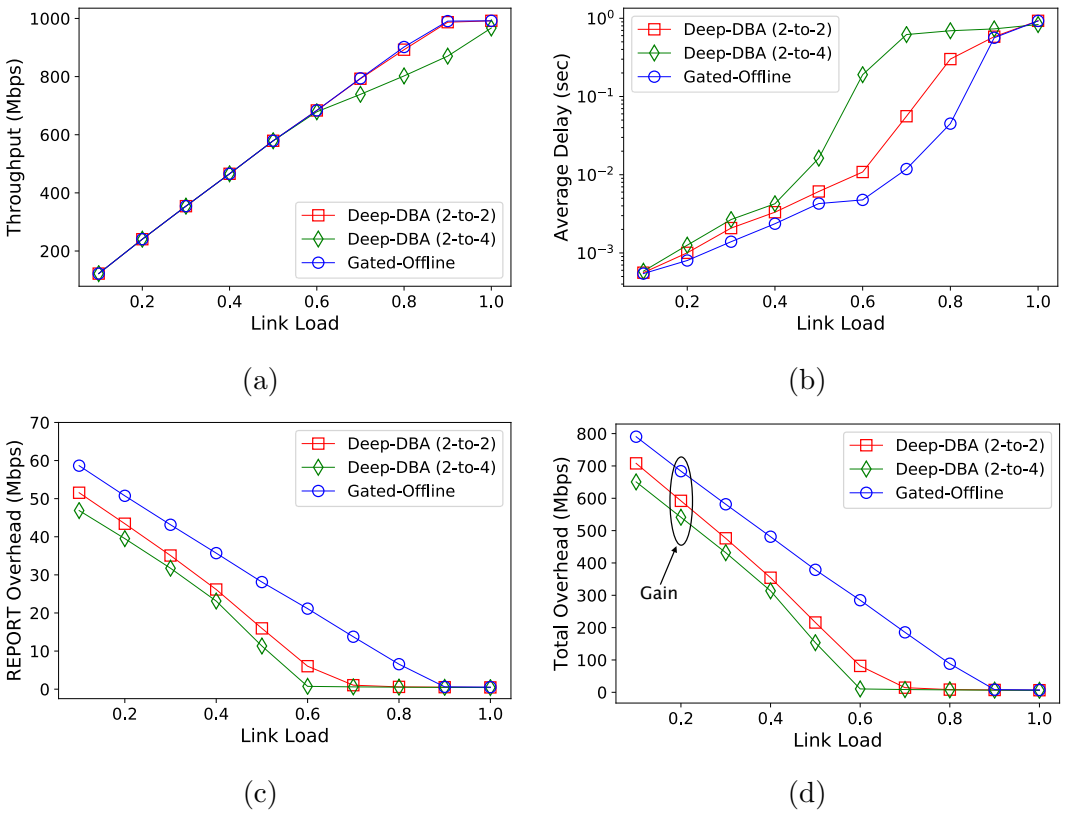


Figure 4.6: Comparison of schemes under the Gated discipline: a) Throughput, b) Average Delay, c) REPORT Overhead, d) Total Overhead

making the OLT grant the ONUs larger transmission windows even at low loads. This effect is caused by the nature of the Gated discipline, which unlike the Limited discipline, does not bound the bandwidth demand by a maximum value; thus, the mis-prediction would have a snowballing effect.

Finally, Fig. 4.6c and Fig. 4.6d show how Deep-DBA exhibit lower REPORT and total overheads than *Gated-Offline*. More importantly, the results here show how the chosen model (i.e., 2-to-2 or 2-to-4) presents a trade-off between higher bandwidth utilization (that is, higher bandwidth gain) and downgraded network performance (i.e., throughput and delay).

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this work we proposed Deep-DBA, a novel DBA scheme for PON, which employs deep learning to predict the bandwidth demands of ONUs for several future polling cycles by peep-holing only a few previous cycles, so as to reduce the overhead due to the request-grant mechanism. Results demonstrate how Deep-DBA is able to combine the advantages of both the online and offline schemes, thereby improving the network utilization achieved with online schemes, and at the same time maintaining the properties of fairness and QoS support that offline schemes enable, without impairing the network's performance. The fast progress in the field of machine learning promises new and better architectures and techniques that will be able to increase the number of prediction cycles and decrease the prediction error. The proposed method has the flexibility to employ

any current or future sequence-to-sequence machine learning model. Moreover, Deep-DBA can operate with any scheduling scheme.

5.2 Future Work

Our proposed work paves the way to further investigate and solve some interesting and potential issues. We list some of them as follows:

- According to [53], the global energy consumption will increase by 29 percent by year 2040, with ICT accounting for 2 percent of the total CO_2 emmissions, which is expected to double in the next 10 years [54]. Several DBA algorithms were proposed in the literature to reduce the energy consumption of PON. However, with such schemes, the average packet delay in the network is notably increased to prolong the sleep time of the network components. Energy-aware PONs will reduce the green house gas emissions as well as the energy cost. Nevertheless, this should not affect the network performance, and must ensure QoS, especially for delay sensitive applications like Tactile Internet. Therefore, with the use of deep learning, an energy-aware version of the proposed Deep-DBA, that reduces power consumption without affecting network performance in terms of delay and utilization, can be an interesting extension of this work.

- Machine learning can be employed to perform Internet traffic prediction for different classes of service (i.e., EF, AF, BE). This has the potential of

providing better QoS for high-priority traffics without starving low-priority traffics. In addition, predicting the global demand of all the ONUs in the network will give a comprehensive view of the traffic demand of the whole network. This can better organize and oversee granting procedure for different priority queues and delay sensitive flows. Moreover, modifying Deep-DBA to support Intra-ONU and Inter-ONU scheduling requires further investigation.

- As discussed in Section 3.1, in addition to REPORT messages, the bandwidth demands can be predicted as incoming traffic flows at the end-user side and/or as GATE messages. Further studies are needed to compare the effects of these predictions on the network performance, especially that traffic flows are independent from the employed DBA algorithm and the network architecture and settings.

- Improving the long-horizon forecasting of the LSTM model so as to increase the accuracy and the number of predicted future cycles would increase the reduced overhead and further increase the network bandwidth utilization.

Appendix A

Acronyms

ADSL	Asymmetric DSL
AF	Assured Forwarding
AI	Artificial Intelligence
ANN	Artificial Neural Network
BE	Best Effort
CBR	Constant Bit Rate
CIR	Committed Information Rate
CM	Cable Modem
CO	Central Office
DBA	Dynamic Bandwidth Allocation
DSL	Digital Subscriber Line
EF	Expedited Forwarding
EIR	Excess Information Rate
EPF	Earliest Packet First
EPON	Ethernet PON
FFNN	Feed Forward Neural Network
GPU	Graphics Processing Unit
HOL	head-of-line
ICT	Information and Communication Technology
IoT	Internet of Things
IP	Internet Protocol
IPACT	Interleaved Polling with Adaptive Cycle Time
IPACT-GE	IPACT with Grant Estimation
LQF	Longest Queue First
LR-PON	Long-Range PON
LSTM	Long Short-Term Memory
LSTP	Limited Sharing with Traffic Prediction
MAC	Medium Access Control
ML	Machine Learning

MLP	Multi-Layer Perceptron
MLP-DBA	ML based Predictive DBA
MPCP	Multi-Point Control Protocol
MSE	Mean Squared Error
NG-PON	Next Generation PON
NN	Neural Network
OLT	Optical Line Terminal
ONU	Optical Network Unit
PIR	Peak Information Rate
PON	Passive Optical Network
QoS	Quality of Service
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RTT	Round Trip Time
SAE	Stacked AutoEncoder
SLA	Service Level Agreement
SPID	SLA Proportional-Integral-Derivative
TDM	Time Division Multiplexing
VDSL	very high-speed DSL
VoIP	Voice-over-IP
WDM	Wavelength Division Multiplexing

Bibliography

[1] A. R. Dhaini, P.-H. Ho, and G. Shen, "Toward Green Next-Generation Passive Optical Networks," *IEEE Communications Magazine*, vol. 49, no. 11, 2011.

[2] M. P. Mcgarry, M. Reisslein, and M. Maier, "Ethernet Passive Optical Network Architectures and Dynamic Bandwidth Allocation Algorithms," *IEEE Communications Surveys Tutorials*, vol. 10, no. 3, pp. 46–60, 2008.

[3] L. G. Kazovsky, N. Cheng, W.-T. Shaw, D. Gutierrez, and S.-W. Wong, *Broadband Optical Access Networks*. John Wiley & Sons, 2011.

[4] "The Zettabyte Era: Trends and Analysis." Cisco White Paper, 2017.

[5] G. Kramer, *Ethernet Passive Optical Networks*. McGraw Hill Professional, 2005.

[6] B.-W. Kim, H. Song, and B. Mukherjee, "Long-Reach Optical Access," in *Broadband Access Networks* (A. Shami, M. Maier, and C. Assi, eds.), ch. 10, Springer, 2009.

[7] G. Kramer, "How Efficient is EPON?," *white paper, available at www. ieeecommunities. org/epon*, 2007.

[8] R. Roy, G. Kramer, M. Hajduczenia, and H. J. Silva, "Performance of 10G-EPON," *IEEE Communications Magazine*, vol. 49, no. 11, 2011.

[9] G. P. Fettweis, "The Tactile Internet: Applications and Challenges," *IEEE Vehicular Technology Magazine*, vol. 9, pp. 64–70, March 2014.

[10] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, "A Comprehensive Survey on Machine Learning for Networking: Evolution, Applications and Research Opportunities," *Journal of Internet Services and Applications*, vol. 9, no. 1, p. 16, 2018.

[11] F. Musumeci, C. Rottondi, A. Nag, I. Macaluso, D. Zibar, M. Ruffini, and M. Tornatore, "A Survey on Application of Machine Learning Techniques in Optical Networks," *arXiv preprint arXiv:1803.07976*, 2018.

[12] J. Mata, I. de Miguel, R. J. Duran, N. Merayo, S. K. Singh, A. Jukan, and M. Chamania, "Artificial Intelligence (AI) Methods in Optical Networks: A Comprehensive Survey," *Optical Switching and Networking*, 2018.

[13] N. E. Frigui, T. Lemlouma, S. Gosselin, B. Radier, R. L. Meur, and J. Bonnin, "Optimization of the Upstream Bandwidth Allocation in Passive Optical Networks Using Internet Users' Behavior Forecast," in *2018 International Conference on Optical Network Design and Modeling (ONDM)*, pp. 59–64, May 2018.

[14] P. Sarigiannidis, D. Pliatsios, T. Zygiridis, and N. Kantartzis, "DAMA: A Data Mining Forecasting DBA Scheme For XG-PONs," in *2016 5th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, pp. 1–4, May 2016.

[15] L. Ruan and E. Wong, "Machine Intelligence in Allocating Bandwidth to Achieve Low-Latency Performance," in *2018 International Conference on Optical Network Design and Modeling (ONDM)*, pp. 226–229, May 2018.

[16] E. Wong, M. P. I. Dias, and L. Ruan, "Predictive Resource Allocation for Tactile Internet Capable Passive Optical LANs," *Journal of Lightwave Technology*, vol. 35, pp. 2629–2641, July 2017.

[17] L. Ruan, S. Mondal, and E. Wong, "Machine Learning Based Bandwidth Prediction in Tactile Heterogeneous Access Networks," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–2, April 2018.

[18] A. S. Thyagaturu, A. Mercian, M. P. McGarry, M. Reisslein, and W. Kellerer, "Software Defined Optical Networks (SDONs): A Comprehensive Survey," *IEEE Communications Surveys Tutorials*, vol. 18, pp. 2738–2786, Fourthquarter 2016.

[19] J. Zheng and H. T. Mouftah, "A Survey of Dynamic Bandwidth Allocation Algorithms for Ethernet Passive Optical Networks," *Optical Switching and Networking*, vol. 6, no. 3, pp. 151–162, 2009.

[20] V. Houtsma, D. van Veen, and E. Harstead, "Recent Progress on Standardization of Next-Generation 25, 50, and 100G EPON," *Journal of Lightwave Technology*, vol. 35, no. 6, pp. 1228–1234, 2017.

[21] C. Knittle, "IEEE 100G-EPON," in *Optical Fiber Communication Conference*, pp. Th1I–6, Optical Society of America, 2016.

[22] J. Zheng and H. T. Mouftah, "Media Access Control for Ethernet Passive Optical Networks: An Overview," *IEEE Communications Magazine*, vol. 43, no. 2, pp. 145–150, 2005.

[23] G. Kramer, B. Mukherjee, and G. Pesavento, "IPACT a Dynamic Protocol for an Ethernet PON (EPON)," *IEEE Communications Magazine*, vol. 40, no. 2, pp. 74–80, 2002.

[24] T. Holmberg, "Analysis of EPONs Under the Static Priority Scheduling Scheme with Fixed Transmission Times," in *Next Generation Internet Design and Engineering, 2006. NGI'06. 2006 2nd Conference on*, pp. 8–pp, IEEE, 2006.

[25] S. Bhatia, D. Garbuzov, and R. Bartos, "Analysis of the Gated IPACT Scheme for EPONs," in *Communications, 2006. ICC'06. IEEE International Conference on*, vol. 6, pp. 2693–2698, IEEE, 2006.

[26] A. Dhaini and C. Assi, "Quality of Service in Ethernet Passive Optical Networks (EPONs)," in *Broadband Access Networks* (A. Shami, M. Maier, and C. Assi, eds.), ch. 10, Springer, 2009.

[27] P. Cortez, M. Rio, M. Rocha, and P. Sousa, "Multi-Scale Internet Traffic Forecasting Using Neural Networks and Time Series Methods," *Expert Systems*, vol. 29, no. 2, pp. 143–155, 2012.

[28] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (extended version)," *IEEE/ACM Transactions on networking*, vol. 2, no. 1, pp. 1–15, 1994.

[29] A. Azzouni and G. Pujolle, "NeuTM: A Neural Network-Based Framework for Traffic Matrix Prediction in SDN," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–5, IEEE, 2018.

[30] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, "Sketch-based Change Detection: Methods, Evaluation, and Applications," in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pp. 234–247, ACM, 2003.

[31] A. Sang and S.-q. Li, "A Predictability Analysis of Network Traffic," *Computer networks*, vol. 39, no. 4, pp. 329–345, 2002.

[32] K. Papagiannaki, N. Taft, Z.-L. Zhang, and C. Diot, "Long-Term Forecasting of Internet Backbone Traffic," *IEEE transactions on neural networks*, vol. 16, no. 5, pp. 1110–1124, 2005.

[33] A. Esposito, F. Giudicepietro, S. Scarpetta, and S. Khilnani, *Multidisciplinary Approaches to Neural Computing*. Springer, 2016.

[34] T. P. Oliveira, J. S. Barbar, and A. S. Soares, "Multilayer Perceptron and Stacked Autoencoder for Internet Traffic Prediction," in *IFIP International Conference on Network and Parallel Computing*, pp. 61–71, Springer, 2014.

[35] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition," *arXiv preprint arXiv:1402.1128*, 2014.

[36] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] A. Azzouni and G. Pujolle, "A Long Short-Term Memory Recurrent Neural Network Framework for Network Traffic Matrix Prediction," *arXiv preprint arXiv:1705.05690*, 2017.

[38] R. Vinayakumar, K. Soman, and P. Poornachandran, "Applying Deep Learning Approaches for Network Traffic Prediction," in *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on*, pp. 2353–2358, IEEE, 2017.

[39] F. A. Gers and E. Schmidhuber, "LSTM Recurrent Networks Learn Simple Context-Free and Context-Sensitive Languages," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1333–1340, 2001.

[40] G. Kramer, B. Mukherjee, S. Dixit, Y. Ye, and R. Hirth, "Supporting Differentiated Classes of Service in Ethernet Passive Optical Networks," *Journal of Optical Networking*, vol. 1, no. 8, pp. 280–298, 2002.

[41] Y. Luo and N. Ansari, "Limited Sharing with Traffic Prediction for Dynamic Bandwidth Allocation and QoS Provisioning Over Ethernet Passive Optical Networks," *Journal of Optical Networking*, vol. 4, no. 9, pp. 561–572, 2005.

[42] I.-S. Hwang, Z.-D. Shyu, L.-Y. Ke, and C.-C. Chang, "A Novel Early DBA Mechanism with Prediction-based Fair Excessive Bandwidth Allocation Scheme in EPON," *Computer Communications*, vol. 31, no. 9, pp. 1814–1823, 2008.

[43] S. De, V. Singh, H. M. Gupta, N. Saxena, and A. Roy, "A New Predictive Dynamic Priority Scheduling in Ethernet Passive Optical Networks (EPONs)," *Optical Switching and Networking*, vol. 7, no. 4, pp. 215–223, 2010.

[44] I. Hwang, J. Lee, K. Lai, and A. Liem, "Generic QoS-Aware Interleaved Dynamic Bandwidth Allocation in Scalable EPONs," *IEEE/OSA Journal*

of Optical Communications and Networking, vol. 4, pp. 99–107, February 2012.

[45] X. Li, L. Dan, and Q. Wu, "Adaptive Dynamic Bandwidth Allocation Algorithm Supporting Multi-Services over Ethernet Passive Optical Networks," *Optik - International Journal for Light and Electron Optics*, vol. 124, no. 4, pp. 287 – 291, 2013.

[46] Y. Zhu and M. Ma, "IPACT With Grant Estimation (IPACT-GE) Scheme for Ethernet Passive Optical Networks," *Journal of Lightwave Technology*, vol. 26, no. 14, pp. 2055–2063, 2008.

[47] N. Merayo, D. Juárez, J. C. Aguado, I. D. Miguel, R. J. Durán, P. Fernández, R. M. Lorenzo, and E. J. Abril, "PID Controller Based on a Self-Adaptive Neural Network to Ensure QoS Bandwidth Requirements in Passive Optical Networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, pp. 433–445, May 2017.

[48] N. E. Frigui, T. Lemlouma, S. Gosselin, B. Radier, R. L. Meur, and J. Bonnin, "Dynamic Reallocation of SLA Parameters in Passive Optical Network Based on Clustering Analysis," in *2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pp. 1–8, Feb 2018.

[49] Y. Tian and L. Pan, "Predicting Short-Term Traffic Flow by Long Short-Term Memory Recurrent Neural Network," in *Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on*, pp. 153–158, IEEE, 2015.

[50] A. Vargas, "Omnet++."

[51] A. R. Dhaini, P.-H. Ho, G. Shen, and B. Shihada, "Energy Efficiency in TDMA-based Next-Generation Passive Optical Access Networks," *IEEE/ACM Transactions on Networking*, vol. 22, no. 3, pp. 850–863, 2014.

[52] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015. Software available from tensorflow.org.

[53] "International Energy Outlook 2017." [Online]. Available: https://www.eia.gov/outlooks/ieo/index.php.

[54] "Green Touch." [Online]. Available: http://www.greentouch.org.