

Article

# Traffic-Estimation-Based Low-Latency XGS-PON Mobile Front-Haul for Small-Cell C-RAN Based on an Adaptive Learning Neural Network

Ahmed Mohammed Mikaeil <sup>1,\*</sup> , Weisheng Hu <sup>1,\*</sup> , Syed Baqar Hussain <sup>1,2</sup>   
and Amber Sultan <sup>1</sup> 

<sup>1</sup> State Key Laboratory of Advanced Optical Communication Systems and Networks, Shanghai Jiao Tong University, No. 800, Road Dongchuan, Shanghai 200240, China; baqar.hussain@sjtu.edu.cn (S.B.H.); ambersultan@sjtu.edu.cn (A.S.)

<sup>2</sup> Department of Electronics and Electrical Systems, The University of Lahore, Lahore 54590, Pakistan

\* Correspondence: ahmed\_mikaeil@yahoo.co.uk (A.M.M.); wshu@sjtu.edu.cn (W.H.); Tel.: +86-130-6188-5290 (A.M.M.)

Received: 6 May 2018; Accepted: 28 June 2018; Published: 6 July 2018



**Abstract:** In this paper, we propose a novel method for low-latency 10-Gigabit-capable symmetric passive optical network (XGS-PON) mobile front-haul for small cell cloud radio access network (C-RAN) based on traffic estimation. In this method, the number of packets that arrive to the optical network unit (ONU) buffer from the remote radio unit (RRU) link is predicted using an adaptive learning neural network function integrated into the dynamic bandwidth allocation (DBA) module at the optical line terminal (OLT). By using this predictive method, we are able to eliminate the additional DBA processing delay and the delay required for reporting ONU buffer occupancy to the OLT. As a result, the latency is as low as required for mobile front-haul in C-RAN architecture. The performance of the new method is evaluated by means of simulation under XGS-PON standard. The simulation results confirmed the capability of the proposed method of achieving the latency requirement for mobile front-haul while outperforming some other XGS-PON standard compliant algorithms that are optimized to support mobile front-haul and backhaul traffic.

**Keywords:** C-RAN; front-haul; traffic estimation; neural network; XGS-PON

## 1. Introduction

Time division multiplexing passive optical networks (TDM-PONs) (e.g., 10-Gigabit-capable passive optical network (XG-PON) and 10 Gb/s Ethernet passive optical network (10G-EPON)) are considered promising technologies for realizing cost-efficient mobile front-haul for small cell C-RAN, as they allow the sharing of optical fibers and transmission equipment. However, the latency in TDM-PONs upstream transmission due to the DBA mechanism is much higher than the latency required for mobile front-haul in C-RAN architecture (i.e., 300  $\mu$ s [1]). There is considerable research on different low-latency DBA methods to support mobile front-haul over TDM-PONs system. However, most of the focus was on IEEE EPON-based mobile front-haul. For example, the DBA method reported in [2], which minimizes the upstream latency of 10G-EPON-based mobile front-haul by reducing the DBA cycle length through reduction of DBA grant processing time. Another example is a simple statistical traffic analysis-based DBA method [3], which introduces the usage of simple statistical traffic estimation function with the fixed bandwidth allocation (FBA) method to accommodate dynamic data-rate mobile front-haul. However, these methods are not fully compatible with international telecommunication union (ITU) TDM-PON standards such as XGS-PON. As an alternative, there are some other TDM-PON-based low-latency front-haul proposals

reported in the literature that are compatible with both XG-PON and 10G-EPONs standards such as the Mobile-DBA proposal [4,5], in which the mobile uplink scheduling information is utilized to schedule the grants for the ONUs to minimize the upstream latency of TDM-PON-based mobile front-haul. Another example is the Mobile-PON proposal [6], which focuses on eliminating the conventional PON scheduling by unifying the TDM-PON scheduler and long-term evolution (LTE) mobile scheduler (i.e., physical layer functional split-based resource block scheduler). However, these proposals require some modifications to current TDM-PONs and BBU processing and protocol; for example, Mobile-DBA requires an additional interface between the baseband unit (BBU) and OLT to exchange the mobile scheduling information, whereas, Mobile-PON requires a modification on current TDM-PON system regarding ONUs synchronization.

The only works that are fully compatible with XG-PON standard are: (1) Group-Assured Giga-PON Access Network DBA (gGIANT DBA) introduced in [7], which has been proposed to optimize XG-PON average upstream delay to support mobile backhaul; however, the delay requirement for mobile front-haul is much lower than in backhaul case; (2) Our pervious proposal, Optimized Round Robin DBA (Optimized-RR DBA), was introduced in [8] to optimize XG-PON to transport MAC-PHY split-based front-haul traffic in small-cell C-RAN architecture. In [8], a front-haul latency as low as 300  $\mu$ s was achieved with Optimized-RR DBA; however, taking into account the new emerging 5G QS requirements, front-haul transmission latency should not exceed 250  $\mu$ s, as recommended by the Next-Generation Mobile Networks Alliance (NGMA) in [9]. Motivated by the above facts, to achieve front-haul transmission latency similar to that recommended by NGMA without modifying current the TDM-PONs and BBU systems' processing and protocols, this paper proposes a traffic-estimation-based low-latency XGS-PON mobile front-haul to support small-cell-based cloud radio access network architecture.

In this paper, we consider the small-cell-based cloud radio access network architecture depicted in Figure 1, where the Physical layer split (PHY split) architecture reported in [6] (i.e., the split between the wireless resource de-mapping and the modulation decoding) is considered the split point between BBU and RRU functional entities of small-cell baseband (note: three functional entities are currently defined for 5G baseband: Centralized Unit (CU), Distributed Unit (DU), and RRU). The CU provides the non-real-time functionalities while the DU provides physical layer and the real-time functions [10]. In the C-RAN architecture (Figure 1), the instantaneous actual LTE system traffic load, in the form of antenna symbols, is mapped into bit-streams. Then these bit-streams are encapsulated into an Ethernet packet and transmitted between RRU and BBU via XGS-PON mobile front-haul. For efficient utilization of XGS-PON mobile front-haul, while avoiding collisions between packets transmitted from different RRUs at the same time, the DBA mechanism is used at the OLT side to govern the assignment of transmission opportunities (i.e., grants). In XGS-PON standard, the DBA engine at OLT assigns the transmission grants to the ONUs/RRUs based on their buffer occupancy reports or their transmission containers, which are also known as T-CONTs (note: in this paper we assume that the ONU has only one T-CONT, as in [7]); therefore, in this paper we will use the term ONU instead of T-CONT).

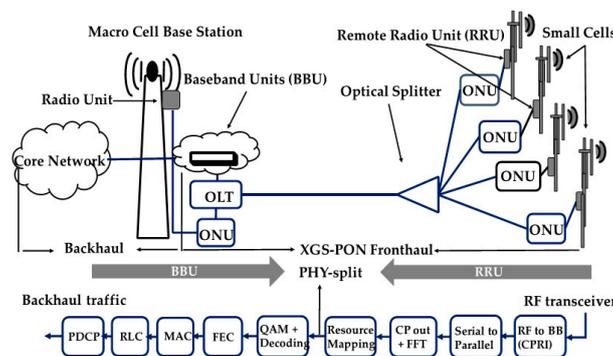


Figure 1. Physical layer split based small-cell C-RAN architecture.

In [8] we have shown that giving the grants to the ONUs based on their buffer occupancy reports, while adjusting the pre-determined limits for congested or heavily-loaded ONUs in the network, can efficiently accommodate the traffic variation of LTE/LTE advance networks (i.e., the actual RRUs traffic load) and attains a considerable delay performance improvement. As a continuation of this work, here we propose predicting the amount of ONU’s buffer occupancy in advance based on historical buffer occupancy reports collected during BBU processing time. Based on these predicted reports, the DBA engine at the OLT prepares in advance the grants for each ONU minimizing the transmission latency for mobile front-haul without modifying the TDM-PON system and the BBU processing and protocols.

We formulate the problem of predicting the future amount of ONU buffer occupancy (i.e., next ONU report) as a machine learning function approximation problem. Although there are multiple machine learning methods to solve such an approximation problem, we focus on artificial neural network (ANNs) as they are considered the most efficient tools to solve function approximation problems [11]. According to Honrik et al. [12], ANNs are able to estimate any arbitrary nonlinear function with any degree of accuracy required. For this reason, in this paper we adopt an adaptive-learning-based artificial neural network to solve the function approximation problem we mentioned earlier. To the best of our knowledge, there are no any prior studies that have applied the ANNs approach for traffic estimation of mobile front-haul networks.

The remainder of this paper is organized as follows. The reported estimation-based bandwidth allocation problem is formulated as a machine learning function approximation problem in Section 2. In Section 3, we introduce an adaptive learning method to solve the function approximation problem and predict the upcoming ONU buffer occupancy report. In Section 4, we discuss the performance evaluation results of the proposed method. We conclude our paper in Section 5.

## 2. Report Estimation-Based Bandwidth Allocation Problem Formulation

In frequency division duplex (FDD)-based LTE, the downlink and uplink sub-frames are time-aligned at BBU, as illustrated in Figure 2. To meet the timing requirements imposed by the hybrid automatic repeat request (HARQ) protocol, every received MAC protocol data unit (PDU) has to be acknowledged (ACK) or non-acknowledged (NACK) back to the UE transmitter within the deadline of 4 ms. Thus, all the processing including air interface uplink and downlink processing, front-haul processing uplink and downlink processing and BBU processing should be completed within this 4 ms deadline. Within this 4 ms, BBU requires around 2.3~2.6 ms [13] to finish preparing and sending the downlink ACK/NACK back to UE in such a way that when it receives the uplink data from UE in sub-frame n, it will send ACK/NACK to the UE in sub-frame n+4.

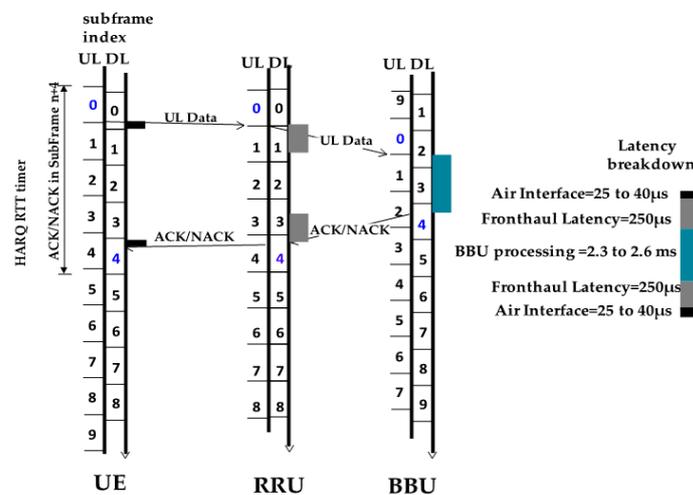


Figure 2. Long-term evolution-based radio access network processing latency breakdown.

Our idea is to take advantage of this 2.3 ms BBU processing time for collecting data statistics (i.e., buffer occupancy reports) from ONUs. Then we will utilize these reports as training examples (samples) to train a prediction model to predict the next reports for the ONUs. We can utilize these predicted reports to generate the scheduling grants for the ONUs and formulate the final bandwidth map for the upcoming XGS-PON frames.

The number of ONU reports that can be collected within one BBU processing time is given by:

$T = \text{BBU processing time} / (\text{one-way propagation delay to transmit one frame from ONU to OLT} + (125 \mu\text{s: the time required for transmitting the previous frame}))$ . For example, for 10 km mobile front-haul, the number of reports will be equal to  $= 2.3 \text{ ms} / (50 + 125) \mu\text{s} = 13$  reports.

Assuming that we collect until  $T + 1$  reports from  $j$ th ONU during one BBU processing cycle, if we extend backward from  $T$ th report to the beginning of cycle, we will have a time series containing the following reports  $\{R_T^j, R_{T-1}^j, R_{T-2}^j \dots \dots R_0^j\}$  for each ONU (i.e.,  $j = 1, 2, 3 \dots K$ , where  $j$  is the index of the ONU). Given this time series, if we want to estimate  $R_t^j$  at some future time  $T + s$ , we can write this problem as follows:

$$\hat{R}_{T+s}^j = f(R_T^j, R_{T-1}^j, R_{T-2}^j, \dots, R_0^j), \tag{1}$$

where  $s$  is the prediction horizon. Assuming  $s = 1$ , we can predict one time step or sample into the future  $\hat{R}_{T+1}^j$ , if we can find an optimal function estimator that can accurately approximate the function  $f$ . Finding such an estimator is a machine learning function approximation problem, which can be solved using machine learning function approximation models such as feedforward artificial neural network. The next section introduces a multi-layered feedforward artificial neural network that employs adaptive learning approach to solve this problem.

### 3. Problem Solution Using Adaptive Learning Artificial Neural Network

Artificial neural networks (ANNs) are constructed using computational functions known as neurons (Figure 3). ANNs' architectures include feedforward neural networks and feed-backward or recurrent neural networks. In feedforward-based artificial neural network (FANN), the neurons are arranged in layers with non-cycled connections between the layers [14] (see Figure 3). The first-layer neurons and the last-layer neurons in FANN represent the input and output variables of the function that we are going to approximate. Also, between the first layer and the last layer neurons in the network (i.e., the input and the output layer), there exist a number of hidden layers with weighted connections, which determine how the FANN performs. The output of the neuron in the hidden layer and the last layer are often fed into an activation function (e.g., sigmoid, linear and stepwise functions, etc.) to restrict the output value of the neurons to a certain finite value.

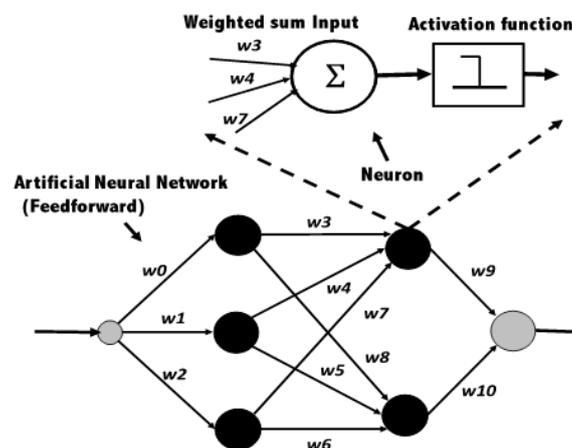


Figure 3. Feedforward artificial neural network.

FANN employs function approximation by learning from the input and output variables (or the training examples), which describe how the function works. Then it adjusts its internal network architecture and the connection weights to produce the same output given in the training examples so that, when FANN is given new examples or input variables, it will produce a correct output for them. However, FANN has the limitation that it can only learn the input–output mapping of the function, which does not change with time [15]. In the C-RAN architecture depicted in Figure 1, the function we are trying to approximate does change over time. This is due to the fact that mobile front-haul traffic transmits via XGS-PON system exhibits a high degree of temporal variation [8]. In order to make our FANN network capable of approximating the changing behavior of this function, we introduce an adaptive learning approach to train the FANN network.

In this approach, we update the learning set (i.e., the set containing the training and testing examples) periodically with a new set after receiving the last report within the BBU processing cycle (note: we assume the length of the learning set to be less than one BBU processing time cycle with one report time i.e.,  $R_{T-1} = R_{12}$ , when  $T = 13$  reports). To generate the new set, we update the existing set by adding the newest report received at the end of the BBU processing cycle and removing the oldest report received from it as illustrated in Figure 4. By doing this, we guarantee that after every BBU processing cycle, the overall learning set will be completely updated with a new instance of training examples of ONU reports. The goal behind using this adaptive learning method is to allow the FANN module to adjust its parameters during the learning process to capture the long-term information regarding the input traffic. This also allows the module to ignore the oldest training examples since they are irrelevant to the current actual ONU input traffic load. This minimizes the processing time and saves OLT system memory.

To solve the report estimation problem mentioned in Section 2, we first train the FANN module to learn the input–output mapping of the function  $f(R_{T-1}^j, R_{T-2}^j, R_{T-3}^j \dots \dots R_0^j)$  to an appropriate degree of accuracy or desired training error we want (note: in this paper we set a higher degree of accuracy by letting the desired training error to be equal to  $10^{-7}$  to guarantee that the prediction process will not introduce additional packet loss rate in the front-haul network). After successfully learning the function mapping, we run the trained FANN model with the set of  $R_T^j$  ( $T$ : total number of reports during the training) as inputs to predict the upcoming report  $\hat{R}_{T+1}^j$  for each ONU after training. Then we use the predicted value of these ONU reports to calculate and assign the final grants for the network.

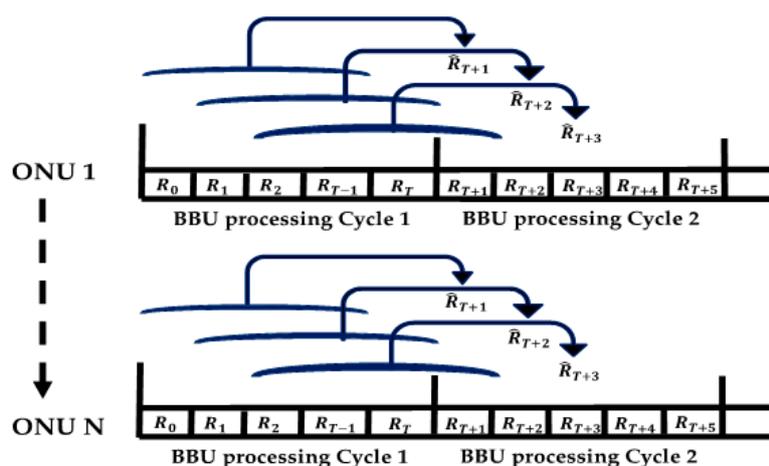


Figure 4. Adaptive learning method for training feedforward artificial neural network.

### 3.1. FANN Learning Phase

The learning phase of FANN model is done using the learning Algorithm 1, which takes a set containing the reports collected from each ONU in the network during a single BBU processing time

cycle as inputs and returns a trained FANN model for that specific ONU. We assume that during the first BBU processing time cycle the OLT gives fixed grants to each ONU in the network as follows:  $G_0^j = BW / K$ , where  $BW$  is the total XGS-PON frame size in bytes and  $K$  is the number of ONUs in the network. This allows OLT to collect the initial set of ONU reports for the training process. After collecting these reports in an individual initial set associated with each ONU (i.e.,  $\{R_0^j, R_1^j, R_2^j \dots R_{T-1}^j\}$ ,  $j = 1, 2, 3 \dots K$ ), we run the learning algorithm 1 to train the FANN module by taking each ONU set of reports  $R_t^j$  ( $t = 0, 1, 2, \dots T - 1$ ) as inputs and the last report collected from that ONU (i.e.,  $R_T^j$ ) as an output as illustrated in Figure 5a. We run the training process until we reach the desired training error. Then we save the network settings of the FANN module for each ONU so that it can be used later in the prediction phase. (Note: we run the training algorithm every time we receive a new set of reports from ONUs, since the initial training set for each ONU is updated upon the arrival of a new report from ONU.)

---

**Algorithm 1.** Learning

---

**Input:** A learning set containing  $j$ th ONU report  $R_t^j$ , where ( $t = 0, 1, 2, \dots T - 1$ )  
 FANN initial setting (e.g., number of layers, initial weights, activation function, learning rate, etc.)  
**Output:** A trained FANN model for  $j$ th ONU  
 Assume a generative FANN model for this ONU.  
**For**  $j = 0$ ;  $j >$  the total number of ONU  $K$ ,  $j++$  **do**  
**While** FANN does not converge to the desired error, **do**

- Train the FANN model with  $R_t^j$  set as inputs and the report following the set  $R_t^j$  (i.e.,  $R_T^j$ ) as an output.

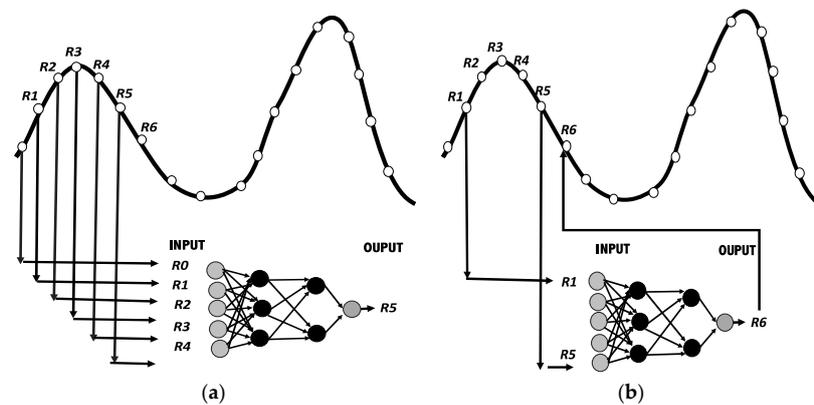
**End while**

- Test trained FANN model on the testing set.

**End for**

- Return the trained FANN model for this ONU.

---



**Figure 5.** Fast artificial neural network (a) FANN learning phase; (b) FANN prediction phase.

### 3.2. FANN Prediction Phase

The prediction phase starts after finishing the learning process when receiving the last ONU report in BBU processing cycle (i.e., the newest ONU report received after finishing the training and testing process). This phase is done using prediction Algorithm 2. The first step of this algorithm is used to update the learning set for ONUs by adjusting the initial ONU learning set by adding the newest report received after the training and testing process to the initial set and eliminating the earliest report received at beginning of the BBU process cycle from it. The second step of the algorithm is to perform the prediction process. In this step a set containing ONU reports up to the newest report

received after the learning phase  $R_t^j$  ( $t = 0, 1, 2 \dots T$ ) is provided as an input to the trained FANN module to predict the upcoming report  $R_{T+1}^j$  for each ONU (see Figure 5b). After predicting the reports for all the ONUs and collecting them in a set, this set of reports is provided as an input to the grant assignment Algorithm 3 to assigns the final grant to each ONU.

**Algorithm 2.** Prediction

**Input:** New ONU report received after the learning  $R_T^j$ ,  
 The learning set for  $j$ th ONU  $[R_0^j, R_1^j, \dots R_{T-1}^j]$ .  
 The trained FANN module for  $j$ th ONU.

**Output:** A set contain the predicted ONU reports  $\hat{R}_{T+1}^j$ .

**For**  $j = 0; j >$  the total number of ONU  $K, j++$  **do**

- Update learning set for  $j$ th ONU by adding the newly received report  $R_T^j$  to the end of the learning set and deleting the first report from it.
- Run the trained FANN network with the set of reports  $[R_0^j, R_1^j, \dots R_T^j]$  as inputs to predict  $\hat{R}_{T+1}^j$  for this ONU.

**End for**

Call the grant assignment Algorithm 3 with the set of the predicted ONU reports  $\hat{R}_{T+1}^j, j = 1, 2, 3 \dots K$ , to calculate the grants for ONUs for the upcoming XGS-PON upstream frame cycle.

**Algorithm 3.** Grant assignment

**Input:** Set of the predicted ONU reports  $\hat{R}_{T+1}^j$

**Output:** Bandwidth map for the upcoming XGS-PON frame

**Initialize:**  $N = 0; t = 1, 2, \dots; t = 1, 2, \dots; j = 0, Ex_0^j = 0; W_0^j = \frac{BW}{K}$

**For** every predicted ONU report  $\hat{R}_{T+1}^j$ , **do**

- Set  $B = False$
- Allocate a grant  $G_t^j$  to  $j$ th ONU using the updated limit from previous cycle ( $t - 1$ ),
- Update the counter of the number of scheduled ONUs  $j$

**If** ( $\hat{R}_{T+1}^j \leq W_{t-1}^j$ )  $G_t^j = \hat{R}_{T+1}^j$

**else**  $G_t^j = W_{t-1}^j$

**End If**

- Determine whether  $j$ -th T-CONT is overloaded in the current cycle; if so, update  $N$  accordingly.

**If**  $\hat{R}_{T+1}^j > W_0$

$N++; B = True$

**End If**

- Update the next cycle allocation byte limit for current ONU using

**If**  $B = True$   $W_t^j = W_0^j + Ex_{t-1}^j$

**else**  $W_t^j = W_0^j + Ex_{t-1}^j$ .

**End If**

**End For**

- Formulate the bandwidth map for ONUs
- Calculate ONUs excess share in this cycle as

$$Ex_t^j = (BW - \sum_{j=0}^K G_t^j) / N.$$

Repeat the procedure from the **Initialize** step to map the upcoming XGS-PON upstream frames.

### 3.3. Grant Assignment Phase

After receiving the set of predicted ONU reports from the prediction phase, these reports are used to compute and assign the grants for the ONUs in the network. In order to improve the utilization of the XGS-PON upstream bandwidth while optimizing the upstream transmission latency, we propose utilizing our previous grant assignment method, which we refer to as the Optimized-RR DBA algorithm [8], to compute and assign the grants for ONUs. However, instead of using actual or instantaneous ONU reports as inputs to the algorithm, as in [8], we utilize the predicated ONU's reports from the prediction phase as inputs to the algorithm.

Optimized-RR DBA algorithm improves the utilization of XGS-PON upstream by utilizing the unallocated remainder of XGS-PON upstream frames during every upstream allocation cycle. The algorithm calculates the amount of total excess bandwidth at the end of every upstream allocation cycle; then, it redistributes the calculated total excess bandwidth equally between the heavily-loaded ONUs in the upcoming allocation cycle. The pseudocode of this algorithm is illustrated in Algorithm 3 and the parameters used in this algorithm are explained as follows:  $W_i^j$ : ONU's dynamic updated Allocation Bytes limit for upcoming DBA allocation cycle,  $N$ : a counter for the number of heavily-loaded ONUs in one DBA allocation cycle,  $B$ : a Boolean variable to determine whether ONU is overloaded in current allocation cycle or not,  $j$ : a counter for the number of scheduled ONU in one allocation cycle,  $BW$ : total XGS-PON upstream frame in bytes (155.52 KB), and  $K$  is the total number of scheduled ONUs in the network.

In the next section we will evaluate the performance of our proposed method in terms of delay (the time required for the packet to travel from ONU buffer ingress to the OLT buffer egress), jitter (the variation of delay between every two successive packets reach to the OLT), packet drop ratio and upstream link utilization against the two other XGS-PON standard compliant algorithms listed below:

1. (gGIANT) DBA, which utilizes a service interval term (i.e., multiple XG-PON frame cycles) to assign the grant to the ONUs to capture the traffic burst-ness.
2. Optimized-RR DBA, which periodically serves every ONU in the network with a grant (i.e., amount of bytes) less than or equal to a pre-determined limit, while allowing heavily-loaded ONUs in the network to adjust their limit during every upstream frame cycle to capture the burst-ness of the input traffic.

## 4. Performance Evolution

To evaluate the performance our new traffic estimation-based bandwidth allocation method, which we refer to here as Adaptive-Learning DBA, compared to the two abovementioned DBA algorithms, we conducted simulation experiments using the XG-PON module [16] of the network simulator NS-3 [17]. We modified XG-PON module to simulate XGS-PON with 10 Gbps in both upstream and downstream directions. In our experiments, we considered a front-haul network with eight LTE small-cell RRUs (i.e., 20 MHz bandwidth channel, single carrier and single user per transmission time interval) connected to 8 ONUs, and each ONU has a buffer size equal to 1 Mbyte. We set 120  $\mu$ s as a round-trip propagation delay to represent the 10 km distance front-haul (note: the round-trip propagation delay of 10 km is 100  $\mu$ s; the 20  $\mu$ s is the additional delay margin to tolerate the latency results from ONUs and other front-haul elements processing). We allowed all ONUs to transmit buffer reports within every upstream frame (i.e., set the polling interval was set to 125  $\mu$ s for all DBAs as in [18]). We generated the mobile front-haul uplink traffic by injecting each ONU with Poisson Pareto Burst Process (PPBP) traffic [19] with Hurst parameter 0.8 and shape parameter 1.4 for a period of 10 s. To assure that the generated traffic is similar to the practical LTE network traffic, we allowed 10% of RRUs to generate 60% of the total aggregated front-haul uplink load as in [20], Table 1 summarizes the abovementioned simulation parameters.

**Table 1.** DBA simulation parameters.

Parameters	Details
Application Traffic Type	Poisson Pareto Burst Process (PPBP)
Simulation Time	10 s
PPBP Hurst Parameter	1.4
PPBP Shape Parameter	0.8
Poisson Arrivals Rates	[704,800~1376 Mbps]
Number of Bursts	5000
Propagation delay	120 $\mu$ s—Round-trip Time (RTT)
ONU Buffer or queue size	1 Mbytes
Packet Size	1470 Bytes
Polling Interval (all DBAs)	125 $\mu$ s
Number of RRUs (ONUs)	8
Traffic distribution	60% Load @10% Remote Radio Unit (RRU)

For the neural network module, we utilized a fast artificial neural network library [21]. To create the module, we used a function in the NS-3 simulator to generate the neural network module. To train the module, we used the value of the neural network settings given in Table 2. During the training process, we set the total learning time (i.e., training + testing time) to be 165  $\mu$ s. We stopped the learning process either when the desired error was achieved or when 165  $\mu$ s of learning time had elapsed. Also, we left a 10  $\mu$ s delay margin (i.e., 175–165  $\mu$ s) to be used as a waiting time for the prediction process of the upcoming ONU reports (i.e., the waiting time to run the module to the predict ONU reports). As we mentioned earlier in Section 2, 13 training samples can be collected during one BBU processing cycle, assuming a 10 km mobile front-haul distance. We divided these 13 samples as follows: we used 12 sample for the learning process by setting nine samples for training and three samples for testing (note: the learning process is done three times in this case; every time the training algorithm is trained on three samples and tested on the fourth sample). The values of the parameters used during the training process of FANN are summarized in Table 3.

**Table 2.** Fast artificial neural network module settings.

Parameters	Detail
Number Inputs layers	12
Number Outputs layers	1
Number Hidden layers	2
Neurons in first hidden layer	3
Neurons in second hidden layer	2
Activation function	Linear
Initial weights	[−0.1~0.1]
Learning rate	0.7

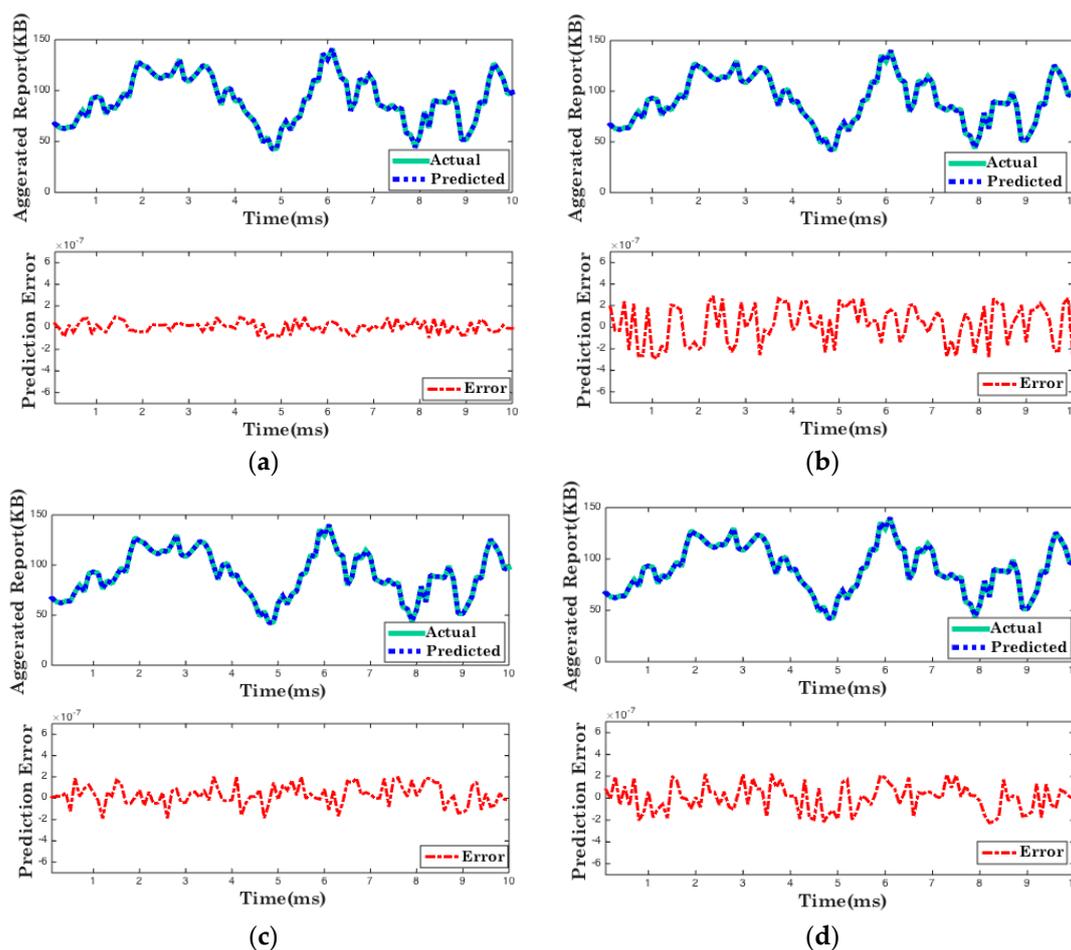
**Table 3.** FANN training parameters' settings.

Parameters	Details
Number of samples used in learning	12
Number of samples used for training	9
Number of samples used for testing	3
Number of sample that FANN predict	1
Time used for learning process	165 $\mu$ s
Time used for prediction	10 $\mu$ s
Desired training error	$10^{-7}$

For training the FANN module we used a back propagation algorithm that updates the weights of the neural network using a gradient descent algorithm. In order to accelerate the training process,

we modified the back propagation algorithm by adding the adaptive learning rate and momentum features to gradient descent algorithm, as in [22,23]. The algorithmic computational complexity of gradient descent with such features can be found in [24]. For the back propagation learning algorithm we used a linear activation function for the neuron in the output layer to produce a linear output value similar to that given during the training of the network. Also, we used the default initial weights and learning rate of the neural network library [22] for the back propagation algorithm.

Figure 6 shows the performance evaluation of the prediction accuracy of FANN module when performing an online learning and prediction process over a training dataset containing ONU reports collected at the OLT during a period of 10 ms. In Figure 6a, the upper plot shows the actual aggregated ONU report in Kbytes (solid) and the predicted aggregated ONU report (dashed). The bottom plot shows the prediction error (i.e., differences between the values of the actual ONU reports and the predicted reports over the time). From this figure it can be seen that training FANN with the model settings given in Table 2 can achieve a prediction error very close to the desired error used during the learning process (i.e.,  $(-1\sim 1) * 10^{-7}$ ).



**Figure 6.** The prediction accuracy performance of FANN module trained over a period of 10 ms with different number of layers and hidden neurons: (a) Two hidden layers with three neurons in first layer and two in second layer; (b) one hidden layer with two neurons; (c) two hidden layers two neurons on each layer; (d) three hidden layers with four neurons in the first layer and two in the second layer.

The FANN model settings given in Table 2 are obtained by applying the model selection method reported in [25]. This method is based on constructing a neural network model with a different number of hidden layers and neurons and evaluating the performance of the network in terms of prediction

error; then, choosing the model settings that achieve prediction error close to the desired error required by the problem we are going to solve. In our simulation experiment, to evaluate the appropriate model setting for our prediction problem, we started by selecting a small number of hidden layers and neurons and evaluating the performance of the network using them; then we increased the number and observed the prediction error change. We began with one hidden layer and two neurons and noticed that the model failed to converge to the desired error (see the prediction error in Figure 6b). Then, we increased the number of hidden layers to two, with two neurons in each layer, and noticed an improvement in the prediction error toward the desired error (see Figure 6c). After that, we kept the number of hidden layers to two and increased the number of neurons in the first hidden layer to three; then, we obtained the prediction error as depicted in Figure 6a. We again noticed an increase in the prediction error when we further increased the number of neurons in the first hidden layer to four, as shown in Figure 6d. For this reason, we adopted the model settings given in Table 2 to conduct the other simulation experiments in this paper.

Figure 7 shows the performance evaluation of three algorithms in terms of upstream delay, utilization, packet loss rate, and jitter. Figure 7a shows the delay performance comparison between the three algorithms. As we can see, Optimized-RR outperforms the other two algorithms in terms of the upstream delay performance. We can see from this figure that Optimized-RR DBA shows a superior delay performance as compared to the gGIANT. This is due to the efficient utilization of XGS-PON upstream bandwidth by Optimized-RR DBA, because of allowing ONUs to adjust their transmission window to accommodate the remainder of the unallocated portion of the upstream frames during every allocation cycle. From the same figure, we notice that Optimized-RR DBA attained a delay performance in the range of 290 to 300  $\mu\text{s}$  when per-ONU/RRU traffic load ranged from 903 to 922 Mbps (i.e., the required front-haul uplink capacity when considering PHY split-based small cell [26]), while gGIANT shows a delay performance higher than 300  $\mu\text{s}$  for the same range of traffic loads. When it comes to the Adaptive-Learning DBA, we can see that it attains a delay performance in the range of 200 to 205  $\mu\text{s}$  for the ONU/RRU traffic load ranging from 903 to 922 Mbps, which is not only much lower than the other two algorithms but also satisfies the latency requirement for mobile front-haul recommended by the next-generation mobile networks alliance (NGMN) i.e., 250  $\mu\text{s}$ . The low-latency performance achieved by Adaptive-Learning DBA is because of eliminating one-way propagation delay for receiving the buffer occupancy report from ONU as well as the waiting time for DBA processing at OLT, as illustrated in Figure 8a,b. We can see from Figure 8a that for Adaptive-Learning DBA the end-to-end delay budget includes (Queuing delay at ONU ( $T_q$ ) + one-way propagation delay for sending the grant from OLT to ONU ( $T_g$ ) + one-way propagation delay for transmitting the data from ONU to OLT ( $T_t$ )). However, for gGIANT and Optimized-RR DBA, as we can see from Figure 8b, on top of the delay budget calculated in the case of Adaptive-Learning DBA, there is a one-way propagation delay for sending the reports from ONU to OLT ( $T_r$ ) in addition to the DBA waiting time ( $T_d$ ) that must be included in the calculation.

Figure 7b shows the utilization performance comparison between the three algorithms. From this figure, we can see that the highest utilization performance is achieved by Adaptive-Learning DBA. This is for two reasons: one, the faster grant assignment process that can be achieved by Adaptive-Learning DBA, which allows ONUs to effectively utilize XGS-PON upstream bandwidth compared to the two other DBAs; two, the Adaptive-Learning DBA also inherits the property of dynamic adjusting of the ONU pre-determined limit, the same as Optimized-RR DBA. This allows ONUs to efficiently exploit any unallocated remainder of the upstream frames, improving the overall XGS-PON upstream link utilization.

Figure 7c shows the packet loss ratio performance comparison of the three algorithms. As we can see, Optimized-RR DBA achieves a lower packet loss ratio compared to gGIANT. This is due to the reduction in the network congestion because of allowing ONUs to use a wider transmission window to transmit their data, and this minimizes the probability of dropping the packets from the ONU buffer. From the same figure it can be clearly seen that our new adaptive-learning-based DBA

performs better than Optimized-RR and provides a lower packet loss rate. The reason for this is that the adaptive-learning-based DBA increases the speed of serving ONU. In other words, it clears the ONU buffer data much faster than optimized-RR, providing a significant reduction in network congestion and waiting time for the packets to get served.

Figure 7d shows the jitter performance comparison of the three algorithms. From this figure, we can see that the Adaptive-Learning DBA achieves the lowest jitter performance in comparison with the other two DBAs. This is due to a significant reduction in the network congestion, as the variation of delay between two consecutive packets is highly affected by the network congestion.

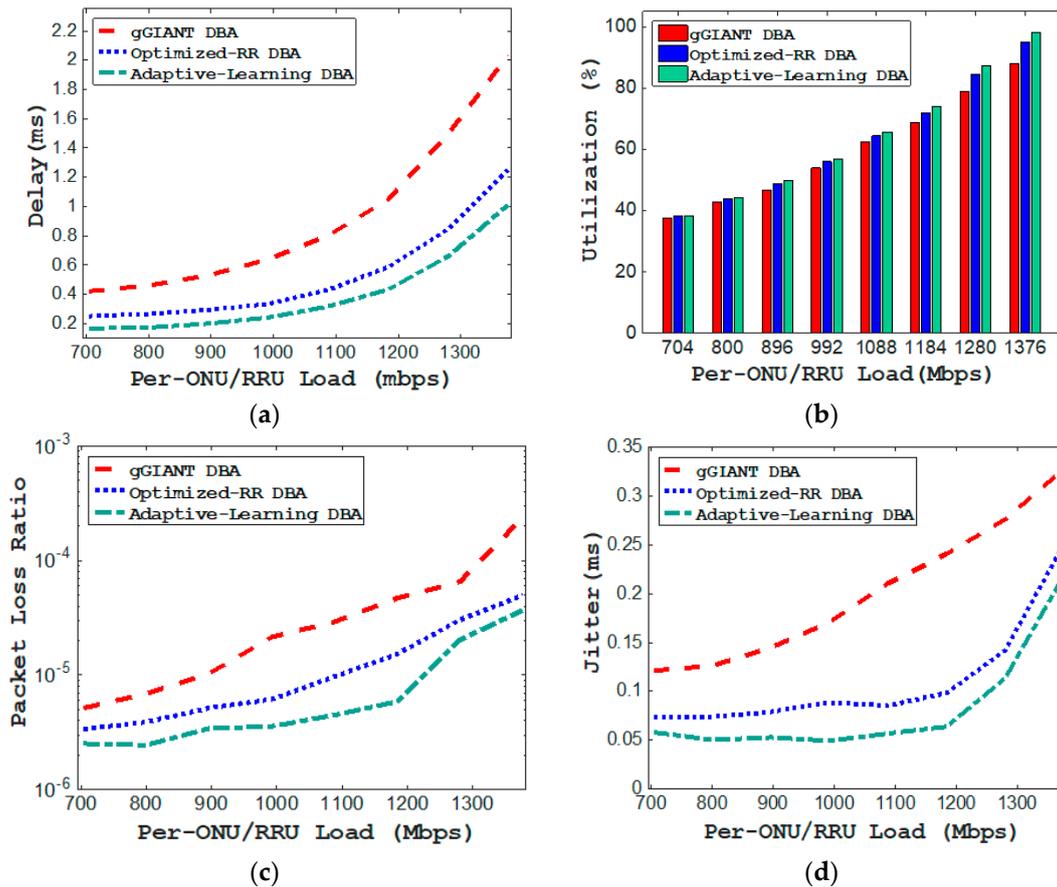


Figure 7. The performance evaluation of three algorithms: (a) The delay performance; (b) the utilization performance; (c) the packet loss performance; (d) the jitter performance.

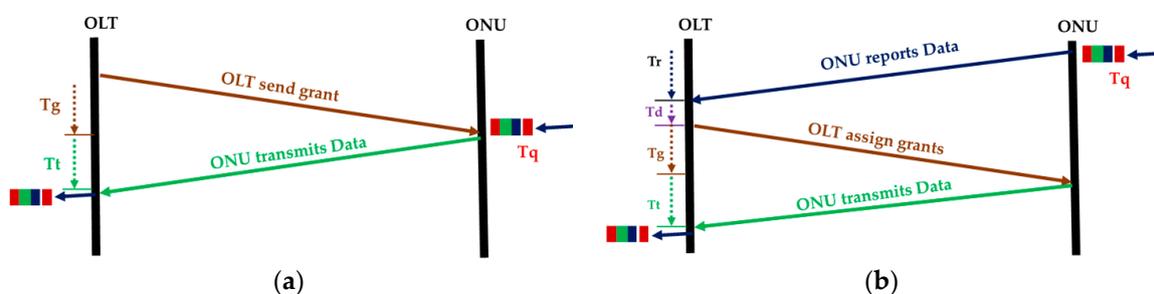


Figure 8. End-to-end packet delay calculation: (a) gGIANT and Optimized-RR DBA; (b) Adaptive-Learning DBA.

## 5. Related Works and Discussion

In a 5G based on the dense deployment of small-cell C-RAN architecture, a huge number of low-latency fiber-based front-haul connections (e.g., dark fiber [10]) are required to connect small-cell RRUs residing in different locations to the BBU pool or cloud. This increases the deployment cost of small cell in C-RAN architecture. Front-haul transport network sharing is a potential method to address C-RAN deployment cost issues in the 5G era. A variety of front-haul transport solutions that support network sharing have been proposed in the literature. The main proposed solutions are: Passive wavelength-division multiplexing (WDM) [27], WDM/optical transport network (OTN) [28], WDM/PON [8], and Ethernet [29]. This paper considers low-cost shared front-haul transport solutions based on TDM-PONs, which are classified as a type of WDM/PON solution. In particular, the paper focuses on optimizing XGS-PON networks to meet the latency requirement of C-RAN while maintaining a high degree of bandwidth efficiency. The main contribution of this paper is introducing a traffic prediction bandwidth allocation solution to overcome the latency and bandwidth efficiency problems of mobile front-haul in C-RAN. This solution takes advantage of the nature of the scheduling process in an LTE network to formulate the traffic prediction bandwidth allocation problem; then, it applies a machine leaning tool to solve this problem.

Machine learning tools have been widely used for traffic prediction in passive optical network and other networking contexts. For example, in [30] a traffic prediction method employing a least mean square (LMS) algorithm was utilized to reduce the burst delay at the edge nodes in optical packet switching networks. In [31] a nonlinear auto-regressive (NAR) neural network is implemented for forecasting video bandwidth requirements (i.e., grant requests) to improve the efficiency of the DBA algorithm in Ethernet passive optical networks (EPONs) to support high-definition video transmission. In [32] several traffic estimation methods such as Gaussian approximation, Courcoubetis approximation and leaky bucket (LB) algorithms have been proposed to optimize PON networks to transport mobile base stations' backhaul traffic with a high degree of bandwidth utilization and a QoS guarantee. In [3] a traffic estimation method utilizing simple statistical traffic analysis function is used as a dynamic bandwidth allocation method to accommodate the bursty nature of the front-haul traffic in TDM-PON-enabled C-RAN architecture. To the best of our knowledge, this is the first time a neural network-based traffic prediction approach has been employed to deal with the front-haul latency issue in C-RAN. Fortunately, our simulation results show that the neural network-based traffic prediction approach not only satisfies the delay requirement of mobile front-haul for C-RAN architecture but also improves other network performance metrics such as upstream utilization, jitter, and packet losses. Although in our current proposal we consider the XGS-PON system (i.e., a single-wavelength PON), in our future work we will consider more complex PON networks (i.e., multi-wavelength networks such as WDM-PON and TWDM-PON), taking into consideration the physical non-linearity of the system described in [33,34].

## 6. Conclusions

In this paper we propose a novel method for predicting the future amount of buffer occupancy of ONU based on the historical set of buffer occupancy reports collected during BBU processing. Based on the amount of ONU buffer occupancy predicted by the aforementioned method, the OLT allocates the scheduling grant in advance to the ONU eliminating the additional DBA processing delay and the delay required for reporting the ONU buffer occupancy to the OLT. We formulated the problem of predicting the future ONU report as a function approximation, and for the first time we applied an adaptive learning approach to solve it. The experimental results showed that the proposed method can satisfy the minimum delay requires for mobile front-haul and significantly improves XGS-PON utilization, jitter, and packet loss performance.

**Author Contributions:** A.M.M. Implemented the simulation model in NS-3 and wrote the paper. W.H. supervised the project, revised the paper, and provided the funding. S.B.H. and A.S. contributed to the results analysis and reviewed the paper, providing very useful comments.

**Funding:** This work was supported by the National Natural Science Foundation of China (NSFC) (61431009, 61371082, and 61521062) and the National Science and Technology Major Project of the Ministry of Science and Technology of China (2015ZX03001021).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Kuwano, S.; Terada, J.; Yoshimoto, N. Operator perspective on next-generation optical access for future radio access. In Proceedings of the 2014 IEEE International Conference on Communications Workshops (ICC), Sydney, NSW, Australia, 10–14 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 376–381.
2. Hatta, S.; Tanaka, N.; Sakamoto, T. Feasibility demonstration of low latency dba method with high bandwidth-efficiency for tdm-pon. In Proceedings of the 2017 Optical Fiber Communications Conference and Exhibition (OFC), Los Angeles, CA, USA, 19–23 March 2017; Optical Society of America: Washington, DC, USA, 2017; p. M3I.2.
3. Kobayashi, T.; Ou, H.; Hisano, D.; Shimada, T.; Terada, J.; Otaka, A. Bandwidth allocation scheme based on simple statistical traffic analysis for tdm-pon based mobile fronthaul. In Proceedings of the 2016 Optical Fiber Communications Conference and Exhibition (OFC), Anaheim, CA, USA, 20–24 March 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–3.
4. Tashiro, T.; Kuwano, S.; Terada, J.; Kawamura, T.; Tanaka, N.; Shigematsu, S.; Yoshimoto, N. A novel dba scheme for tdm-pon based mobile fronthaul. In Proceedings of the 2014 Optical Fiber Communications Conference and Exhibition (OFC), San Francisco, CA, USA, 9–13 March 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–3.
5. Nomura, H.; Ou, H.; Shimada, T.; Kobayashi, T.; Hisano, D.; Uzawa, H.; Terada, J.; Otaka, A. First demonstration of optical-mobile cooperation interface for mobile fronthaul with tdm-pon. *IEICE Commun. Express* **2017**, *6*, 375–380. [[CrossRef](#)]
6. Zhou, S.; Liu, X.; Effenberger, F.; Chao, J. Low-latency high-efficiency mobile fronthaul with tdm-pon (mobile-pon). *J. Opt. Commun. Netw.* **2018**, *10*, A20–A26. [[CrossRef](#)]
7. Alvarez, P.; Marchetti, N.; Payne, D.; Ruffini, M. Backhauling mobile systems with xg-pon using grouped assured bandwidth. In Proceedings of the 2014 19th European Conference on Networks and Optical Communications-(NOC), Milano, Italy, 4–6 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 91–96.
8. Mikaeil, A.M.; Hu, W.; Ye, T.; Hussain, S.B. Performance evaluation of xg-pon based mobile front-haul transport in cloud-ran architecture. *IEEE/OSA J. Opt. Commun. Netw.* **2017**, *9*, 984–994. [[CrossRef](#)]
9. Alliance, N. *Further Study on Critical C-Ran Technologies*; Next Generation Mobile Networks: Frankfurt, Germany, 2015.
10. Eramo, V.; Listanti, M.; Lavacca, F.G. Iovanna, Dimensioning Models of Optical WDM Rings in Xhaul Access Architectures for the Transport of Ethernet/CPRI Traffic. *Appl. Sci.* **2018**, *8*, 612. [[CrossRef](#)]
11. Zainuddin, Z.; Pauline, O. Function approximation using artificial neural networks. *WSEAS Trans. Math.* **2008**, *7*, 333–338.
12. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [[CrossRef](#)]
13. Nikaiein, N. Processing radio access network functions in the cloud: Critical issues and modeling. In Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services, Paris, France, 11 September 2015; ACM: New York, NY, USA, 2015; pp. 36–43.
14. Jain, A.K.; Mao, J.; Mohiuddin, K.M. Artificial neural networks: A tutorial. *Computer* **1996**, *29*, 31–44. [[CrossRef](#)]
15. Koskela, T.; Lehtokangas, M.; Saarinen, J.; Kaski, K. Time series prediction with multilayer perceptron, fir and elman neural networks. In Proceedings of the World Congress on Neural Networks, San Diego, CA, USA, 15–18 September 1996; INNS Press: San Diego, CA, USA, 1996; pp. 491–496.
16. Wu, X.; Brown, K.N.; Sreenan, C.J.; Alvarez, P.; Ruffini, M.; Marchetti, N.; Payne, D.; Doyle, L. An xg-pon module for the ns-3 network simulator. In Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques, Cannes, France, 5–7 March 2013; Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering (ICST): Brussels, Belgium, 2013; pp. 195–202.

17. Riley, G.F.; Henderson, T.R. The ns-3 network simulator. In *Modeling and Tools for Network Simulation*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 15–34.
18. Skubic, B.; Chen, J.; Ahmed, J.; Wosinska, L.; Mukherjee, B. A comparison of dynamic bandwidth allocation for epon, gpon, and next-generation tdm pon. *IEEE Commun. Mag.* **2009**, *47*, S40–S48. [[CrossRef](#)]
19. Ammar, D.; Begin, T.; Guerin-Lassous, I. A new tool for generating realistic internet traffic in ns-3. In Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques, Barcelona, Spain, 21–25 March 2011; Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering (ICST): Brussels, Belgium, 2011; pp. 81–83.
20. Paul, U.; Subramanian, A.P.; Buddhikot, M.M.; Das, S.R. Understanding traffic dynamics in cellular data networks. In Proceedings of the 2011 Proceedings IEEE INFOCOM, Shanghai, China, 10–15 April 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 882–890.
21. Nissen, S. Implementation of a fast artificial neural network library (fann). *Rep. Dep. Comput. Sci. Univ. Cph. (DIKU)* **2003**, *31*, 29.
22. Demuth, H.; Beale, M. *Neural Network Toolbox for Use with Matlab—User’s Guide Verion 3.0*; The MathWorks, Inc.: Natick, MA, USA, 1993.
23. Saduf, A.; Wani, M. Comparative study of adaptive learning rate with momentum and resilient back propagation algorithms for neural net classifier optimization. *Int. J. Distrib. Cloud Comput.* **2014**, *2*, 1–6.
24. Leung, W.K. On the Complexity of Backpropagation with Momentum and Gradient Descent on Sigmoidal Steepness. In Proceedings of the WSES 2001 International Conference of Neural Network and Applications, Tenerife, Spain, 12–14 February 2001.
25. Panchal, G.; Ganatra, A.; Kosta, Y.; Panchal, D. Review on methods of selecting number of hidden nodes in artificial neural network. *Int. J. Comput. Theory Eng.* **2011**, *3*, 332–337. [[CrossRef](#)]
26. Virtualization, S.C. Functional Splits and Use Cases. In *Small Cell Forum Release*; Small Cell Forum: Gloucestershire, UK, 2016.
27. Alimi, I.A.; Teixeira, A.L.; Pereira Monteiro, P. Towards an Efficient C-RAN Optical Fronthaul for the Future Networks: A Tutorial on Technologies, Requirements, Challenges, and Solutions. *IEEE Commun. Surv. Tutor.* **2017**, *99*, 1–63. [[CrossRef](#)]
28. Eramo, V.; Listanti, M.; Sabella, R.; Testa, F. Definition and Performance Evaluation of a Low-Cost/High-Capacity Scalable Integrated OTN/WDM Switch. *IEEE J. Opt. Commun. Netw.* **2012**, *4*, 1033–1045. [[CrossRef](#)]
29. Valcarenghi, L.; Kondepu, K.; Castoldi, P. Time-Versus Size-Based CPRI in Ethernet Encapsulation for Next Generation Reconfigurable Fronthaul. *IEEE J. Opt. Commun. Netw.* **2017**, *8*, D64–D73. [[CrossRef](#)]
30. Daly, C.; Moore, D.L.; Haddad, R.J. Nonlinear auto-regressive neural network model for forecasting hi-def h. 265 video traffic over ethernet passive optical networks. In Proceedings of the SoutheastCon 2017, Charlotte, NC, USA, 30 March–2 April 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–7.
31. Karamitsos, I.; Bowerman, C. A resource reservation protocol with linear traffic prediction for obs networks. *Adv. Opt. Technol.* **2013**, *2013*, 425372. [[CrossRef](#)]
32. Kosmatos, E.; Orphanoudakis, T.; Tsampassis, E.; Leligou, H. Traffic estimation in a pon-based mobile backhaul environment. In Proceedings of the 17th WSEAS International Conference on Communications (CSCC’13), Rhodes Island, Greece, 16–19 July 2013.
33. Maher, R.; Xu, T.; Galdino, L.; Sato, M.; Alvarado, A.; Shi, K.; Savory, S.J.; Thomsen, B.C.; Killely, R.I.; Bayvel, P. Spectrally shaped dp-16qam super-channel transmission with multi-channel digital back-propagation. *Sci. Rep.* **2015**, *5*, 8214. [[CrossRef](#)] [[PubMed](#)]
34. Liga, G.; Xu, T.; Alvarado, A.; Killely, R.I.; Bayvel, P. On the performance of multichannel digital backpropagation in high-capacity long-haul optical transmission. *Opt. Express* **2014**, *22*, 30053–30062. [[CrossRef](#)] [[PubMed](#)]

