Traffic estimation based on long short-term memory neural network for mobile front-haul with XG-PON

Min Zhang (张 敏)[1], Bo Xu (许 渤)[1,*], Xiaoyun Li (栗晓云)[2], Yi Cai (蔡 怡)[1], Baojian Wu (武保剑)[1], and Kun Qiu (邱 昆)[1]

[1]*Key Laboratory of Optical Fiber Sensing and Communications, Ministry of Education, University of Electronic Science and Technology of China, Chengdu 611731, China*

[2]*Business School, University of International Business and Economics, Beijing 100029, China*

**Corresponding author: xubo@uestc.edu.cn*

Received January 28, 2019; accepted April 12, 2019; posted online June 25, 2019

A novel predictive dynamic bandwidth allocation (DBA) method based on the long short-term memory (LSTM) neural network is proposed for a 10-gigabit-capable passive optical network in mobile front-haul (MFH) links. By predicting the number of packets that arrive at the optical network unit buffer based on LSTM, the round-trip time delay in traditional DBAs can be eliminated to meet the strict latency requirement for MFH links. Our study shows that the LSTM neural network has better performance than feed-forward neural networks. Based on extensive simulations, the proposed scheme is found to be able to achieve the latency requirement for MFH and outperforms the traditional DBAs in terms of delay, jitter, and packet loss ratio.

OCIS codes: 060.4250, 060.4510.

doi: 10.3788/COL201917.070603.

The cloud radio access network (C-RAN) is one of the key technologies for fifth generation mobile communication (5G)[1–3]. In a C-RAN, the digital baseband processing units (BBUs) are moved from mobile base station sites to a central location known as the BBU pool that serves a group of distributed radio units known as remote radio heads (RRHs)[1]. Mobile front-haul (MFH) is an optical link that connects RRHs in multiple locations to the BBU pool[2]. The number of MFH links is expected to increase with the increasing traffic requirements of the 5G systems. To reduce the MFH link cost, a time division multiplexed passive optical network (TDM-PON) is proposed as it allows sharing of optical fibers and transmission equipment[2,3]. However, a TDM-PON suffers from a large latency for forwarding uplink traffic because an optical network unit (ONU) has a waiting time of several milliseconds in a typical dynamic bandwidth allocation (DBA) scheme[3]. This transmission waiting time in the ONU is a critical problem since the latency requirement for the MFH link is very strict, e.g., less than 250 μs defined by the Third Generation Partnership Project (3GPP)[4].

Different methods have been proposed in the literature to solve the latency issue of TDM-PON[5–9]. For example, fixed bandwidth allocation (FBA) is used to meet the latency requirement for MFH links[3]. However, the bandwidth usage efficiency is low and the number of ONUs that can be accommodated is limited by the FBA algorithm[3,5]. Instead, a statistical DBA scheme has been proposed to improve the bandwidth usage efficiency[7]. A disadvantage of the statistical DBA is that it cannot deal with burst of MFH traffic. Reference [8] evaluated the performance of group-assured GIANT (gGIANT) and round-robin DBA (RR-DBA). Results show that neither RR-DBA nor gGIANT satisfies the delay requirement for MFH, and RR-DBA has a lower upstream delay than gGIANT.

Machine learning (ML), a branch of artificial intelligence (AI), is regarded as one of the most promising methodological approaches to performing different types of network data analysis for automated network self-configuration[10]. Recently, machine learning techniques have been successfully applied in optical communication and optical networks to improve the intelligence of such systems[10–12]. With its powerful modeling capabilities, artificial intelligence is also desirable to help solving the latency issue of TDM-PON for MFH applications.

In this Letter, we propose a long short-term memory (LSTM)-based predictive DBA method for a slow-latency 10-gigabit-capable passive optical network (XG-PON) MFH for a C-RAN based on traffic estimation in Fig. 1. First, the problem of predicting the number of packets that arrive at the ONU is formulated as an ML function approximation problem. Second, LSTM is investigated for this problem and compared to a feed-forward neural network (FNN). Results show that the LSTM neural network has better prediction performance than FNN. Finally, the

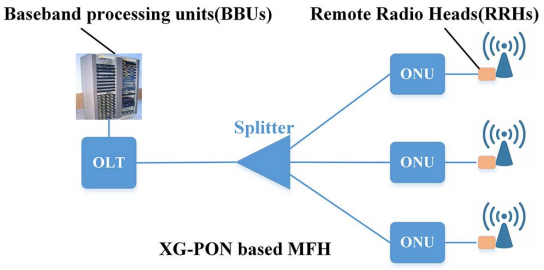

Fig. 1. MFH architecture based on the XG-PON system.

LSTM-based DBA for XG-PONs is extensively studied in terms of delay, jitter, and packet loss ratio. It is verified that the LSTM-DBA outperforms the traditional DBA, such as RR-DBA and FNN-DBA. In the proposed LSTM-based DBA, the number of packets that arrive at the ONU buffer from the RRHs is predicted using an LSTM recurrent neural network. Based on the accurate prediction results, the DBA module at the optical line terminal (OLT) can grant bandwidth without waiting for the ONU buffer occupancy report to the OLT. A low latency can thus be achieved to meet the latency requirement (250 μs[4]) for MFH links.

Recurrent neural network (RNN) is a learning method in the fields of ML that was initially proposed for sequence prediction problems and has gained a lot of attention in recent years[13–15]. An RNN introduces a recurrent structure for implementing a memory mechanism to keep track of past information. Through this recurrent structure, the RNN has better performance than the FNN in the prediction analysis of data with time series. However, the RNN is affected by the gradient exploding or gradient vanishing[13,14] that prevents a complete learning of the time series. Due to this issue, we investigate the LSTM RNNs[14], which is a variant of RNN with neurons replaced by cells, for more accurate traffic prediction in this work.

In the following, R_t^j stands for the buffer occupancy report for the tth cycle sent from the jth ONU to the OLT, and D_t^j stands for the number of packets received in the tth cycle by the OLT from the jth ONU. Clearly, these two time series reports have important information on the working status of each ONU. Assuming that X_t^j is the number of packets that arrive at the jth ONU from the connected RRHs in the tth cycle, X_t^j can be calculated using Eq. (1), where $R_{t-1}^j - D_{t-1}^j$ is the remaining packets in the jth ONU after sending the upstream frame in the $(t-1)$th cycle:

$$X_t^j = R_t^j - (R_{t-1}^j - D_{t-1}^j). \tag{1}$$

If X_{t+1}^j can be known in advance, then the DBA algorithm can allocate corresponding resources to the ONU with reduced waiting time for latency minimization. However, in practice, we can only estimate X_{t+1}^j by some estimate function $\hat{X}_{t+1}^j = f(X_{t-K+1}^j, X_{t-K+2}^j, ..., X_t^j)$. From the ML perspective, this estimation problem can be regarded as a sequence forecasting problem and can be solved using the LSTM neural network model in Fig. 2. The time series sequence $X_{t-K+1}^j, X_{t-K+2}^j, ..., X_t^j$ is the input to the neural network, with K equal to 128 in this work. Simulation results showed that this value of K could achieve a good balance between estimation accuracy and computational complexity. The model includes three hidden layers: LSTM layer (64 cells, dropout with $p = 0.2$) and two fully connected (FC) layers with 64 and 16 neurons, respectively. A single neuron is used for the output layer as the result for the predicted number of packets to arrive at the jth ONU. In the choice of the number of hidden layers and neurons, both accuracy and speed of convergence should be considered. After verification of various parameters, the above parameters are found to be able to achieve good results.

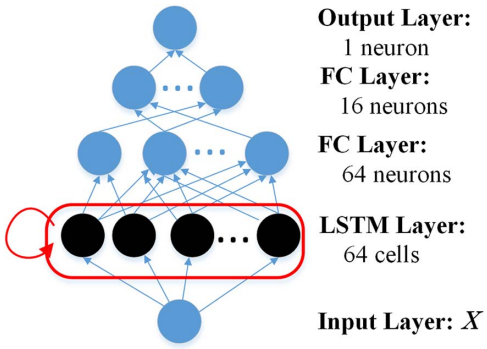

Fig. 2. LSTM neural network architecture used in the proposed method.

The structure of the LSTM cell is shown in Fig. 3, with details shown in Fig. 4. The time series sequence $X_{t-K+1}, X_{t-K+2}, ..., X_{t-1}, X_t$ is used as the input to the layer and the output is denoted as another sequence $h_{t-K+1}, h_{t-K+2}, ..., h_{t-1}, h_t$. For one input sequence at time t, each cell will be recursively used for K times as unfolded in time in Fig. 3 with one element X_{t-K+k} at a time and $k \in \{1, 2, ..., K\}$. The kth output h_{t-K+k} and the kth cell state called C_{t-K+k} are used together with $(k+1)$th input $X_{t-K+k+1}$ for computing the next output and cell state, as shown in Fig. 4. The output sequence $h_{t-K+1}, h_{t-K+2}, ..., h_{t-1}, h_t$ is ready for the next layer after h_t is computed.

One of the advantages of LSTM is its ability to remove or add information to the cell state by a gate structure. There are three gates in each cell, input gate, forget gate, and output gate that can optionally let information pass

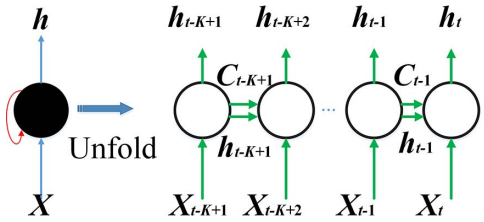

Fig. 3. LSTM cell and its unfolding in time.

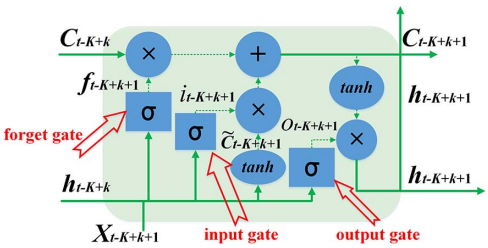

Fig. 4. Detailed structure of the LSTM cell memory block. The sigmoid function is denoted as σ and the pointwise multiplication is denoted as $\times$ in the figure.

through. Each gate is composed of a sigmoid function ("σ" in Fig. 4) and a pointwise multiplication operation ("$\times$" in Fig. 4). The mathematical models for the three gates are shown from Eq. (2) to Eq. (7), where W is a weight vector and b is the bias:

$$f_{t-K+k+1} = \sigma(W_f \cdot [h_{t-K+k}, X_{t-K+k+1}] + b_f), \quad (2)$$

$$i_{t-K+k+1} = \sigma(W_i \cdot [h_{t-K+k}, X_{t-K+k+1}] + b_i), \quad (3)$$

$$\tilde{C}_{t-K+k+1} = \tanh(W_C \cdot [h_{t-K+k}, X_{t-K+k+1}] + b_C), \quad (4)$$

$$C_{t-K+k+1} = f_{t-K+k+1} \times C_{t-K+k} + i_{t-K+k+1} \times \tilde{C}_{t-K+k+1}, \quad (5)$$

$$o_{t-K+k+1} = \sigma(W_0 \cdot [h_{t-K+k}, X_{t-K+k+1}] + b_0), \quad (6)$$

$$h_{t-K+k+1} = o_{t-K+k+1} \times \tanh(C_{t-K+k+1}). \quad (7)$$

First, the forget gate in Fig. 4 is used to decide which information would be thrown away from the cell state. This gate uses a sigmoid function with an input of h_{t-K+k} representing the output of the previous cell, $X_{t-K+k+1}$ representing the input of the current cell, and output $f_{t-K+k+1}$ as in Eq. (2). The output is equal to 1, which represents "completely keep this," while 0 represents "completely get rid of this." Second, the input gate decides which values will be updated in this cell. As shown in Fig. 4, the sigmoid layer will decide the values to be updated, i.e., the result calculated using Eq. (3), and the new candidate value $\tilde{C}_{t-K+k+1}$ calculated in Eq. (4) using the tanh function. As described in Eq. (5), the state of the cell is computed based on the output of the forget gate and the input gate. Finally, the cell has to decide the output value by the output gate. As shown in Fig. 4, the next output $h_{t-K+k+1}$ is computed using the output gate as given in Eq. (6) and Eq. (7).

In order to obtain accurate prediction results on the number of arriving packets at the ONU, 8000 samples collected in a period of 1 s are used for LSTM RNN training and updating. Note that an ONU sends its buffer occupancy report in a cycle of 125 μs. During training, samples are divided into a training part (70%) and a validation part (30%). The back propagation through time (BPTT) algorithm[14] is used for training. In this work, mean square error (MSE) is used for loss function following Ref. [16]. By computing the partial derivatives of the outputs, weights, and input values of hidden layers, the network can move backward to trace the error between the real output values and the predicted output values. The weights are updated using the gradient descent method in order to reduce the prediction errors. After training, the neural network is used for real-time predicting with low complexity[16].

Each DBA cycle contains one prediction phase and one grant assignment phase, as shown in Algorithm 1. $\hat{X}^j_{t+1}$ (the number of packets that will arrive at the ONU in the next cycle) is predicted by the LSTM, as mentioned above in the prediction phase. In the grant assignment phase, a pre-grant assignment $\hat{G}^j_{t+1}$ is computed as the sum of the predicted number of arriving packets $\hat{X}^j_{t+1}$ plus the remaining packets in the ONU buffer $R^j_t - D^j_t$. In the case that the sum of the pre-grant assignment packets is in excess of the max upstream bandwidth (MUB) of the cycle, the maximum of $\hat{G}^j_{t+1}$ is reduced by one ($\hat{G}^j_{t+1} - -$) until the sum of the pre-grant assignment packets can be accommodated by the MUB, as shown from step 6 to step 8 in Algorithm 1.

Algorithm 1: LSTM Based-DBA

Input: time series reports 1: $\{R_{t-k+1}, R_{t-k+2}, \ldots, R_{t-1}, R_t\}$
time series reports 2: $\{D_{t-k+1}, D_{t-k+2}, \ldots, D_{t-1}, D_t\}$

Output: Grant Assignment G^j_{t+1} for the next cycle

1: Get time series reports 3 $\{X_{t-k+1}, X_{t-k+2}, \ldots, X_t\}$ by Eq. (1)
2: **For** each ONU **do** — Prediction
3: $\hat{X}^j_{t+1} = f(X^j_{t-K+1}, X^j_{t-K+2}, \ldots, X^j_t)$
4: **End for**
5: Pre-Grant Assignment $\hat{G}^j_{t+1} = \hat{X}^j_{t+1} + (R^j_t - D^j_t)$ — Grant Assignment
6: **While** $(\sum_j \hat{G}^j_{t+1} > MUB)$
7: $\max(\hat{G}^j_{t+1}) - -$
8: **End while**
9: $G^j_{t+1} = \hat{G}^j_{t+1}$

In order to verify the performance of the proposed LSTM-based prediction method, simulations are conducted to analyze its performance and compared with the FNN-based method from Ref. [9]. The sizes of the FNN are configured as 128/512/64/16/1 for input-layer/hidden-layer1/hidden-layer2/hidden-layer3/output-layer with best system performance. In the simulations, the Poisson Pareto burst process (PPBP) model, with main parameters shown in Table 1, is used to generate the upstream data traffic. The PPBP model is chosen as it matches the statistical properties of real-life network traffic[17]. The number of packets arriving at the ONU (equal to the number of packets produced by the PPBP) X^j_t is collected every 125 μs and saved in the dataset. In this simulation, the simulation time in different Poisson arrival rates is 1 s. In the training phase, the data in the dataset was divided into two parts, the training part

Table 1. Traffic Simulation Parameters

Parameters	Values
Number of bursts	5000
Mean burst time length	2 ms
Poisson arrival rate	[95, 110, 125–200 Mbps]
Hurst parameter	0.8
Pareto shape parameter	1.4
Packet size	1470 bytes

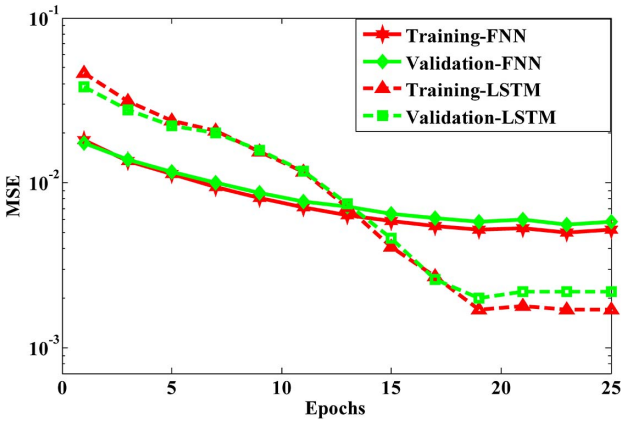

Fig. 5. MSE in the training process.

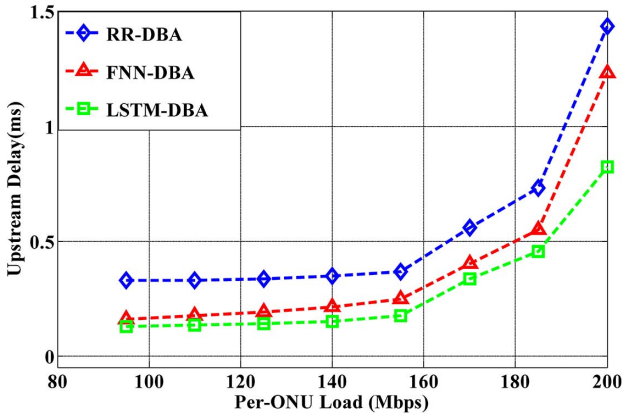

Fig. 6. Upstream delay performance comparison of RR-DBA, FNN-DBA, and LSTM-DBA.

and the validation part. The training part is used to train the LSTM RNN or FNN, and the other data is used for validation. Figure 5 shows the MSE curves between the predicted and the real number of packets that arrived as the number of epochs increases in the process of training. The results show that the LSTM can get more accurate prediction results than the FNN as long as enough epochs are included in the training.

Furthermore, the proposed LSTM-based DBA algorithm is evaluated by numerical simulations, with NS-3[18] using the XG-PON module[19] in terms of the average packet latency, jitter, and packet loss ratio of the traffic. The total number of ONUs in the XG-PON is assumed to be 10, in which 10 small-cell RRHs are connected to 10 ONUs following Ref. [9] (assuming that each ONU had only one T-CONT). The buffer size of the ONU (i.e., T-CONT) is 1 Mbytes. The roundtrip propagation delay (RTT) is 100 μs, which can cover up to a 10 km front-haul distance. The parameters for the PPBP model to simulate the burst of the front-haul data traffic in the uplink transmission are the same as in Table 1. The maximum transfer data rate for the PPBP traffic is set to 2.048 Gbps. Table 2 summarizes the abovementioned simulation parameters.

The upstream delay is one of the most important parameters for real-time applications. Figure 6 shows the average upstream delay performance comparison among the three DBA algorithms. The upstream delay is computed as the difference in time between the arrival time of a certain packet to the T-CONT buffer at the ONU side and the arrival time of the same packet at the OLT side, including a one-way propagation delay of 50 μs for packet transmission, as in Ref. [8]. As we can see, the LSTM-DBA outperforms the other two algorithms from low load to high load. From Fig. 6, the conventional RR-DBA is found to fail to satisfy the delay requirement of 250 μs for MFH links when the per-ONU (RRH) load exceeds 95 Mbps.

Table 2. DBA Simulation Parameters

Parameters	Values
Application traffic model	PPBP
Simulation time	10 s
Max polling interval (all DBAs)	125 μs
Number of RRHs (ONUs)	10
T-CONT per ONU	1
Roundtrip propagation delay	100 μs
ONU queue size (T-CONT buffer)	1 Mbytes

Figure 6 also shows that the FNN-DBA obtained an average delay less than 250 μs until the per-ONU/RRU traffic load is increased to 140 Mbps. While for the LSTM-DBA, not only does it have a lower average upstream delay, but also its delay performance can still meet the requirement until the pre-ONU/RRH traffic load is increased up to 160 Mbps. Compared with the RR-DBA, the low-latency performance achieved by the other two DBAs is due to the elimination of the one-way propagation delay for receiving the buffer occupancy report from the ONU as well as the waiting time for the DBA processing at the OLT. The 'memory' of the LSTM is more suitable for the data arrival prediction than the FNN-DBA and helps to further reduce the latency.

Figure 7 shows the jitter performance comparison among the three DBA algorithms. It is found that the LSTM-DBA achieves the lowest jitter in comparison with the other two DBAs. As the variation of delay between two consecutive packets is highly affected by network congestion, a significant reduction in network congestion by the LSTM-DBA helps to reduce the upstream jitter.

Figure 8 shows the packet loss ratio performance comparison among the three DBA algorithms. As we can see, the FNN-DBA and LSTM-DBA achieve a lower packet loss ratio compared to the RR-DBA. This is due to the reduction in the transmission waiting time, which allows ONUs to transmit their received packets more quickly, and this minimizes the probability of dropping the packets from the ONU buffer. As shown in Fig. 5, the LSTM

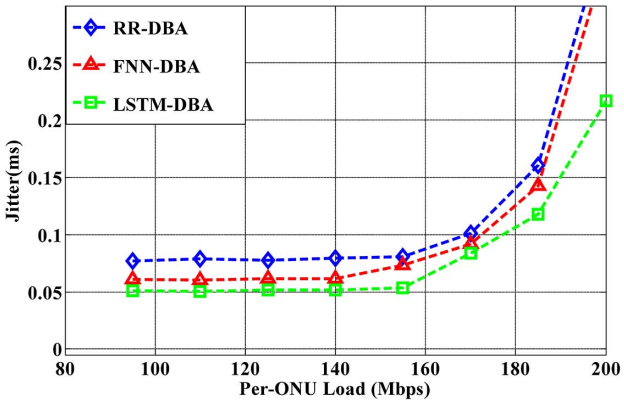

Fig. 7. Upstream jitter performance comparison of RR-DBA, FNN-DBA, and LSTM-DBA.

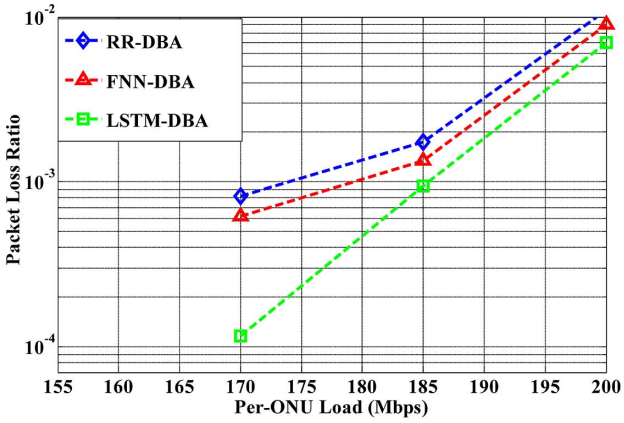

Fig. 8. Upstream packet loss ratio performance comparison of RR-DBA, FNN-DBA, and LSTM-DBA.

can get a more accurate prediction result than the FNN, and it can clear the ONU buffer data much faster than the FNN-DBA with a shorter waiting time for the packets to get served. Therefore, the LSTM-DBA provides a lower packet loss ratio than the FNN-DBA in Fig. 8.

For PON systems with a low traffic load, FBA can be used and can achieve the lowest latency for the MFH links as the packets can be sent from the ONUs to the OLT immediately without waiting for bandwidth grant. To check whether the proposed LSTM-DBA can have a performance close to the FBA under this case, simulations with only one active ONU is conducted. All upstream bandwidth is allocated to the active ONU for the FBA while the other three DBAs work as if the other ONUs are working. All other parameters are the same as above. From the results shown in Fig. 9, it is evident that the LSTM-DBA has a lower delay than the FNN-DBA and RR-DBA and its delay performance is very close to the FBA. This is due to the accurate prediction of the number of packets that arrive at the ONU buffer.

In this Letter, a novel predictive DBA method based on the LSTM for low-latency XG-PON mobile front-haul for a C-RAN is proposed. In the proposed scheme, the buffer

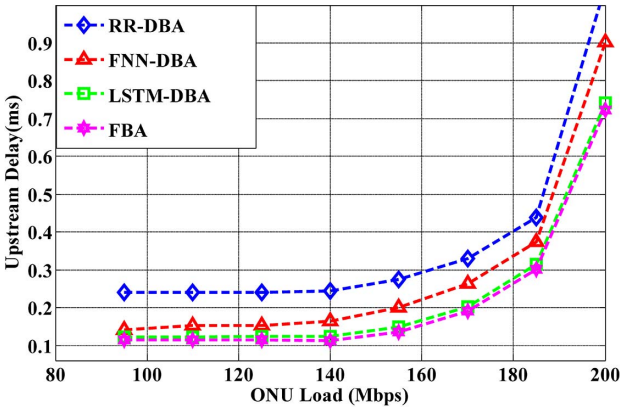

Fig. 9. Upstream delay performance comparison of RR-DBA, FNN-DBA, LSTM-DBA, and FBA for one active ONU case.

occupancy report from the ONUs to the OLT is collected and used for predicting the number of arriving packets by the LSTM. Compared with traditional DBA, the LSTM-DBA can eliminate one-way propagation delay. Simulation results show the performance superiority of the proposed LSTM-DBA algorithm in comparison to the RR-DBA and FNN-DBA algorithms in terms of the upstream delay, jitter, and packet loss ratio. The results also show that the LSTM-DBA can satisfy the strict minimum delay requirement for mobile front-haul.

This work was supported by the National Natural Science Foundation of China (Nos. 61471088 and 61420106011).

References

1. I. Chih-Lin, J. Huang, R. Duan, C. Cui, J. X. Jiang, and L. Li, IEEE Access **2**, 1030 (2014).
2. A. Pizzinat, P. Chanclou, T. Diallo, and F. Saliou, in *Proceedings of ECOC* (2014), p. 1.
3. D. Hisano, H. Uzawa, Y. Nakayama, H. Nakamura, J. Terada, and A. Otaka, IEEE J. Sel. Areas Commun. **36**, 2508 (2018).
4. 3GPP, TR 38.801, v14.0.0, "Study on new radio access technology: radio access architecture and interfaces (Release 14)" (2016).
5. J. Zheng and H. T. Mouftah, Opt. Switch. Netw. **6**, 151 (2009).
6. Z. Xie, H. Li, and Y. Ji, Chin. Opt. Lett. **7**, 4 (2009).
7. T. Kobayashi, H. Ou, D. Hisano, T. Shimada, J. Terada, and A. Otaka, in *Proceedings. of OFC* (2016), p. 1.
8. A. M. Mikaeil, W. Hu, T. Ye, and S. B. Hussain, IEEE/OSA J. Opt. Commun. Netw. **9**, 984 (2017).
9. A. M. Mikaeil, W. Hu, and S. B. Hussain, in *Proceedings of ICTON* (2018), p. 1.
10. F. Musumeci, C. Rottondi, A. Nag, I. Macaluso, D. Zibar, M. Ruffini, and M. Tornatore, https://arxiv.org/abs/1803.07976 (2018).
11. H. Huang, A. Yang, L. Feng, G. Ni, and P. Guo, Chin. Opt. Lett. **15**, 050601 (2017).
12. J. Mata, I. de Miguel, R. J. Duran, N. Merayo, S. K. Singh, A. Jukan, and M. Chamania, Opt. Switch. Netw. **28**, 43 (2018).
13. Y. Bengio, P. Simard, and P. Frasconi, IEEE Trans. Neural Netw. **5**, 157 (2002).

14. S. Hochreiter and J. Schmidhuber, Neural Comput. **9**, 1735 (1997).
15. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, Comput. Sci. **3**, 212 (2012).
16. I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," http://www.deeplerningbook.org (2016).
17. D. Ammar, T. Begin, and I. Guerin-Lassous, in *Conference of ICST* (2011).
18. G. F. Riley and T. R. Henderson, *The ns-3 Network Simulator* (Springer, 2010).
19. X. Wu, K. N. Brown, C. J. Sreenan, P. Alvarez, M. Ruffini, N. Marchetti, D. Payne, and L. Doyle, in *Conference on ICST 195* (2013).