

Research Article

Implementation of a One-LLID-per-Queue DBA Algorithm in EPON

N. A. M. Radzi,¹ N. M. Din,¹ M. H. Al-Mansoori,² and H. Zainol Abidin³

¹ Centre for Communications Service Convergence Technologies, Universiti Tenaga Nasional, Jalan Ikram-Uniten, 43000 Kajang, Selangor, Malaysia

² Faculty of Engineering, Sohar University, 311 Sohar, Oman

³ Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia

Correspondence should be addressed to N. A. M. Radzi; asyikin@uniten.edu.my

Received 7 January 2014; Accepted 29 April 2014; Published 25 May 2014

Academic Editor: Giancarlo C. Righini

Copyright © 2014 N. A. M. Radzi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The advantages of Ethernet passive optical network (EPON) are setting it to be a natural ubiquitous solution for the access network. In the upstream direction of EPON, the directional property of the splitter requires that the traffic flow be mitigated to avoid collision. A dynamic bandwidth allocation (DBA) scheme is desirable in optimizing the bandwidth usage further. In this paper, a global priority DBA mechanism is discussed. The mechanism aims to reduce the overall delay while enhancing the throughput and fairness. This study was conducted using MATLAB where it was compared to two other algorithms in the literature. The results show that the delay is reduced up to 59% and the throughput and fairness index are improved up to 10% and 6%, respectively.

1. Introduction

One of the most attractive solutions to the last mile bandwidth bottleneck in communication systems is Ethernet passive optical network (EPON). It gains its popularity due to the large coverage area, the reduction in energy usage as compared to copper, the reduction in fiber deployment, and lower cost of maintenance [1].

In EPON, collision may occur in the upstream transmission when multiple optical network units (ONUs) transmit data to the optical line terminal (OLT) simultaneously. It occurs because the data from multiple ONUs need to share the same fiber from the splitter to the OLT.

A current method of avoiding collision is by using dynamic bandwidth allocation (DBA) algorithm. Up to date, a number of DBA algorithms have been proposed in the literature [2–12]. In general, we can divide the algorithms into two parts: DBAs that use one logical link identifier (LLID) per ONU and DBAs that use multiple LLIDs per ONU.

One LLID per ONU means that an LLID is allocated to the entire ONU. It creates a hierarchical scheduling structure in which the OLT assigns the bandwidth to the ONUs and the ONUs will further subdivide the bandwidth to multiple

queues inside the ONU. On the other hand, multiple LLIDs per ONU mean that an LLID is allocated to each queue in every ONU.

Luo and Ansari [2] propose DBA with multiple services algorithm that combines limited scheduling in inter-ONU allocation with nonstrict priority scheduling in intra-ONU allocation. Excessive bandwidth is also combined with a nonstrict priority scheduling in [3]. However, the shortcoming of this method is increased queuing delay because all packets would have to wait a full cycle between report and transmission.

In order to overcome the problem in nonstrict priority scheduling, strict priority scheduling is used. It allows newly arriving packets with high priority to be transmitted over the lower-priority packets which are already stored and reported in the OLT. It would cause the lower-priority packets to starve. This phenomenon is known as light load punishment.

An example of the algorithm that uses strict priority scheduling is proposed by Nikolova et al. that combines limited scheduling with strict priority scheduling [4]. In this algorithm, two types of scheduling are proposed in intra-ONU allocation which are full priority scheduling (FPS) and interval priority scheduling (IPS). With FPS, packets are sent

according to the standard strict priority scheduling, whereas IPS is proposed where it eliminates the light load punishment by using the two-stage buffer explained by Kramer et al. [5]. Besides that, algorithms in [6, 7] are proposed by combining excessive bandwidth mechanism with a strict priority scheduling in intra-ONU.

In modified smallest available report first (MSARF) [7], ONUs are sorted in an ascending order according to their bandwidth demands. The smallest ONUs are served first. Using excessive bandwidth mechanism in the inter-ONU allocation, a minimum guaranteed bandwidth is defined for every ONU. ONUs that request less than the minimum guaranteed bandwidth are considered underloaded ONUs; otherwise, they are considered overloaded ONUs. Underloaded ONUs are granted as per request, and the excessive bandwidth from these ONUs is accumulated so that it can be distributed to the overloaded ONUs according to the proportion to ensure fairness. In order to avoid overgranting, the granting of overloaded ONUs is capped up to their requested bandwidth only. For intra-ONU allocation, only expedited forwarding (EF) packets that have been reported in the previous REPORT message are granted first. Subsequently, assured forwarding (AF) packets are granted before best effort (BE) packets including the new AF packets that arrived during EF and AF transmission.

The excessive bandwidth mechanism is also used in intra-ONU allocation rather than in inter-ONU as has been used in [8, 9]. In cyclic-polling-based DBA scheme with service level agreement (CPBA-SLA) [9], inter-ONU allocation is done by dividing the ONUs into three groups: Groups A, B1, and B2 according to their SLA priority. The granted time slots in the ONUs are then divided into two different subframes. In the first subframe, while OLT schedules Group A, it collects REPORT from Group B1. Then, while OLT schedules Group B1, it collects REPORT from Group A. Nevertheless, in the second subframe, while OLT schedules Group A, it collects Group B2. Inside the ONU, bandwidth allocation is done based on the excessive bandwidth mechanism to each type of DiffServ traffic.

The disadvantage of one LLID per ONU is that it puts intelligence in both OLT and ONU and thus adds the complexity of the algorithm. More importantly, the quality of service (QoS) can only be supported either inside the OLT or inside the ONU. Thus, it does not ensure fairness of the entire system. The overall performance can also be degraded.

In order to ensure that QoS is supported in the entire system, multiple LLIDs per ONU are proposed. It has been used in a two-cycle allocation scheme which is proposed in [10]. The schemes are Grant Before REPORT that is used to allocate EF bandwidth via prediction method and Grant After REPORT that allocates AF and BE bandwidth. Class-of-service (CoS) oriented packet scheduling [11] regulates the traffic of each ONU and CoS using two sets of credit pools: one per ONU and one per CoS. In [12], a proportional sharing with load reservation algorithm provides bandwidth that guarantees per-flow basis and redistributes the unused bandwidth among active flows in proportion to their priority level.

This paper discusses the universal DBA (UDBA) algorithm that uses one LLID per queue. The algorithm reduces the delay up to 59% and increases the throughput and fairness index up to 10% and 6%, respectively. It is organized as follows. Section 2 discusses the DBA algorithm. Section 3 presents the simulation results for the algorithm. Section 4 provides the conclusion.

2. Universal Dynamic Bandwidth Allocation

UDBA algorithm uses the multiple LLIDs per ONU method. The DBA algorithm is placed in the OLT where three queues are placed in the ONU. The three queues are divided into high priority (EF), medium priority (AF), and low priority (BE).

UDBA algorithm uses a strict priority scheduling where the QoS is supported globally inside both OLT and ONUs. One of the advantages of using strict priority scheduling is that it reduces the delay and jitter of high priority traffic.

The UDBA unit works by first collecting all the REPORT messages from every queue in a group before calculating how much bandwidth will be allocated to each queue. Then, the OLT issues the grant by using GATE messages. The polling protocol in UDBA is cycle based. In a single polling cycle in UDBA, each queue is polled once and the bandwidth allocation is based on demand. Bandwidth is distributed in time slots or transmission windows. Each time slot with the size of S bytes is given to a queue once every cycle time, T seconds.

The UDBA algorithm is shown in Pseudocode 1. The DBA algorithm in the OLT will check first whether the queues are underloaded or overloaded. Underloaded queues are granted as per request and overloaded queues are granted with distribution of the excessive bandwidth, $B_{i,j}^{\text{distribution}}$, as follows:

$$B_{i,j}^{\text{distribution}} = B_{i,j}^{\text{min}} + B_{\text{total}}^{\text{excess}} \left(\frac{B_{i,j}^{\text{shortage}}}{B_{\text{total}}^{\text{shortage}}} \right), \quad (1)$$

where $B_{i,j}^{\text{min}}$ is the limitation bandwidth for queue i in ONU j , $B_{\text{total}}^{\text{excess}}$ is the total excessive bandwidth, $B_{i,j}^{\text{shortage}}$ is the shortage bandwidth for queue i in ONU j , and $B_{\text{total}}^{\text{shortage}}$ is the total shortage bandwidth.

However, the $B_{i,j}^{\text{distribution}}$ is capped at the requested bandwidth to avoid overgranting.

The granting process is done for EF, followed by AF and finally BE. The benefit of granting EF followed by AF then finally BE traffic is to ensure that the QoS is supported in each group. By supporting the QoS, the real time traffic delay is decreased, the throughput performance is increased, and the fairness of the system is maximized.

In UDBA, the underloaded queues are categorized as the “underloaded” group. The reason why OLT divides the queues into underloaded and overloaded groups is to free UDBA from the overhead. This can be best depicted in Figure 1. Figure 1 differs from the traditional method in which the OLT does not need to wait for all queues to arrive before calculating the DBA and granting the bandwidth. This way,

```

Begin
If requested bandwidth ≤ limitation bandwidth,
    Calculate excessive bandwidth for each queue,
    Calculate total excessive bandwidth from every queue,
    Grant EF as per request,
    Grant AF as per request,
    Grant BE as per request,
Else
    Calculate shortage bandwidth of each queue,
    Calculate total shortage bandwidth from every queue,
    Calculate distribution of excessive bandwidth,
    If distribution of excessive bandwidth ≤ requested bandwidth,
        Grant distribution of excessive bandwidth for EF,
        Grant distribution of excessive bandwidth for AF,
        Grant distribution of excessive bandwidth for BE,
    Else,
        Grant EF as per request,
        Grant AF as per request,
        Grant BE as per request,
    End
End
    
```

PSEUDOCODE 1: Pseudocode for UDBA algorithm.

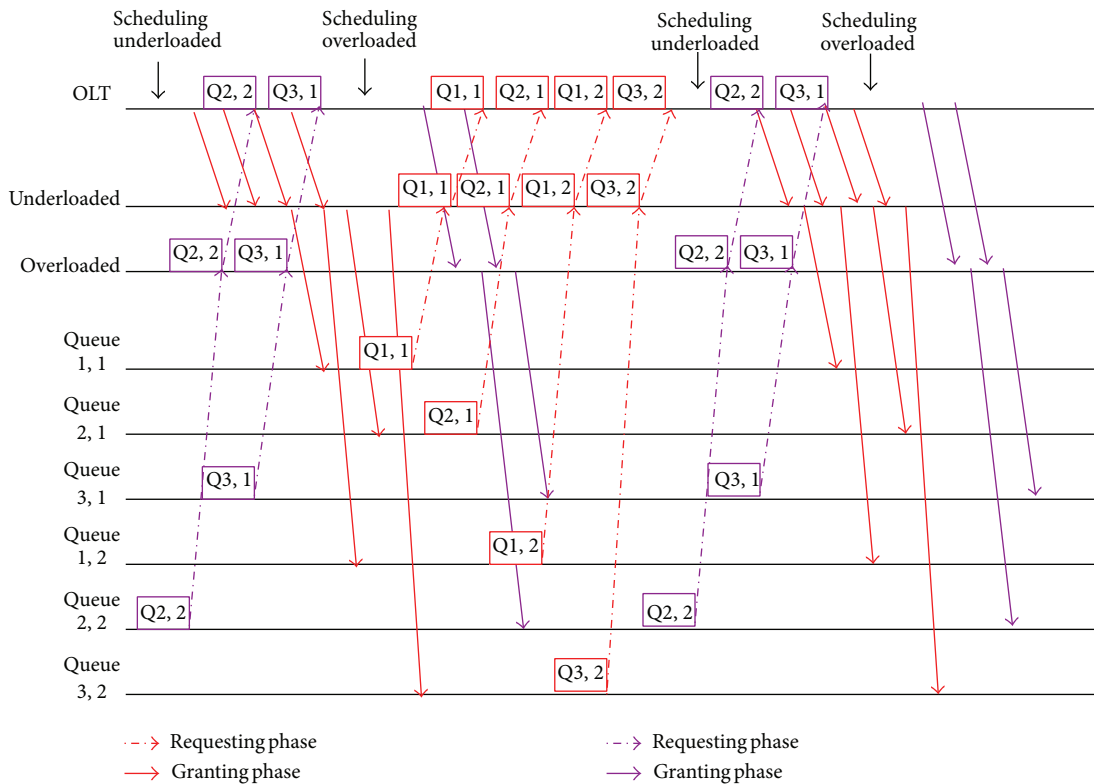


FIGURE 1: UDBA scheduling.

TABLE 1: Default system parameters.

Description	Value
Number of ONUs	16
Number of priority queues	3
EPON line rate	1 Gbps
Buffer size in ONU	10
Maximum cycle time	2 ms
Guard time between adjacent slots	5 μ s
Limitation EF bandwidth	6125 bytes
Limitation AF bandwidth	12250 bytes
Limitation BE bandwidth	12250 bytes

each cycle will not incur an overhead equal to the maximum roundtrip delay plus message processing delay at each level in the hierarchy.

However, there is a tradeoff between the delay and fairness considering that scheduling the groups independently provides fairness only within each group. The impact of fairness is reduced in UDBA by carefully grouping the queues.

With the UDBA algorithm, the function of ONU becomes very simple where it does not need to perform any bandwidth allocation. This is because, if traffic priorities are only managed by each ONU itself, OLT allocates the bandwidth to ONUs just in terms of the ONUs' buffer sizes and without considering priorities. Therefore, priorities have local meaning, that is, within each ONU. Consequently, the bursty higher priority data at one ONU may not get more bandwidth than the bursty lower priority data at another ONU. Hence, performing the DBA in centralized manner gives OLT the overall view of the queues so that it is able to ensure the priorities globally.

3. Simulation Study

A discrete event simulation of the discussed UDBA was conducted using MATLAB for an EPON with one OLT to 16 ONUs architecture. The simulation parameters are summarized in Table 1. The simulation was done four times for four different traffic mixtures (EF, AF, and BE) which are (20%, 40%, 40%), (25%, 25%, 50%), (29%, 42%, 29%), and (20%, 20%, 60%), respectively.

The typical traffic mixture ratio of EF, AF, and BE classes is (20%, 40%, 40%) [7]. However, since the Ethernet traffic is bursty, the proportion of the UDBA traffic profile is varied to the other three different traffic mixtures in order to simulate their effects. This is to prove that, even in any traffic conditions, UDBA shows advantages towards the overall performance of the system.

The traffic is generated based on the source code from NS-2 where we classified services into three classes, EF, AF, and BE, based on differentiated services per hop behaviours.

We assume the EPON link capacity to be 1 Gbps for upstream with the link rate of 100 Mbps between the users and ONU.

It is simulated as constant bit rate (CBR) with a packet length of 70 bytes [13]. ON-OFF source generator traffic is used in AF traffic where the ON and OFF intervals are drawn according to a Pareto distribution with a burst, $H = 0:8$. Pareto has been widely used to model self-similar traffic in the Internet [14]. For BE traffic, ON-OFF source generator traffic is also used where the ON and OFF intervals are drawn according to an exponential distribution. The length of packets generated during ON state of AF and BE traffic follows the trimodal distribution used in [15]. These three modes correspond to the most frequent packet sizes 64, 594, and 1518 bytes observed in backbone and cable networks. In the simulation, each of these packets is generated with a frequency of 62%, 10%, and 28% of the entire packet sizes, respectively. These are based on the measurements taken for these packets in the cable network head-ends [16].

In order to prove the advantages of the UDBA algorithm discussed, we compare it with two other algorithms in the literature known as MSARF and CPBA-SLA. The significance of using these algorithms is that, because both of these algorithms support QoS and they are class-based, they support the same type of DiffServ as UDBA.

From the four traffic mixtures observed in Figure 2, it can be seen that the EF delay for the three algorithms complies with the IEEE 802.1D standard that sets voice delay to be less than 10 ms. In all four traffic mixtures, it proves that EF traffic for UDBA is shorter than MSARF and CPBA-SLA.

EF delay is shorter in UDBA algorithm because the packets are granted globally within the group where OLT always grants EF packets first before considering AF packets and BE packets. In other words, the strict priority scheduling is used globally within the group in UDBA algorithm. This makes every EF packet arrive during waiting time for a GATE message to be granted at the same cycle and thus reduces the EF delay. This differs from MSARF and CPBA-SLA that grant the packets according to local priority. Granting the packets according to local priority means that OLT grants the packets to the ONU first according to first in first out (FIFO); only then ONU grants them to their own queue according to their priority. In other words, it means that some AF and BE packets are served before serving the EF packets, causing higher EF delay. Besides that, since local priority algorithms use two-stage buffers, EF delay increases up to three times [16]. This can be explained by the fact that a packet that arrives at a random time would have to wait, on average, half a cycle in the first buffer (multiple-priority queues) and exactly one cycle in the second buffer (FIFO queues). However, with UDBA where only one priority buffer is implemented, the average EF delay is only half a cycle time.

Figure 3 shows that the AF delays for all three algorithms in four different traffic mixtures are less than 100 ms, which comply with the IEEE 802.1D standard. It can be observed that UDBA algorithm has the shortest AF delay compared to MSARF and CPBA-SLA for all four traffic mixtures. UDBA is better than MSARF around 29%, whereas it is better than CPBA-SLA around 59%. The reason that contributes to a shorter AF delay in UDBA algorithm is again because the AF packets are granted after the EF packets and the allocation is done globally within the group according to the strict priority

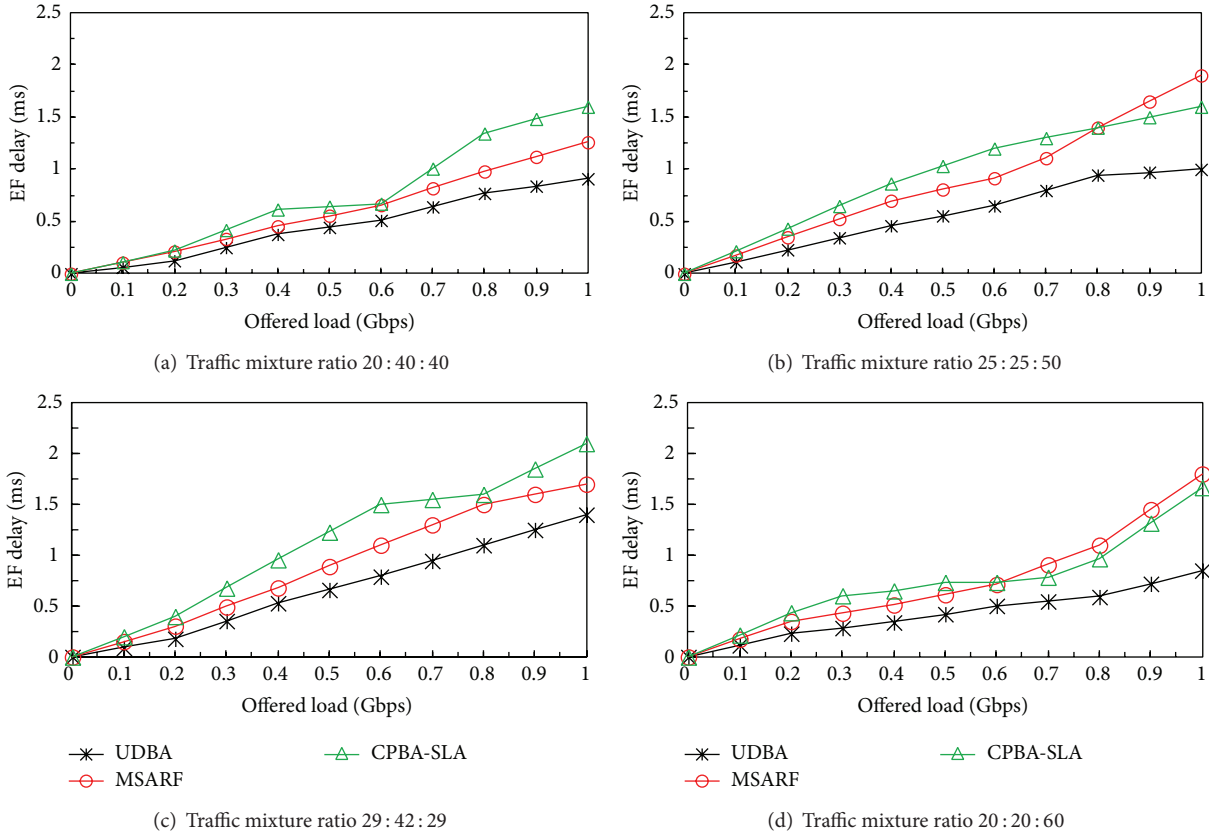


FIGURE 2: EF delay versus offered load for UDBA, MSARF, and CPBA-SLA.

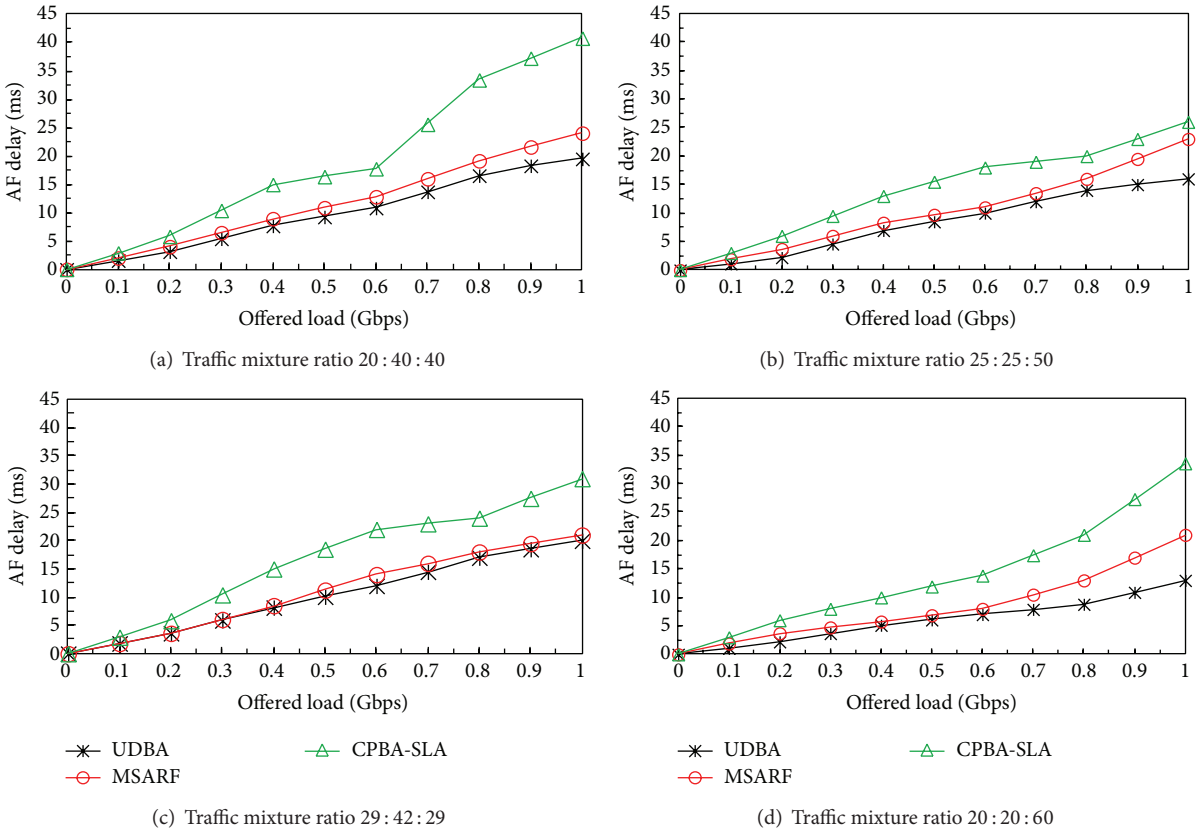


FIGURE 3: AF delay versus offered load for UDBA, MSARF, and CPBA-SLA.

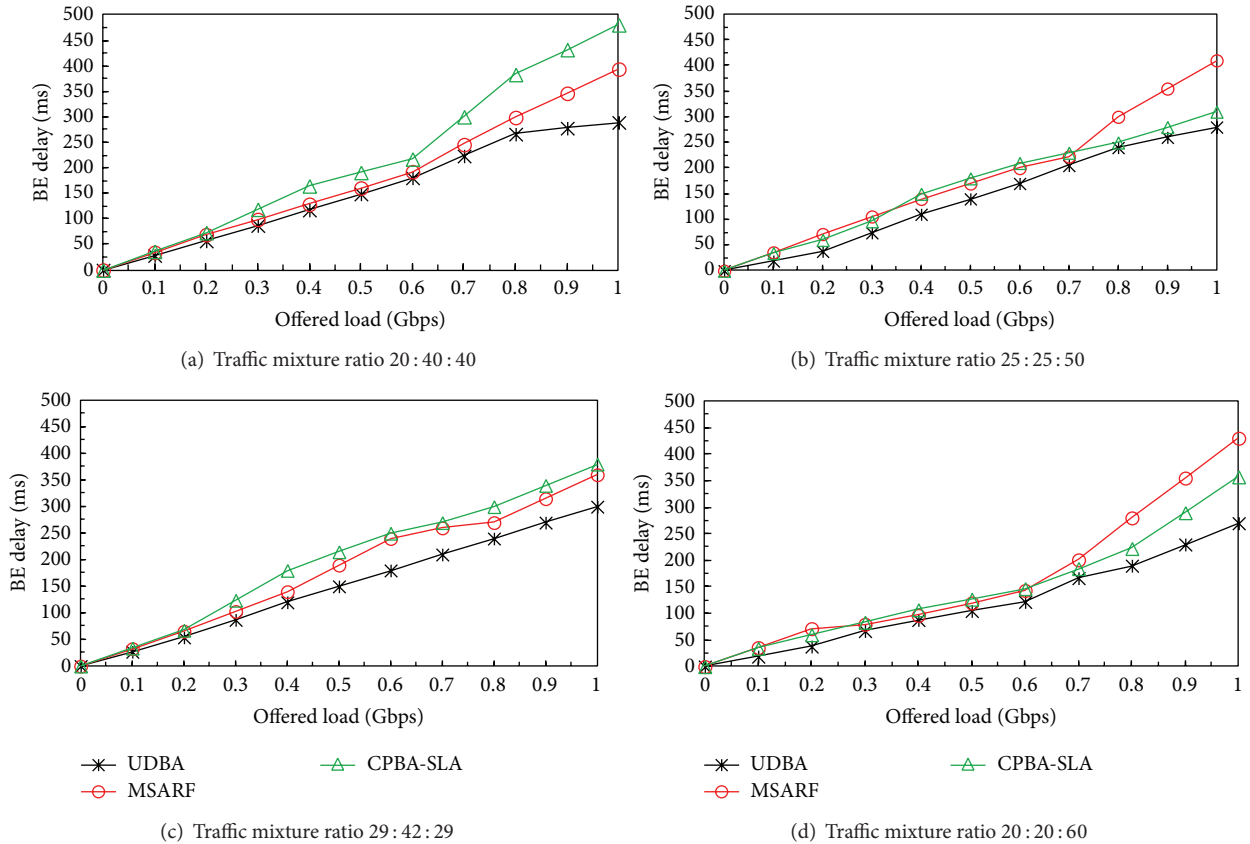


FIGURE 4: BE delay versus offered load for UDBA, MSARF, and CPBA-SLA.

scheduling. In other words, every AF packet that arrives during the waiting time for a GATE message is granted at the same cycle. Besides that, the cycle time is lower as it uses only one priority buffer.

The BE traffic for light loads in UDBA in Figure 4 shows almost similar delay compared to MSARF and CPBA-SLA. This means that, when the traffic load is light, the UDBA algorithm does not really affect the system performance. This is because, at light loads, there is a tradeoff between strict priority scheduling and light load punishment. Light load punishment occurs when the OLT grants the requested time slot through the next GATE message. This is because, during the waiting time of a GATE message, more packets arrive to the queue. These packets sometimes have higher priority than the ones already stored in the queue. Therefore, at the next transmission, they will be transmitted first before the lower-priority packets. This causes some lower-priority packets to be left unattended in the queue. This situation may take place repeatedly, causing some lower-priority packets to be delayed for multiple cycle times. A lower-priority packet will finally be transmitted when lower-priority packets are accumulated and reported to the OLT more than the newly arriving higher-priority packets.

However, as mentioned earlier, this problem occurs only in light loads, and hence the term “light load” punishment originates. Since MSARF and CPBA-SLA use FIFO queuing that involves no traffic priority in inter-ONU level, it is

expected to have less light load punishment. Thus, at light loads, all three algorithms have more or less the same performance. However, as the load increases, the queue size of each traffic in local priority DBA algorithms becomes larger, making it harder for the packets in the lower-priority traffic streams to enter the second stage FIFO queue since it can only carry the packets from higher-priority queues in each polling cycle.

Furthermore, with high traffic load, some of the packets from the lower-priority traffic streams could stay in the FIFO queue waiting for the transmission in the next polling cycle. These packets can delay the transmission of the packets from the higher-priority traffic in the next polling cycle resulting in higher delay to the higher-priority traffic. This differs from UDBA where, as the load increases, the queue of lower-priority packets grows faster and the light load punishment decreases. This is the reason why at high loads UDBA outperforms MSARF and CPBA-SLA in terms of BE delay.

From all four graphs in Figure 5, it can be observed that at light load all three algorithms show nearly similar performance in terms of throughput. This is because at light load more queues are underloaded. During underloaded scheduling, all three algorithms show good performance in utilizing the bandwidth. But as the load increases further, the overloaded queues increase and that is when UDBA starts to show improvement because it grants the bandwidth

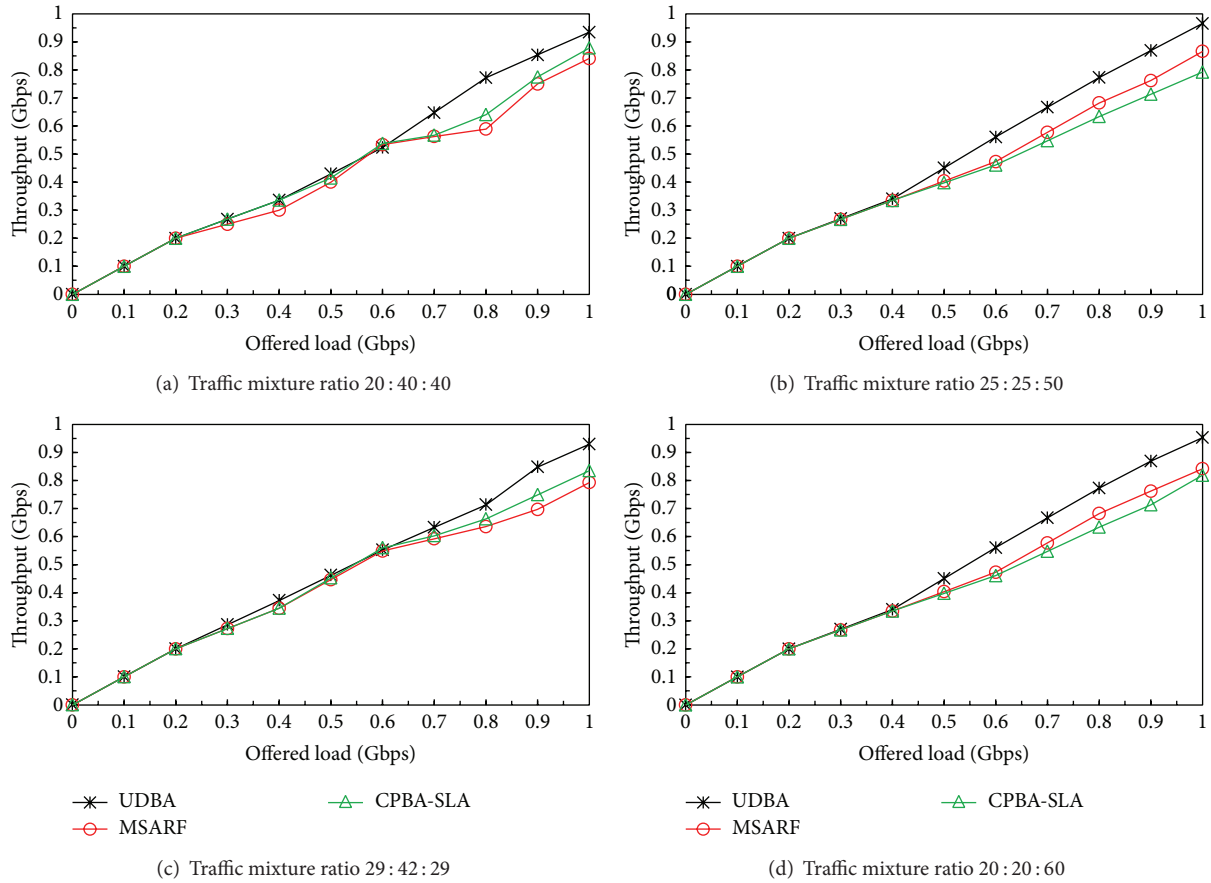


FIGURE 5: Throughput versus offered load for UDBA, MSARF, and CPBA-SLA.

accordingly. Thus, it improves the throughput of the entire EPON system.

Overall, it can be observed that UDBA algorithm has higher throughput as high as 10% compared to MSARF algorithm and as high as 6% compared to CPBA-SLA algorithm. This is due to the fact that the OLT has the overall view of all packet sizes in each queue. Thus, the number of packets transmitted by a queue would correspond to the threshold reported in the REPORT message. This decreases the unused slot remainder and in turn increases the throughput. It differs from local priority algorithms where the OLT is unable to foresee the number of packets each queue should transmit. It causes an assigned slot to have an unused remainder, even if the OLT chooses a slot size for an ONU exactly equal to a sum of thresholds reported by the ONU.

Besides that, UDBA allocates excessive bandwidth universally and fairly from underloaded queues to overloaded queues. The ability of the OLT to have the overall view of every queue causes the allocation of excessive bandwidth to be done more accurately.

But as the load increases further, the overloaded queues increase and that is when UDBA starts to show improvement because it grants the bandwidth accordingly. Thus, it improves the bandwidth utilization of the entire EPON system.

The fairness index versus offered load of UDBA, MSARF, and CPBA-SLA is shown in Figure 6 for various traffic conditions as has been conducted previously. When the traffic load is low, the fairness index shows nearly similar performance since most queues and ONUs are underloaded. As the load gets higher and the overloaded ONUs and queues increase, UDBA shows higher fairness index compared to MSARF and CPBA-SLA. The fairness index is relatively higher in UDBA because it ensures global priority where OLT is in charge of allocating the bandwidth for each queue.

4. Conclusion

This paper discusses a one-LLID-per-queue DBA algorithm that uses strict priority scheduling to support QoS. We conducted a simulation work using MATLAB, where we study the performance in terms of delay, throughput, and fairness index. The result shows that the packet delay and jitter are lower in UDBA algorithm, especially for real time traffic. UDBA reduces the delay as high as 37% compared to MSARF and it reduces the delay as high as 59% compared to CPBA-SLA. In terms of throughput, UDBA is better as high as 10% compared to MSARF and as high as 6% compared to CPBA-SLA. UDBA is also better than MSARF in terms of

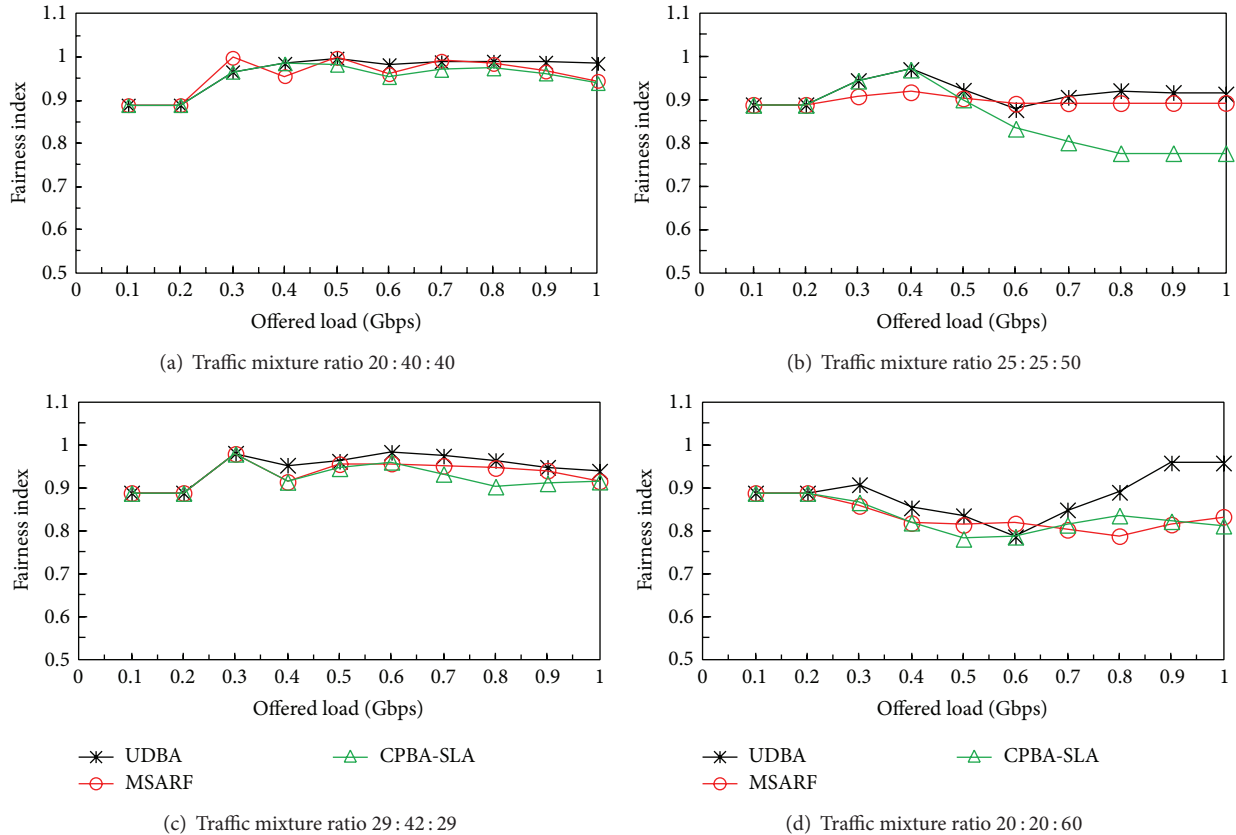


FIGURE 6: Fairness index versus offered load for UDBA, MSARF, and CPBA-SLA.

fairness index as high as 4% and is better than CPBA-SLA as high as 6%. It can be concluded that the delay is reduced and the throughput and fairness index are enhanced by using one LLID per queue.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] I. S. Hwang, J. Y. Lee, and Z. D. Shyu, "A novel dynamic bandwidth allocation mechanism for star-ring-based EPON," *ISRN Communications and Networking*, vol. 2011, Article ID 837453, 9 pages, 2011.
- [2] Y. Luo and N. Ansari, "Bandwidth allocation for multiservice access on EPONs," *IEEE Communications Magazine*, vol. 43, no. 2, pp. S16–S21, 2005.
- [3] C. Sue and W. Sung, "Fitting scheduling timing-elastic weighted granting (FST-EWG): an EPON DBA algorithm," *Journal of Optical Communications and Networking*, vol. 4, no. 6, pp. 468–479, 2012.
- [4] D. Nikolova, B. Van Houdt, and C. Blondia, "Dynamic bandwidth allocation algorithms for ethernet passive optical networks with threshold reporting," *Telecommunication Systems*, vol. 28, no. 1, pp. 31–52, 2005.
- [5] 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. 9, pp. 280–298, 2002.
- [6] H. Lee, W. Yoon, T. Lee, H. Choo, and M. Chung, "Two-Phase Cycle DBA (TCDBA) for differentiated services on EPON," *IEICE Transactions on Communications*, vol. 92, no. 9, pp. 2823–2837, 2009.
- [7] M. V. Dolama and A. G. Rahbar, "Modified smallest available report first: new dynamic bandwidth allocation schemes in QoS-capable EPONs," *Optical Fiber Technology*, vol. 17, no. 1, pp. 7–16, 2011.
- [8] N. Merayo, R. J. Durán, P. Fernández, R. M. Lorenzo, I. De Miguel, and E. J. Abril, "EPON bandwidth allocation algorithm based on automatic weight adaptation to provide client and service differentiation," *Photonic Network Communications*, vol. 17, no. 2, pp. 119–128, 2009.
- [9] S. I. Choi and J. Park, "SLA-aware dynamic bandwidth allocation for QoS in EPONs," *Journal of Optical Communications and Networking*, vol. 2, no. 9, Article ID 5560808, pp. 773–781, 2010.
- [10] A. Shami, X. Bai, C. M. Assi, and N. Ghani, "Jitter performance in Ethernet passive optical networks," *Journal of Lightwave Technology*, vol. 23, no. 4, pp. 1745–1753, 2005.
- [11] H. Naser and H. T. Mouftah, "A joint-ONU interval-based dynamic scheduling algorithm for ethernet passive optical networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 4, pp. 889–899, 2006.

- [12] F. D. M. Pereira, N. L. S. da Fonseca, and D. S. Arantes, "A fair scheduling discipline for Ethernet passive optical networks," *Computer Networks*, vol. 53, no. 11, pp. 1859–1878, 2009.
- [13] X. Bai, A. Shami, and C. Assi, "On the fairness of dynamic bandwidth allocation schemes in Ethernet passive optical networks," *Computer Communications*, vol. 29, no. 11, pp. 2123–2135, 2006.
- [14] V. Paxson and S. Floyd, "Wide area traffic: the failure of Poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, 1995.
- [15] G. Kramer, B. Mukherjee, and G. Pesavento, "Interleaved polling with adaptive cycle time (IPACT): a dynamic bandwidth distribution scheme in an optical access network," *Photonic Network Communications*, vol. 4, no. 1, pp. 89–107, 2002.
- [16] D. Sala and A. Gummalla, "PON functional requirements: services and performance," presented at the IEEE 802.3ah meeting in Portland, Ore, July 2001, http://www.ieee802.org/3/efm/public/jul01/presentations/sala_1.0701.pdf.

