Development of Efficient Dynamic Bandwidth Allocation Algorithm for XGPON

Man Soo Han, Hark Yoo, and Dong Soo Lee

This paper proposes an efficient bandwidth utilization (EBU) algorithm that utilizes the unused bandwidth in dynamic bandwidth allocation (DBA) of a 10-gigabit-capable passive optical network (XGPON). In EBU, an available byte counter of a queue can be negative and the unused remainder of an available byte counter can be utilized by the other queues. In addition, EBU uses a novel polling scheme to collect the requests of queues as soon as possible. We show through analysis and simulations that EBU improves performance compared to that achieved with existing methods. In addition, we describe the hardware implementation of EBU. Finally we show the test results of the hardware implementation of EBU.

Keywords: Dynamic bandwidth allocation, utilization, scheduling, XGPON.

I. Introduction

The emerging solution for high-speed Ethernet in access networks is passive optical networks (PONs). A PON consists of an optical line termination (OLT), a passive splitter, and multiple optical network units (ONUs). In the downstream, a PON is a point-to-multipoint system. The OLT broadcasts a frame to ONUs through the passive splitter. In the upstream, a PON is a point-to-point system. ONUs transmit frames to the OLT in a time division multiplex manner. Only one ONU at a time is allowed to transmit frames to the OLT. To allocate noncollision transmission slots to ONUs, the OLT receives requests from ONUs and performs dynamic bandwidth allocation (DBA).

The gigabit PON (GPON) and Ethernet PON (EPON) are the two major standards in PONs. Recently, to satisfy the high bandwidth demands, the 10-gigabit-capable PON (XGPON) was developed by extending the GPON. In October 2010, ITU-T published XGPON standards G.987.1 and G.987.3. An XGPON supports the data rate of 10 Gbps in the downstream direction and the data rate of 2.5 Gbps in the upstream direction [1].

A GPON is different from an EPON in many ways. The first major difference is that a GPON is synchronous whereas an EPON is not [2]-[4]. The operation of a GPON is synchronized with the downstream frame duration (DFD), which is fixed at 125 μs [5], [6]. The OLT executes a DBA algorithm and transmits a frame to ONUs for every DFD. The upstream frame duration (UFD) is also fixed at 125 μs. The DBA algorithm produces an allocation result for every upcoming UFD. On the other hand, an EPON neither has a fixed frame duration nor is synchronized with the frame duration.

The second major difference is the bandwidth requirement

Manuscript received Jan. 25, 2012; revised May 24, 2012; accepted July 16, 2012.

Man Soo Han (corresponding author, phone: + 061 450 2744, mshan@mokpo.ac.kr) is with the Division of Information Engineering, Mokpo National University, Jeonnam, Rep. of Korea.

Hark Yoo (harkyoo@etri.re.kr) and Dong Soo Lee (d-soolee@etri.re.kr) are with the Communications Internet Research Laboratory, ETRI, Gwangju, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.13.0112.0061

for service classes. To support quality of service (QoS), a GPON has a transmission container (T-CONT) type that represents a service class [5], [6]. A queue of an ONU has its own T-CONT type and must be assigned a transmission bandwidth in accordance with its T-CONT type. For example, an assured bandwidth is assigned to T-CONT types 2 and 3 that the OLT must guarantee. In addition, a nonassured bandwidth is assigned to T-CONT types 3 and 4 that the OLT may dynamically assign. An EPON does not have a bandwidth requirement for a service class because such a requirement is out of the EPON standardization scope.

EPON DBA has received a lot of attention. The recent surveys for EPON DBA can be found in [7], [8]. In contrast, GPON DBA has received little attention [3], [9]-[14]. Furthermore, only a few GPON DBA algorithms are compliant with the GPON standard ITU-T G.984.3 [5]. The GigaPON access network (GIANT) algorithm was the first DBA algorithm to be compliant with the GPON standard [10], [11]. GIANT uses down counters to allocate bandwidth to queues. The OLT can allocate bandwidth to a queue only when the down counter of the queue has expired. Although GIANT provides desirable performance, its allocation mechanism can degrade performance since a request cannot be granted until the down counter expires.

In [12], the predictive colorless grant offset-based scheduling with flexible intervals (PCG-OSFI) method was introduced, which is compatible with the GPON standard requirements. When a grant cannot be allocated to a preplanned transmission slot, the transmission moment of the grant is delayed and the size of the grant is increased in proportion to the delay. Moreover, the traffic arrival amount is predicted based on the past arrival amount. This form of scheduling was used in [12] only for the assured bandwidth allocation of T-CONT types 2 and 3. Like GIANT, however, PCG-OSFI can allocate a grant only when the down counter has expired for the nonassured bandwidth of T-CONT types 3 and 4.

To improve GIANT, the immediate allocation with colorless grant (IACG) method was introduced [13]. IACG is compliant with the GPON standard. IACG uses available byte counters as well as down counters to immediately allocate bandwidth to queues whether or not the down counters have expired. In addition, the OLT evenly distributes the unallocated remainder of bandwidth to all ONUs. It was shown that IACG dramatically improves performance compared to GIANT [13].

The abovementioned GPON DBA algorithms do not consider utilization of unused bandwidth. If the amount of allocated bandwidth of a queue is less than its reserved service amount, a part of the bandwidth remains unused. In the above algorithms, the unused bandwidth cannot be used by a queue whose request amount is greater than its reserved service amount. To achieve high performance, the upstream bandwidth must be efficiently utilized. To the best of our knowledge, the utilization problem of unused bandwidth has never been studied regarding XGPONs or GPONs.

In this paper, we propose an efficient bandwidth utilization (EBU) algorithm to solve the utilization problem. EBU is based on IACG. However, because the available byte counter can be negative in EBU, to utilize the unused bandwidth, the unused remainder of the available byte counter is added to the negative available byte counters. In addition, EBU employs a novel polling scheme to know the requests of queues as soon as possible.

The feasibility of implementation is the crucial factor of GPON DBA algorithms since the OLT must produce the DBA result every 125 μs. To the best of our knowledge, only the IACG and GIANT algorithms have been physically implemented [10], [13]. We implement the EBU algorithm in an FPGA chip and develop an XGPON OLT board that contains the FPGA chip. The board supports a 10-Gbps downstream speed, 2.5-Gbps upstream speed, 256 ONUs, and 4 T-CONT types.

In section II, we explain the XGPON system briefly. In section III, we describe the proposed EBU algorithm. Then, we analyze the IACG and EBU algorithms in section IV using queueing models. In section V, we evaluate the performance of each algorithm and show that EBU outperforms the existing methods, using computer simulations. In section VI, we discuss the hardware implementation issue of the EBU algorithm. In section VII, we show the testbed and describe the test results of the hardware implementation.

II. System Description

An XGPON consists of an OLT and N ONUs. To provide QoS, an XGPON supports multiple service classes, which are known as T-CONT types in the XGPON standard. An ONU maintains a queue for each T-CONT type. Each queue has an allocation identifier (AllocID).

Each operation of an XGPON system is synchronized with a DFD whose length is fixed at 125 μs. In the downstream direction, the OLT broadcasts frames to all ONUs in each DFD. An ONU accepts only frames that match the AllocIDs of its queues. In the upstream direction, only one ONU can transmit frames to the OLT at a time. In each DFD, the OLT collects the bandwidth requests from ONUs and assigns a nonoverlapping transmission time slot to each ONU. The total sum of the assigned transmission time slots is less than or equal to a UFD whose length is fixed at 125 μs. The OLT sends the grant result to each ONU at the beginning of the next DFD. An ONU transmits its frames to the OLT only at its assigned

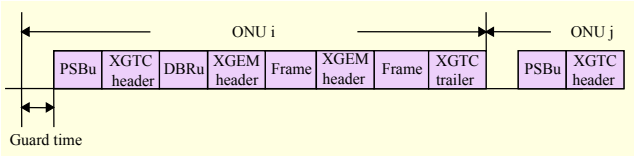

Fig. 1. Burst overhead and XGEM header.

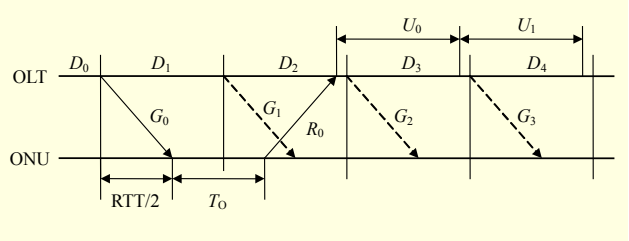

Fig. 2. Transmission timing diagram.

transmission time slot.

To collect the bandwidth requests from ONUs, the OLT assigns a dynamic bandwidth report upstream (DBRu) transmission slot to a queue of an ONU. When a queue receives the DBRu slot, the queue sends its queue length to the OLT using the DBRu slot. In addition, during scheduling, the OLT must consider the transmission overhead, such as a guard time or an upstream physical synchronization block (PSBu) section. To send the scheduling results to ONUs, a bandwidth map (BWmap) is transmitted to ONUs at the beginning of each DFD.

The ONU starts the transmission of its payload with the guard time and the PSBu section. The ONU needs an XGPON transmission convergence (XGTC) header at the beginning of the transmission. The XGTC header shows which ONU transmits the payload. In addition, the ONU requires an XGTC trailer at the end of the transmission to evaluate the transmission bit error rate.

A frame has an XGPON encapsulation method (XGEM) header to show the flow information of the frame. In an XGPON, a frame can be fragmented if the size of a transmission time slot is less than the size of that frame. Each fragment is prepended with the XGEM header. The XGEM header also indicates whether or not a fragment is the last fragment. Figure 1 depicts the burst overhead and the XGEM header.

In this paper, we consider the following four T-CONT types, which are distinguished by their assignable bandwidth [6]:

- T-CONT type 2: the assured bandwidth;
- T-CONT type 3: the assured bandwidth and the nonassured bandwidth;
- T-CONT type 4: the best-effort bandwidth;
- T-CONT type 5: all bandwidths.

Since T-CONT type 1 has a fixed bandwidth and it is statically served, we do not consider T-CONT type 1 in this paper, as in [12], [13]. The assured bandwidth is dynamically assigned to a queue based on the bandwidth demand of the queue. The nonassured bandwidth and the best-effort bandwidth are the surplus bandwidth that the OLT may dynamically allocate to a queue based on the bandwidth demand of the queue.

The service priority order is T-CONT type 2, the assured bandwidth of T-CONT type 3, the nonassured bandwidth of T-CONT type 3, and T-CONT type 4. T-CONT type 5 is a consolidation of other T-CONT types. In this paper, we use T-CONT type 5 to represent the colorless grant that can be used for frames of any T-CONT type. Note that an ONU does not have a queue for T-CONT type 5.

Figure 2 shows the transmission timing diagram of an XGPON when the maximum distance between an ONU and the OLT is 20 km. D_i denotes the DFD and U_i represents the UFD. The DFD and UFD are fixed at 125 μs. The propagation delay T_P is RTT/2 where RTT is a round trip time between an ONU and the OLT. The RTT is 200 μs when the maximum distance between an ONU and the OLT is 20 km. Each ONU requires the ONU response time T_O to receive the BWmap and to prepare an upstream response. The standard value of T_O = 35±1 μs [6]. In addition, an ONU needs the equalization delay T_E to align its upstream transmission with the upstream transmissions of other ONUs [6].

In this paper, we use the polling mechanism of [13] to obtain the actual requests of ONUs. In Fig. 2, G_i, i=0,…,3, is the grant result and R_0 is the bandwidth request. The grant result G_i is produced by scheduling in the DFD D_i and is transmitted to ONUs at the beginning of the next DFD D_{i+1}. When ONU receives the grant result G_0, it waits T_O+T_E, then sends its request R_0 and data frames within the UFD U_0. The request R_0 is the total queue length of a queue. Since the ONU sends R_0 using the DBRu field before it transmits its data frames, the grant result G_0 is not reflected in R_0. The requests received within U_0 will be used in the scheduling at D_4. Since ONU may receive the grants during D_1, D_2, and D_3, the grant results G_1, G_2, and G_3 must be reflected in R_0. Thus, to obtain the actual request, the OLT subtracts the grant results G_0,…,G_3 from R_0 at the beginning of D_4. The OLT remembers the four most recent grants for the actual request calculation. If the maximum distance between an ONU and the OLT is 60 km, the RTT is 600 μs. In this case, the OLT must remember the eight most recent grants to obtain the actual requests.

III. EBU Algorithm

Table 1 shows the service parameters and counters that are used in IACG and EBU.

Table 1. Service parameters and counters

T-CONT type	Bandwidth	Service parameters	Counters
2	assured	*SI, AB*	*SI*_timer, *VB*
3	assured	*SI, AB*	*SI*_timer, *VB*
3	nonassured	*SI′, AB′*	*SI′*_timer, *VB′*
4	best-effort	*SI, AB*	*SI*_timer, *VB*

Let queue(*j*) be a queue with AllocID *j*. The queue(*j*) has two service parameters, *SI*(*j*) and *AB*(*j*), where *SI*(*j*) is the service interval of queue(*j*) with the unit of 125 μs and *AB*(*j*) is the maximum allocation bytes of queue(*j*) that can be used during *SI*(*j*). In addition, the queue(*j*) has two counters, *SI*_timer(*j*) and *VB*(*j*), where *SI*_timer(*j*) is decreased by 1 in each DFD and recharged to *SI*(*j*) when it has expired. *VB*(*j*) denotes the remaining available bytes of queue(*j*) during *SI*(*j*). If queue(*j*) is T-CONT type 3, it has another two service parameters, *SI′*(*j*) and *AB′*(*j*), and another two counters, *SI′*_timer(*j*) and *VB′*(*j*), for the nonassured bandwidth.

Let grant(*j*) and request(*j*) be the grant and the request sizes of queue(*j*), respectively. In IACG, grant(*j*) is the minimum of request(*j*) and *VB*(*j*) and both request(*j*) and *VB*(*j*) are decreased by grant(*j*). *VB*(*j*) is recharged to *AB*(*j*) when *SI*_timer(*j*) has expired. Using an example, we now explain how this mechanism can waste bandwidth. Assume that request(1) = 0, request(2) = 500, *VB*(1) = 500, and *VB*(2) = 100. Also suppose that *SI*_timer(1) expires before *SI*_timer(2). Then, grant(1) = 0, grant(2) = 100, and *VB*(2) becomes 0. To transmit the remainder of request(2), queue(2) must wait until *VB*(2) is recharged. If request(1) = 0 until *SI*_timer(1) expires, then the remaining amount of *VB*(1), 500 bytes, is wasted. If IACG could use the remaining amount of *VB*(1) to grant the remainder of request(2), the bandwidth waste would not exist.

1. Scheduling Mechanism

To mitigate the bandwidth waste, we introduce EBU that improves scheduling and polling mechanisms of IACG. First, we explain the scheduling mechanism. In EBU, *VB*(*j*) can be negative unlike in IACG. The OLT grants the minimum of *AB*(*j*) and request(*j*) only when *VB*(*j*) ≥ 0. Both request(*j*) and *VB*(*j*) are decreased by grant(*j*). This mechanism allows queue(*j*) to use more bandwidth than its *AB*(*j*) during *SI*(*j*).

If we have *VB*(*i*) < 0 and *VB*(*j*) > 0 when *SI*_timer(*j*) expires, it implies queue(*i*) utilized the unused reserved bandwidth of queue(*j*). Hence, *VB*(*j*) must be added to *VB*(*i*) to utilize the bandwidth efficiently. Also, the maximum of *VB*(*i*) is limited to 0, and *VB*(*j*) is decreased by the added amount. The remainder of *VB*(*j*) is added to the other *VB*(*k*) if there exists *VB*(*k*) < 0. In addition, when *SI*_timer(*j*) expires, *VB*(*j*) is increased by *AB*(*j*) and its maximum is limited to *AB*(*j*).

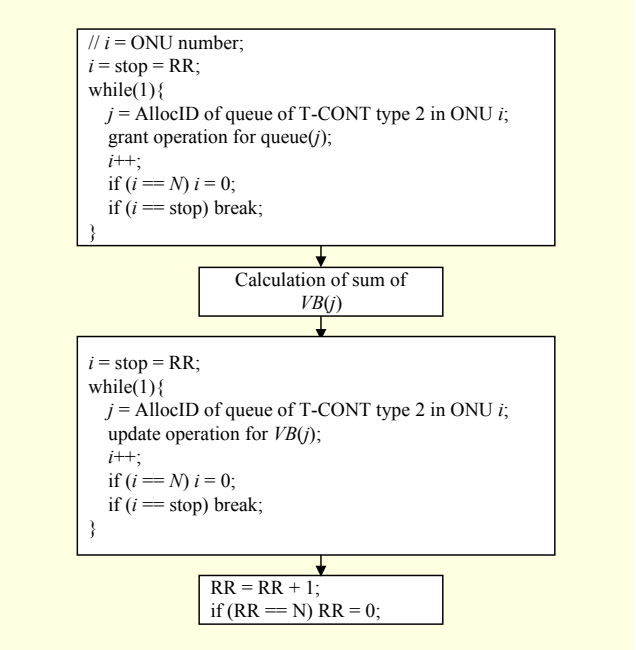

Fig. 3. Scheduling procedure of EBU for T-CONT type 2.

We now show the scheduling procedure of EBU for T-CONT type 2 in Fig. 3, where RR represents the starting ONU of the scheduling operation. First, the grant operation for queue(*j*) is performed in a round-robin manner. Then, the total remaining bandwidth is calculated. Next, the update operation of *VB*(*j*) is performed in a round-robin manner. Finally, the starting ONU is changed in a round-robin manner. The scheduling procedure for other T-CONT types is similar to that shown in Fig. 3.

We now illustrate the pseudocode of the grant operation of queue(*j*) in Fig. 3.

```
if (VB(j) ≥ 0 and FB > 0) {
    grant(j)=min(AB(j),request(j),FB);
    VB(j) -= grant(j);
    request(j) -= grant(j);
    FB -= grant(j);
}
```

The variable *FB* is used to ensure that the sum of all grants is less than the size of the UFD. The initial value of the *FB* is equal to the size of the UFD. For service fairness, the starting queue of the grant operation changes in a round-robin manner. For example, if the grant operation starts from a queue in ONU 1 in the current DBA, then it starts from a queue in ONU 2 in the next DBA.

We now describe the pseudocode of the update operation of $VB(j)$ reflected in Fig. 3. After the grant operation is finished, the update operation starts. Let S_k be the sum of $VB(j)$ whose queue is T-CONT type k and assume that $SI\_timer(j)$ has expired. That is, $S_k = \sum_{j \in V} VB(j)$, where $V=\{i \mid VB(i) > 0, SI\_timer(i) = 0, \text{queue}(i) \in \text{T-CONT type } k\}$.

```
if (VB(j) < 0 and S_k > 0){
   S_k = S_k + VB(j);
   VB(j) = min(0, S_k);
}
if (SI_timer(j) = 0){
   SI_timer(j) = SI(j);
   VB(j) = min(VB(j)+AB(j), AB(j));
}
SI_timer(j)--;
```

For service fairness, the starting queue of the update operation changes in the round-robin manner of the grant operation.

For the above example of IACG, if we use EBU, grant(1) = 0, grant(2) = 500, and $VB(2)$ becomes –400. When $SI\_timer(1)$ expires, $VB(2) = 0$ since $VB(1)$ is added to $VB(2)$. Therefore, EBU uses the unused bandwidth of queue(1) to transmit frames from queue(2).

2. Polling Mechanism

Here, we explain the polling mechanism of EBU. In IACG, the OLT allocates the DBRu slot to queue(j) once per $SI(j)$. In IACG, queue(j) has a polling flag PF(j). The OLT can allocate the DBRu slot to queue(j) when PF(j) = 0. The flag PF(j) is set to 1 when the OLT allocates the DBRu slot to the queue. The flag PF(j) is reset to 0 when $SI\_timer(j)$ expires. Hence, queue(j) gets the DBRu slot once per $SI(j)$, independent of its queue status.

For efficient scheduling, the OLT must know the actual requests of queues as soon as possible. In respect to polling, the best scheme is that every queue reports its request to the OLT in every UFD. The DBRu slot is 4 bytes long [6]. In addition, the DBRu slot requires the burst overhead in Fig. 1. Therefore, if the DBRu slots are allocated to all queues in every UFD, the upstream bandwidth will be wasted, especially when the number of queues is large.

Like IACG, EBU allocates one DBRu slot to queue(j) once per $SI(j)$, independent of the status of queue(j). Unlike IACG, however, EBU can allocate an additional DBRu slot queue(j) during its $SI(j)$. To mitigate the bandwidth waste and to minimize the burst overhead due to the excessive number of DBRu slots, EBU allocates the additional DBRu slot only when a specific condition is met. In EBU, the OLT can allocate the additional DBRu slot to queue(j) if grant(j) > 0 after the grant operation.

IV. Approximate Analysis

In this section, we approximate IACG and EBU methods using M/G/1 queueing models to show how EBU improves the mean delay compared with IACG. For simplicity, we consider L queues of T-CONT type 2. Suppose queue(j) has the arrival rate of λ_j. In IACG, the service amount of queue(j) is limited by $AB(j)$ during $SI(j)$. Hence, the service rate of queue(j) is given by $\mu_j = AB(j)/SI(j)$. The mean waiting time in queue(j) is [15]

$$W_j = \frac{\lambda_j b_j^2}{2(1-\rho_j)}, \tag{1}$$

where b_j^2 is the second moment of the service time distribution of queue(j) and $\rho_j = \lambda_j/\mu_j$ is the utilization factor. Let $\lambda = \sum_{j=1}^{L} \lambda_j$; then, the input traffic ratio of queue(j) is given by λ_j/λ. Therefore, we obtain the mean waiting time of IACG, WI, as

$$WI = \sum_{j=1}^{L} W_j \frac{\lambda_j}{\lambda}. \tag{2}$$

In EBU, the service rate of queue(j) is not limited by $AB(j)/SI(j)$ since queue(j) can use the unused bandwidth of other queues. If the unused bandwidth of a queue is fully utilized by other queues, packets are served with the service rate of $\mu = \sum_{j=1}^{L} \mu_j$. Then, the EBU method can be modeled by a single queue whose service rate and arrival rate are μ and λ, respectively. Therefore, we have the mean waiting time of EBU, WE, as [15]

$$WE = \frac{\lambda b^2}{2(1-\rho)}, \tag{3}$$

where b^2 is the second moment of the service time distribution and $\rho = \lambda/\mu$.

To illustrate the difference between WI and WE, consider a simple example. Assume the service rate and the arrival rate of all queues are $\mu_j = \mu/L$ and $\lambda_j = \lambda/L$, respectively, in both methods. Also assume that every packet has the same size (that is, the M/D/1 system). Then, we have $b_j^2 = 1/\mu_j^2$ and $b^2 = 1/\mu^2$ for the IACG and EBU methods, respectively [15]. From (2) and (3), we obtain $WI = \lambda L/(2\mu^2(1-\rho))$ and $WE = \lambda/(2\mu^2(1-\rho))$. In this example, EBU is L times better than IACG in the mean waiting time.

V. Performance Evaluation

We now compare the respective performance of EBU, IACG, and PCG-OSFI. We consider an XGPON system with 16 ONUs, a line rate from users to the ONU link of 200 Mbps, an upstream channel rate of 2.48832 Gbps, that RTT = 200 μs, and that T_O = 35 μs. The size of each queue is 1 MB. We suppose traffic is balanced so that each ONU has an identical load. In addition, we suppose that the T-CONT type probability of an incoming packet is uniformly distributed so that each T-CONT has an identical traffic load.

For T-CONT type 2, we set AB = 7,812 and SI = 5, which is equivalent to 100 Mbps. For T-CONT type 3, we set $AB = AB' =$ 7,812 and $SI = SI' = 10$, which means 50 Mbps is given to both the assured bandwidth and the nonassured bandwidth. For T-CONT type 4, we set AB = 15,624 and SI = 10, which is equivalent to 100 Mbps. The initial value of FB is 38,880 bytes. IACG uses the same parameters. The reserved bandwidth for each T-CONT type of PCG-OSFI is equal to that of EBU. We use the self-similar traffic model that each ONU is fed by a number of Pareto distributed on-off processes. The shape parameters for the on and off intervals are set to 1.4 and 1.2, respectively [17]. In addition, the frame size follows the trimodal distribution in which the frame sizes are 64, 500, and 1,500 bytes and their load fractions are 60%, 20%, and 20%, respectively, as in [10]. Each simulation is performed until the total number of frames transmitted by ONUs exceeds 10^9 for each algorithm.

Increasing the offered load rate of an ONU from 0.1 to 0.99, we simulate the algorithms and compare their performance. Figures 4 through 6 respectively illustrate the mean delay, the frame delay variance, and the frame loss rate of each method. As we can see from the figures, EBU outperforms the other methods for T-CONT types 2 and 3. EBU is worse than IACG in the performance of T-CONT type 4, especially when the traffic is heavy. PCG-OSFI is worse than other methods in the performance of T-CONT types 3 and 4. The main reason is that PCG-OSFI can grant the nonassured bandwidth of T-CONT types 3 and 4 only when their down counters have expired [12].

ITU-T G.987.1 recommends that an XGPON system must accommodate services that require a maximum mean delay of 1.5 ms [1]. In our simulation scenario, only EBU satisfies the requirement for T-CONT types 2 and 3 for all traffic loads. The frame loss rate of EBU is better than the frame loss rates of other methods for T-CONT types 2 and 3 in Fig. 6. For T-CONT type 2, the loss rate is zero for all traffic loads in EBU, owing to the utilization of the unused bandwidth.

In general, the mean delay increases and the packet loss rate decreases as the queue size increases. The mean delay and the packet loss rate of a queue decrease if the value of AB increases. As the value of SI decreases, the polling interval decreases.

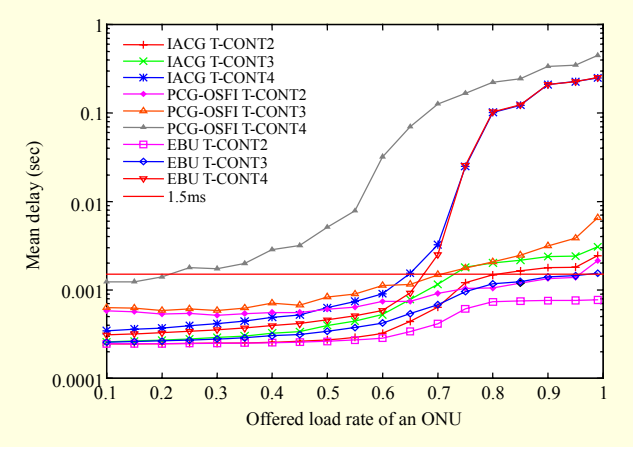

Fig. 4. Mean delay of each algorithm.

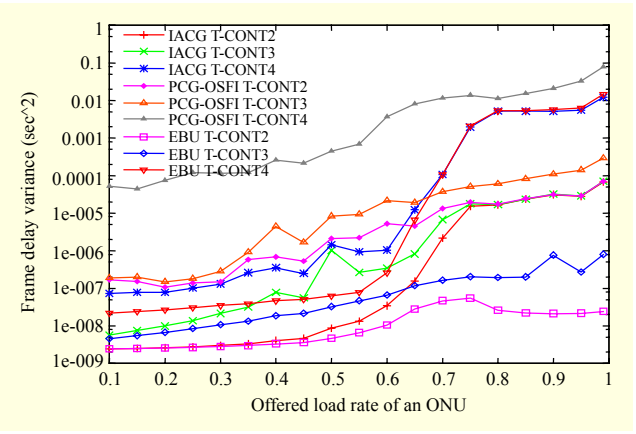

Fig. 5. Frame delay variance of each algorithm.

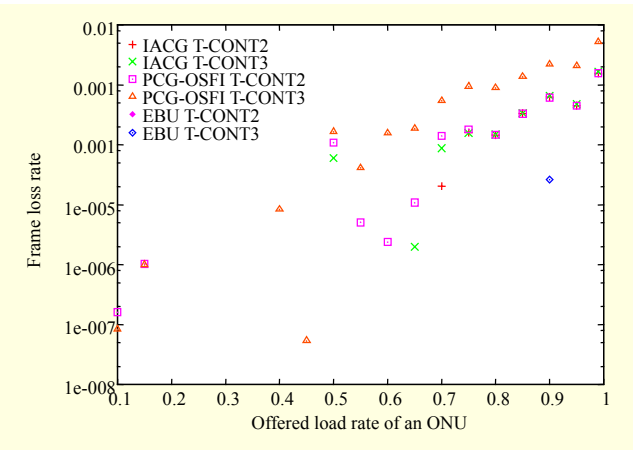

Fig. 6. Frame loss rate of each algorithm.

Hence, the efficiency of DBA improves. As the value of SI decreases, however, the upstream bandwidth waste owing to DBRu slots increases since a queue gets a DBRu slot at least once per SI, as we described in section III. The optimal value of SI will be our next research topic.

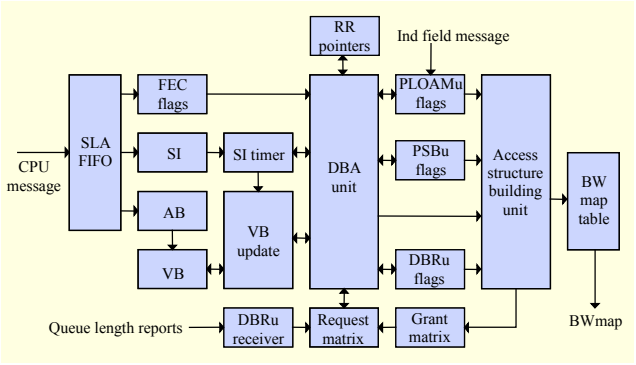

Fig. 7. Block diagram of EBU DBA module.

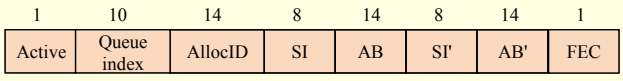

Fig. 8. SLA parameters.

VI. Hardware Implementation

We implement the EBU DBA module using a Xilinx FPGA chip. Figure 7 shows the block diagram of the EBU DBA module. The CPU stores the service level agreement (SLA) parameters of a queue into the SLA FIFO (first-in first-out). The SLA parameter is 70 bits wide, as in Fig. 8. In the EBU DBA module, a queue has a 10-bit queue index to identify its scheduling order and T-CONT type. The active bit indicates whether or not the corresponding queue is active. The forward error correction (FEC) bit indicates whether or not an FEC function is turned on for the corresponding queue.

The *SI* and *AB* modules save the *SI* and the *AB* values of queues, respectively. The *SI* timer module contains the down counters of queues, and the *VB* module contains the *VB* values of queues. The FEC flags module denotes the FEC bits of queues. These modules are implemented with the internal RAM of the FPGA chip. The *VB* update module performs the update operation of the *VBs*.

The request matrix manages the actual request size of each queue. The DBRu receiver examines the DBRu fields in the upstream frame and extracts the queue length in the field. The grant matrix remembers the eight most recent grant results to support the 60 km maximum distance between an ONU and the OLT. The RR pointers module saves the round-robin pointers of the scheduling operation.

The PLOAMu flags module represents whether or not a queue sends the physical layer operations administration and maintenance upstream (PLOAMu) slot, which conveys network management information. From the indication (Ind) field of the PSBu, the PLOAMu slot is assigned to the ONU that sent the PLOAM message. The PSBu flags module indicates whether or not the PSBu slot is required for each ONU. The DBRu flags module indicates whether or not a queue sends the DBRu slot. The DBA unit performs the EBU DBA operation and computes an FEC parity byte.

In the upstream FEC computation of an XGPON, each codeword is 248 bytes long. Within a codeword, 232 data bytes are followed by 16 FEC parity bytes [6]. In addition, in an XGPON, the frame and its elements are aligned to one-word (4 bytes) boundaries and the grant size has the granularity of one word [6]. The FEC parity bytes are expressed in the word size. Whenever the new grant size is generated, the FEC size must be recalculated. Since the FEC size consumes the upstream bandwidth, the next grant size can be affected by the FEC size. Let X be the FEC word size and D be the data word size; then, X is given by

$$X = \left\lceil \frac{D}{58} \right\rceil, \tag{4}$$

or X is the quotient of $(D + 57)/58$. The division operation consumes multiple clocks in FPGA implementation. If D is 14 bits wide, the number of required clocks for the division operation is at least 16 if we use the Xilinx divider [16]. To obtain the DBA result every 125 μs, the delay in the division operation must be minimized.

For high-speed division, we approximate (4). From $X = \left\lfloor \frac{512(D+57)}{58} \frac{1}{512} \right\rfloor$, we obtain the approximated value Y as follows:

$$Y = \frac{1}{512}\left(8D + \frac{D}{2} + \frac{D}{4} + \frac{D}{16} + \frac{D}{64} + 506\right). \tag{5}$$

When $D \le 9{,}720$ words (= 38,880 byte), we have $Y = X$. Since Y consists of only the shift and the addition operations, it can be calculated within a single clock in a commercial FPGA chip.

The access structure building unit adds the grant results for the assured and nonassured bandwidth of T-CONT type 3. Also, the unit generates the access structure using the PLOAMu flags, the DBRu flags, and the grant results. Then, the unit transfers the access structure into the BWmap table.

Figure 9 shows the XGPON OLT board that we have developed. The OTRx part includes the optical transmitter and receiver, which support 10 Gbps and 2.5 Gbps, respectively. The CPU block manages the software operation of the system. The uplink interface provides the data path between the board and the back plane. The FPGA block contains the Xilinx FPGA chip and an external memory.

We use a Xilinx Virtex-5 LX155t FPGA chip for the hardware implementation. The EBU DBA module supports a

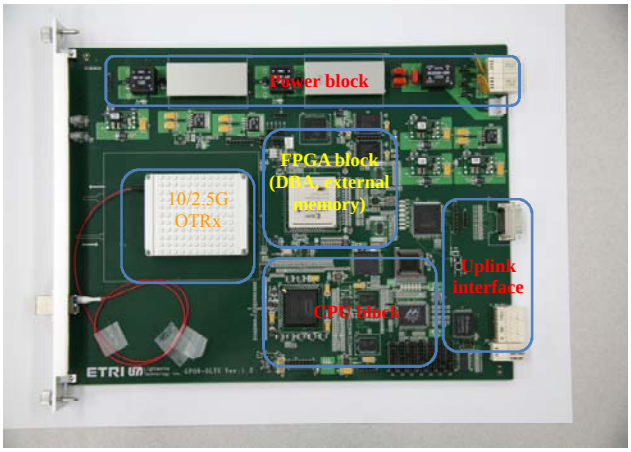

Fig. 9. XGPON OLT board.

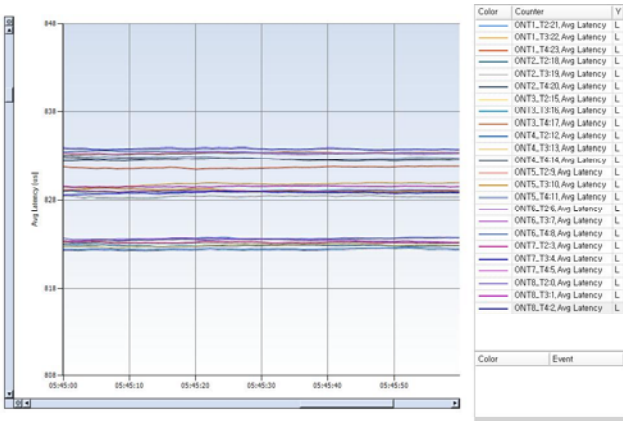

Fig. 11. Mean delay test result.

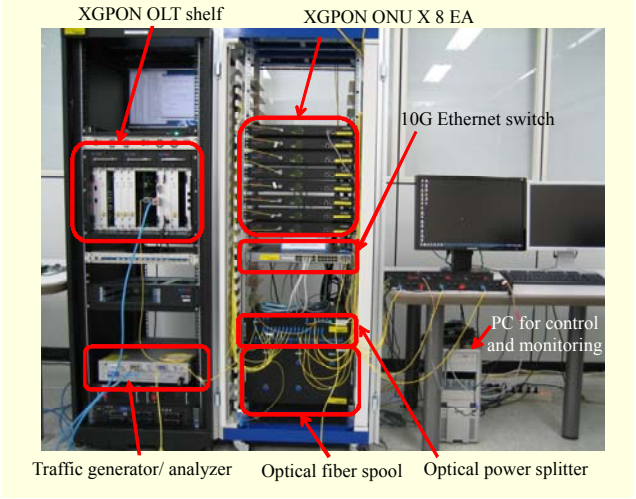

Fig. 10. Test bed for XGPON system

10-Gbps downstream speed, 2.5-Gbps upstream speed, and 256 ONUs, and each ONU has four queues (one queue for each T-CONT type, 1 through 4). The total FPGA utilization is 5% slice logic utilization and 15% internal RAM utilization, and the maximum operation frequency is 130 MHz.

VII. Hardware Test Result

Figure 10 shows the testbed that we developed to test the XGPON OLT board. The OLT shelf includes our XGPON OLT board. Currently, we have eight ONUs whose user side interfaces are connected to a 10-Gbps Ethernet switch. Additionally, the ONUs are connected to the OLT board through the optical power splitter. The optical fiber spool contains 20-km-long optical fibers.

The traffic generator/analyzer generates packets and feeds them to the Ethernet switch. The traffic generator/analyzer can generate only uniform traffic. The packet sizes are uniformly distributed from 64 bytes to 1,500 bytes. In addition, the traffic generator/analyzer collects packets transmitted from the XGPON OLT board. The PC is used for system control and to monitor test results.

The line rate from the Ethernet switch to each ONU is 200 Mbps. The input load of each T-CONT type in an ONU is evenly distributed. The SLA parameters, such as *AB* and *SI*, are identical to those used in the simulation in section V.

Figure 11 shows the captured image of the mean delay test result. In Fig. 11, ONT*i*_T*j* represents the mean delay of T-CONT type *j* in ONU*i*, where $i = 1,\ldots,8$, and $j = 2, 3, 4$. The X-axis is the current time in hour:minute:second format, and the Y-axis is the delay in μs. The result shows three groups: T-CONT type 2 is the lowest group, T-CONT type 3 is the next lowest group, and T-CONT type 4 is the highest group. Since the traffic generation model is different from that of the simulation and since the delays in the Ethernet switch and the traffic generator/analyzer are included in the packet delay, the mean delay of the test is greater than that of the simulation.

VIII. Conclusion

To utilize unused bandwidth, EBU uses the available byte counter that can be negative. EBU adds the unused amount of the available byte counter to the available negative byte counters. In addition, EBU allocates multiple polling slots to a queue during its service interval to know the request of the queue as soon as possible. Using approximate analysis, we showed that EBU is better than IACG regarding mean waiting time. Through simulations, we showed that EBU outperforms the existing methods in mean delay, delay variance, and loss rate. We implemented the EBU algorithm in a Xilinx FPGA chip and developed an XGPON OLT board that supports a 10-Gbps downstream speed, 2.5-Gbps upstream speed, 256 ONUs, and 4 T-CONT types. We developed the testbed and

described the test results of the hardware implementation.

References

[1] ITU-T Rec. G.987.1, "10 Gigabit-Capable Passive Optical Network (XG-PON): General Requirements," 2010.

[2] D. Parsons, "GPON vs. EPON Costs Comparison," BroadLight Inc., June 2005. http://www.broadlight.com

[3] B. Skubic et al., "A Comparison of Dynamic Bandwidth Allocation for EPON, GPON, and Next-Generation TDM PON," *IEEE Comm. Mag*., vol. 47, no. 3, Mar. 2009, pp. 40-48.

[4] J. Kim et al., "Compact 2.5 Gb/s Burst-Mode Receiver with Optimum APD Gain for XG-PON1 and GPON Applications," *ETRI J*., vol. 31, no. 5, Oct. 2009, pp. 622-624.

[5] ITU-T Rec. G.984.3, "Gigabit-Capable Passive Optical Networks (G-PON): Transmission Convergence Layer Specification," 2008.

[6] ITU-T Rec. G.987.3 Rev.2, "10-Gigabit-Capable Passive Optical Networks (XG-PON): Transmission Convergence (TC) Specifications," 2010.

[7] M. Mcgarry, M. Reisslein, and M. Maier, "Ethernet Passive Optical Network Architectures and Dynamic Bandwidth Allocation algorithms," *IEEE Commun. Surveys Tutorials*, vol. 10, no. 3, 2008, pp. 46-60.

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

[9] J.D. Angelopoulos, H.C. Leligou, and T. Argyriou, "Prioritized Multiplexing of Traffic Accessing an FSAN-Compliant GPON," *3rd IFIP-TC6 Netw. Conf*., Athens, Greece, 2004.

[10] H.C. Leligou et al., "Efficient Medium Arbitration of FSAN-Compliant GPONs," *Int. J. Comm. Syst*., vol. 19, no. 5, June 2006, pp. 603-617.

[11] J.D. Angelopoulos et al., "Efficient Transport of Packets with QoS in an FSAN-Aligned GPON," *IEEE Comm. Mag*., vol. 42, no. 2, Feb. 2004, pp. 92-98.

[12] K. Kanonakis and I. Tomkos, "Offset-Based Scheduling with Flexible Intervals for Evolving GPON Networks," *J. Lightw. Technol*., vol. 27, no. 15, Aug. 2009, pp. 3259-3268.

[13] M.-S. Han et al., "Efficient Dynamic Bandwidth Allocation for FSAN-Compliant GPON," *OSA J. Opt. Netw*., vol. 7, no. 8, July 2008, pp. 783-795.

[14] C.H. Chang et al., "Full-Service MAC Protocol for Metro-Reach GPONs," *J. Lightw. Technol*., vol. 28, no. 7, Apr. 2010, pp. 1016-1022.

[15] H. Takagi, *Queueing Analysis: A Foundation of Performance Evaluation: Vol. 1: Vacation and Priority Systems, Part 1*, Amsterdam: North Holland Publishing Co., 1991.

[16] Product Specification, "Divider Generator v3.0," Xilinx, June 2009.

[17] G. Kramer, *Ethernet Passive Optical Networks*, New York: McGraw-Hill, 2005.

Man Soo Han received his BS, MS, and PhD in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea, in 1992, 1994, and 1999, respectively. He was a senior researcher at ETRI, Daejeon, Rep. of Korea, from 1999 to 2003. He is an associate professor in the Department of Information and Communications Engineering at Mokpo National University, Mokpo, Rep. of Korea. His research interest includes scheduling in high-speed networks, passive optical networks, and wireless networks. He is a member of IEEE, OSA, IEICE, and KICS.

Hark Yoo received his BS in electrical engineering from Yonsei University, Seoul, Rep. of Korea, in 1998 and received his MS and PhD in communications engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea, in 2000 and 2005, respectively. His research interests are next-generation optical access networks, all-optical network systems, and optical computing devices. In 2005, he joined ETRI, Daejeon, Rep. of Korea, where he has engaged in the research and development of XGPON technology for next-generation optical access networks.

Dong Soo Lee received his BS in physics from Sogang University, Seoul, Rep. of Korea, in 1993 and received his MS and PhD in communications engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea, in 2000 and 2004, respectively. His research interests are next-generation optical access networks, high-speed optical transmission systems, and optical wireless communications. In 2005, he joined ETRI, Daejeon, Rep. of Korea, where he has engaged in the research and development of XGPON technology for next-generation optical access networks.