Iterative dynamic bandwidth allocation for XGPON

Man-Soo Han
Dept. of Information and Communication, Mokpo National Univ., Korea
mshan@mokpo.ac.kr

***Abstract*—This paper proposes an immediate allocation with reallocation (IAR) algorithm for a dynamic bandwidth allocation of 10-gigabit-capable passive optical networks (XGPONs). IAR iterates scheduling with the unused bandwidth when bandwidth remains unused after the first scheduling. Moreover, IAR assigns an additional polling bandwidth to a queue in order to improve scheduling efficiency. Using simulations, we show the proposed method improves performance compared to existing methods.**

***Index Terms*—Scheduling; XGPON; reallocation, dynamic bandwidth allocation;**

I. Introduction

Gigabit passivee optical network (GPON) is one of the major standards in access networks. GPON consists of an optical line termination (OLT), a passive splitter, and multiple optical network units (ONUs). In downstream, the OLT broadcasts frames to ONUs using the passive splitter. In upstream, ONUs transmit frames to the OLT in a time division multiplex manner. The OLT performs dynamic bandwidth allocation (DBA) to allocate non-overlapping transmission slots to ONUs.

Recently, to satisfy the high bandwidth demands, 10-gigabit-capable PON (XGPON) has been developed by extending GPON. In October 2010, ITU-T has published XGPON standards G.987.1 and G.987.3. XGPON supports the data rate of 10 Gbps in the downstream direction and the data rate of 2.5 Gbps in the upstream direction [2].

GPON standards related to DBA are described in ITU-T G.984.3 [3]. The standards are satisfied by only few GPON DBA algorithms, such as a GigaPON access network (GIANT) algorithm [9], [10], predictive-colorless-grant offset-based scheduling with flexible intervals (PCG-OSFI) [13], and immediate allocation with colorless grant (IACG) [11].

In GIANT, the OLT maintains a down counter for each queue. The OLT is allowed to allocate bandwidth to a queue only when its down counter has expired. In GIANT, performance can be degraded since a queue can not be served until its down counter expires. In PCG-OSFI, a service interval of a queue is increased if the queue can not be allocated to a pre-planned interval. In addition, the service amount of the queue is proportional to the service interval. However, the service interval can not be shorter than the pre-planned interval. Thereby, performance can be worse in PCG-OSFI.

In IACG, the OLT maintains an available byte counter and the down counter for each queue. The OLT can immediately allocate a transmission slot to a queue up to its available byte counter. The available byte counter is decreased by the allocation amount and recharged when the down counter expires. It was shown that performance of IACG is better than that of GIANT [11].

Although IACG provides good performance, it does not effectively utilize the unallocated bandwidth of queues. When a request size of a queue is less than its reserved service bandwidth, a part of bandwidth will remain unallocated. For high scheduling efficiency, the unallocated bandwidth must be utilized by queues whose request sizes are larger than their reserved service bandwidth. Also, for service fairness, the unallocated bandwidth must be distributed to the queues in proportion to their service weights. However, IACG does not distribute the unallocated bandwidth to queues by considering the request size and the service weight.

In this paper, we propose an immediate allocation with reallocation (IAR) algorithm in order to fairly utilize the unallocated bandwidth and to decrease the bandwidth waste at the same time. IAR is based on IACG but it repeats scheduling with the unallocated bandwidth if bandwidth remains unused after the first scheduling. When IAR repeats scheduling, the assignable bandwidth of a queue has a lower limit to decrease bandwidth waste. In addition, IAR assigns an additional polling bandwidth to a queue in order to improve scheduling efficiency. Using simulations, we show that IAR is superior to existing methods in mean delay and frame loss rate.

II. System Model

An XGPON system consists of an OLT and N ONUs. ONU i has multiple queues; one queue, q_{ij}, per each T-CONT type j. If an incoming frame of ONU i has the T-CONT type j, the frame is saved in the queue q_{ij}. Also each queue q_{ij} has its unique Allocation Identifier (AllocID).

In downstream, XGPON operation is synchronized with a downstream frame duration (DFD) that has a fixed length of 125 μs. In upstream, XGPON operation is synchronized with a upstream frame duration (UFD) that also has a fixed length of 125 μs. DFD and UFD are not necessarily synchronized. In each DFD, the OLT collects requests from queues in ONUs and allocate non-overlapping transmission slots to the queues for a upcoming UFD. Also, the OLT allocates a dynamic bandwidth report upstream (DBRu) slot to permit a queue to report its request. When a queue receives the DBRu slot, it reports its queue length to the OLT. Also, during scheduling, the OLT has to consider the transmission overhead such as a guard time and an upstream physical synchronization block (PSBu) section.

The ONU needs an XGPON transmission convergence (XGTC) header at the beginning of a transmission. Also the

ONU requires an XGTC trailer at the end of the transmission. In this paper, the sum of the guard time, the PSBu section, the XGTC header and the XGTC trailer is denoted as the burst overhead (BO).

A frame has an XGPON encapsulation method (XGEM) header of 8 bytes in order to indicate the flow information of the frame. When the OLT assigns a grant time-slot, its size must be grater than or equal to 16 bytes. Otherwise, the grant is ignored by an ONU [1]. In 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 pre-pended with the XGEM header. Fig. 1 depicts the burst overhead and the XGEM header.

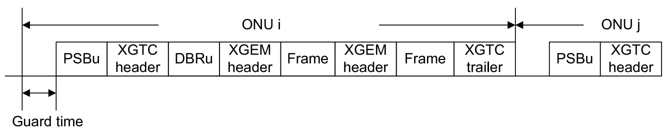

Fig. 1. Burst overhead and XGEM header

GPON and XGPON use five T-CONT types which are distinguished by their assignable bandwidth [1], [3]:

- T-CONT type 1: the fixed bandwidth.
- T-CONT type 2: the assured bandwidth.
- T-CONT type 3: the assured bandwidth and the non-assured bandwidth.
- T-CONT type 4: the best-effort bandwidth.
- T-CONT type 5: all bandwidth.

Since the T-CONT type 1 is statically served and the other T-CONT types are dynamically served, we do not consider the T-CONT type 1 in this paper as in [11], [13]. The service priority order is the assured bandwidth of T-CONT type 2, the assured bandwidth of T-CONT type 3, the non-assured bandwidth of T-CONT type 3, and the best-effort bandwidth of T-CONT 4. T-CONT type 5 is a consolidation of other T-CONT types. In this paper, we use T-CONT type 5 in order to represent the colorless grant (CG) 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.

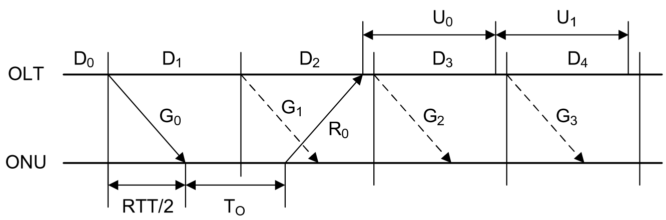

Fig. 2. Transmission timing diagram

Fig. 2 shows the operation timing diagram of GPON. D_i denotes the i-th DFD and U_i represents the i-th UFD. In Fig. 2, RTT is a round trip time between an ONU and the OLT, where the maximum of RTT is 200 μs in this paper. Also, T_O is an ONU response time to prepare an upstream response. The i-th grant result, G_i, is produced by the DBA operation at D_i. R_0 represents a request of the ONU that is the total number of frames in a queue. Since the ONU transmits R_0 before its frames, G_0 is not reflected in R_0. The OLT has to know the actual request for an efficient allocation. To do so, we use the polling mechanism of IACG [11] in this paper. The requests arrived in U_0, including R_0, will be used in the DBA at D_4. As we can see from Fig. 2, the other grants, G_1, G_2, G_3 can be produced for the ONU before the DBA at D_4. Since those grants are not reflected in R_0, the OLT subtracts $G_0, \cdots, G_3$ from R_0 before the DBA at D_4 to obtain the actual request.

III. Scheduling algorithm

Let us first introduce the basic operation of IACG in order to explain the unallocated bandwidth problem of IACG. Table I shows the service parameters and counters that are used in IACG. In Table I, SI represents the service interval with the unit of 125 μs and AB means the maximum allocation bytes that can be maximally allocated to a queue during its SI. In addition, SI_timer is a down counter which is decreased by 1 for each DFD and recharges to SI when it has expired. We use VB to represent the remaining available bytes during SI. We use SIs, ABs, SIs_timer, and VBs for the surplus bandwidth of T-CONT type 3. Each queue has its own service parameters and counters. Let queue(j) be the queue with AllocID j. The service parameters of queue(j) are represented as SI(j) and AB(j). In addition, the counters of queue(j) are represented as SI_timer(j) and VB(j).

TABLE I
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	surplus	SIs, ABs	SIs_timer, VBs
4	surplus	SI, AB	SI_timer, VB

The down counter SI_timer(j) is decreased by 1 for each DFD and recharged to SI(j) when it has expired. Also VB(j) is reset to AB(j) when its SI_timer(j) has expired. For queue(j), the OLT grants the minimum of the request of the queue and the value of VB(j). The value of VB(j) is immediately decreased by the grant amount. As a result, the grant amount for queue(j) is limited by AB(j) during SI(j).

We now explain the unallocated bandwidth problem of IACG. The problem occurs if some of queues are *under-requested* and the others are *over-requested*. Queue(j) is under-requested when its request, request(j), is less than its available bytes VB(j). Also, queue(j) is over-requested when request(j) is greater than VB(j). The unused available bytes of under-requested queues can not be used by over-requested queues in IACG. For example, assume queue(i) and queue(j) are T-CONT type 2. Also suppose request(i) = 100, VB(i) = 10, request(j) = 0, and VB(j) = 1000. In IACG, the OLT will grant only 10 for queue(i) despite that VB(j) remains unused. To receive the grant for the remainder of request(i), queue(i)

has to wait until VB(i) is recharged to AB(i). This problem degrades performance of IACG.

A similar unallocated bandwidth problem has been studied in Ethernet PON (EPON). To solve the problem, many algorithms, such as an excessive bandwidth reallocation (EBR) and a weight-based DBA (WDBA), used a reallocation method [7], [8]. The reallocation method repeats scheduling when the over-requested and under-requested queues exist. For service fairness, the reallocation method distributes the unallocated bandwidth of under-requested queues to the over-requested queues in proportion to their service weights in repeated scheduling. Let E be the total unallocated bandwidth saved by the under-requested queues and w_j be the service weight of queue(j). In repeated scheduling, the excessive bytes of an over-requested queue(j), EB(j), is given by

$$\text{EB}(j) = \frac{w_j E}{\sum_{k \in V} w_k} \tag{1}$$

where V is the set of over-requested queues. EB(j) is the maximum bytes that can be allocated to queue(j) in repeated scheduling. The OLT will grant the minimum of the remaining request and EB(j) for queue(j) in repeated scheduling.

If the reallocation method of EPON is directly used in XGPON, it can cause bandwidth waste. In repeated scheduling, if EB(j) of an over-requested queue(j) is less than 16 bytes, queue(j) may not transmit any frame [1]. Since each ONU requires the BO to transmit a data frame, the worst case happens when queue(j) is over-requested with VB(j) = 0 and its EB(j) is equal to 15 bytes. In the worst case, the bandwidth for EB(j) and the BO is wasted. For example, suppose the sum of the BO size is 48 bytes and the UFD size is 38,880 bytes. If the worst case happens to 150 queues, the total wasted bandwidth is 150 $\times$ (48+15) = 6,300 bytes. The wasted amount is more than 24% of the UFD.

A. IAR algorithm

In order to solve the unallocated bandwidth problem of IACG, we introduce an immediate allocation with reallocation (IAR). IAR is based on IACG and consists of four steps; first step for scheduling of the assured bandwidth, second step for repeating scheduling of the assured bandwidth, third step for scheduling of the surplus bandwidth, and fourth step for the allocation of colorless grant (CG). IAR is identical to IACG except the second step.

Now let us explain the second step of IAR. In IAR, the service weight of queue(j) is defined as $w_j = \frac{AB(j)}{SI(j) \cdot 125\mu s} \frac{125\mu s}{\text{UB}} = \frac{AB(j)}{SI(j) \cdot \text{UB}}$, where UB is the byte size of the UFD. Let E be the total unallocated bandwidth of under-requested queues with T-CONT types 2 and 3 after the first step. Then we have

$$E = \text{UB} \sum_{j \in A} w_j - \sum_{j \in A} \text{grant}(j) \tag{2}$$

where grant(j) is the grant amount of queue(j) at the first step and A is the set of queues with T-CONT types 2 and 3. If $E > 0$, then IAR performs the second step.

In the second step, EB(j) of an over-requested queue(j) should be

$$\text{EB}(j) = \alpha \frac{w_j E}{\sum_{k \in V} w_k} \tag{3}$$

where V is the set of over-requested queues with T-CONT types 2 and 3. In addition, the variable α is used to prevent waste of EB(j) when the size of EB(j) is less than 16 bytes. Since the minimum grant size is 16 bytes, the minimum value of α is calculated from the following relation

$$\alpha \frac{w_m E}{\sum_{k \in V} w_k} \geq 16 \tag{4}$$

where m is the AllocID of the queue which has the smallest EB, i.e., EB$(m) \leqslant$ EB(j), for all $j \in V$, and $m \in V$.

Because of α, it is possible that $\sum_{j \in V} \text{EB}(j) > E$. Thereby the OLT may not be able to allocate the excessive bandwidth to some of over-requested queues. To remedy this fairness problem, IAR changes the starting point of the second step scheduling in a round-robin manner. IAR employs a round-robin pointer RR to denote at which ONU the OLT starts scheduling in the second step. The round-robin pointer RR is increased by 1 in each second step. The OLT grants the minimum of EB(j), E and request(j) for queue(j). The granted amount is added to grant(j) and subtracted from each of E and request(j).

B. Polling mechanism

In IACG, the OLT can allocate the DBRu slot to queue(j) once per SI(j). For efficient scheduling, the OLT has to know the actual requests of queues as soon as possible. In the polling perspective, the best scheme is that every queue reports its request to the OLT in every UFD. The DBRu slot is 4 bytes long [1]. If the DBRu slots are allocated to all queues for every UFD, the upstream bandwidth for data frames will be wasted especially when the number of queues is large. IAR allocates multiple DBRu slots to queue(j) during its SI(j). However, IAR mitigates the bandwidth waste due to the excessive number of DBRu slots by allocating the DBRu slot only when a specific condition meets. In IAR, the OLT can allocate the DBRu slot to queue(j) once per SI(j) independent of the DBA result. In addition, the OLT can allocate the DBRu slot to queue(j) if grant$(j) > 0$.

C. Pseudo code

The OLT first performs the first step for the assured bandwidth of T-CONT type 2, and then for the assured bandwidth of T-CONT type 3. Then, if $E \geq 16$, the OLT executes the second step for the assured bandwidth of T-CONT type 2, and then for the assured bandwidth of T-CONT type 3. Next the OLT runs the third step for the surplus bandwidth of T-CONT type 3, then the surplus bandwidth of T-CONT type 4. Lastly the OLT performs the fourth step for the CG allocation of T-CONT type 5. Fig. 3 shows the pseudo code of the first step for the assured bandwidth of T-CONT types 2 and 3, and the third step for the surplus bandwidth of T-CONT type 4.

In Fig. 3, the round robin pointer R_i indicates the ONU number at which scheduling starts. The variable B_{BO} is the size of the BO. The variable FB is the remaining bytes of the UFD and its initial value is 38,880 when the upstream channel speed is 2.48832 Gbps. The condition $C(k, FB)$ denotes if FB is sufficient to allocate a frame byte to a queue in ONU k. That is $C(k, FB) = 1$, if $(BO(k) = 0$ and $FB \geq B_{BO} + 16)$ or $(BO(k) = 1$ and $FB \geq 16)$. Otherwise, $C(k, FB) = 0$. The variable $alloc\_end$ represents the end of the allocation. When $alloc\_end = 1$, the allocation has ended. The initial values of grant(j), BO(i), and $alloc\_end$ are zero.

```
// i = T-CONT type 2 or 3 or 4
// k = ONU number
k = stop = R_i;
while(1){
    j = AllocID of q_ki;
    if (VB(j) ≥ 16 and C(k, FB) = 1){
        if (request(j) > 0){
            FB− = (1 − BO(k)) × B_BO;
            BO(k) = 1;
            grant(j) = min(VB(j), request(j), FB);
            VB(j)− = grant(j);
            request(j)− = grant(j);
            FB− = grant(j);
        }
    } else if (alloc_end = 0 and FB ≤ B_BO + 16){
        R_i = k;
        alloc_end = 1;
    }
    SI_timer(j) − −;
    if (SI_timer(j) = 0){
        SI_timer(j) = SI(j);
        VB(j) = AB(j);
    }
    k++;
    k = k  mod N;
    if (k = stop) break;
}
```

Fig. 3. Pseudo code of IAR for first and third steps

We omit the pseudo code of the surplus bandwidth of T-CONT type 3 since it is similar to that of the assured bandwidth of T-CONT type 3. Fig. 4 illustrates the pseudo code of the second step for the assured bandwidth of T-CONT types 2 and 3. The variable UB is the byte size of the UFD and is 38,880 when the upstream channel speed is 2.48832 Gbps.

When an unallocated remainder of bandwidth exists after the surplus bandwidth allocation, IACG allocates CG to ONUs [11]. The OLT evenly distributes the unallocated remainder of bandwidth to all ONUs. Since the CG is an additional grant, using the CG, an ONU can transmit the frames arrived after the ONU reported its queue length. Since the newly arrived frames can be served by the CG without polling, performance will be improved. Each ONU uses the CG to transmit frames from its queues in the order of service priority. To distinguish CG from other grants, CG has T-CONT type 5. IAR uses the CG allocation scheme of IACG in the fourth step. The pseudo code of the CG allocation scheme can be found in [11].

```
// i = T-CONT type 2 or 3
// k = ONU number
k = stop = RR;
E = UB Σ_{j∈A} w_j − Σ_{j∈A} grant(j);
EB(j) = α (w_j · E) / (Σ_{n∈V} w_n), for all j ∈ V;
while(1){
    j = AllocID of q_ki;
    if (E ≥ 16 and C(k, FB) = 1){
        if (request(j) > 0){
            FB− = (1 − BO(k)) × B_BO;
            BO(k) = 1;
            G = min(EB(j), request(j), E, FB);
            E− = G;
            request(j)− = G;
            FB− = G;
            grant(j)+ = G;
        }
    }
    k++;
    k = k  mod N;
    if (k = stop) break;
}
if (i = 3){
    RR++;
    RR = RR  mod N;
}
```

Fig. 4. Pseudo code of IAR for second step

IV. PERFORMANCE EVALUATION

We now compare performance of IAR, IACG and PCG-OSFI. We consider an XGPON system with 16 ONUs, the line rate from users to ONU link of 200 Mbps, the upstream channel rate of 2.48832 Gbps, the maximum RTT of 200 μs, and the ONU response time of 35 μs. The size of a queue q_{ij} is 1 Mbytes. We suppose traffic is balanced so that each ONU has an identical load. In addition, we suppose that each T-CONT in an ONU has a uniformly distributed traffic load.

For the T-CONT type 2, we set AB = 7812, SI = 5, which is equivalent to 100 Mbps. For the T-CONT type 3, we set AB = 7812, SI = 10, ABs = 7812, and SIs = 10. That is, 50 Mbps is given to each the assured bandwidth and the surplus bandwidth of the T-CONT type 3. For the T-CONT type 4, AB = 15624, and SI = 10, which is equivalent to 100 Mbps. The total sum of the assured bandwidth is 16(100 Mbps + 50 Mbps) = 2.4 Gbps which is less than the upstream channel rate. The reserved bandwidth for each T-CONT type of PCG-OSFI is equal to that of IAR. In the second step of IAR, α is calculated from the relation $EB(j) > 7812$ for T-CONT type 2 and $EB(j) > \frac{7812}{2}$ for T-CONT type 3.

We use the self-similar traffic model of [12] where 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. Also, the frame size follows the tri-modal distribution [12], where the frame sizes are 64, 500, and 1500 bytes and their load fractions are 60%, 20% and 20%, respectively as in [9]. Each simulation is performed until the total number of frames transmitted by ONUs exceeds 10^9 for each algorithm.

Increasing the ONU load rate from 0.2 to 0.99, we simulate the algorithms and compare their performance. Figs. 5 and 6 illustrate the mean delay of each method. Note that the offered load means the input traffic load of an ONU. As we can see from Figs. 5 and 6, IAR outperforms other methods in mean delay of T-CONT types 2 and 3. ITU-T G.987.1 recommends that an XGPON system must accommodate services that require a maximum mean signal transfer delay of 1.5 ms [2]. In our simulation scenario, only IAR satisfies the requirement for T-CONT type 2 in all traffic loads.

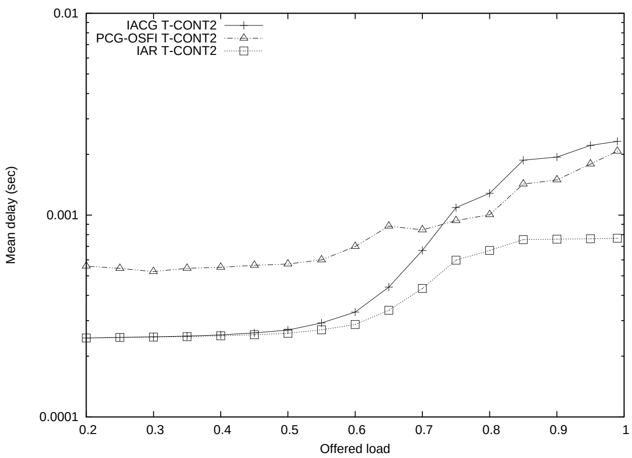

Fig. 5. Mean delay of T-CONT type 2

The PCG-OSFI method is better than IACG in mean delay of T-CONT type 2 when traffic is heavy. However, PCG-OSFI is significantly worse in mean delay of T-CONT types 3 and 4 in all traffic loads. The main reason is that PCG-OSFI inherits the drawback of GIANT in the surplus bandwidth allocation. Like GIANT, PCG-OSFI can assign the surplus bandwidth to the queues with T-CONT types 3 and 4 only when their down counters have expired [13]. Also, unlike IACG and IAR, the service interval can not be shorter than the predetermined value in PCG-OSFI. Thereby, the mean delay is worse than other methods when traffic is light.

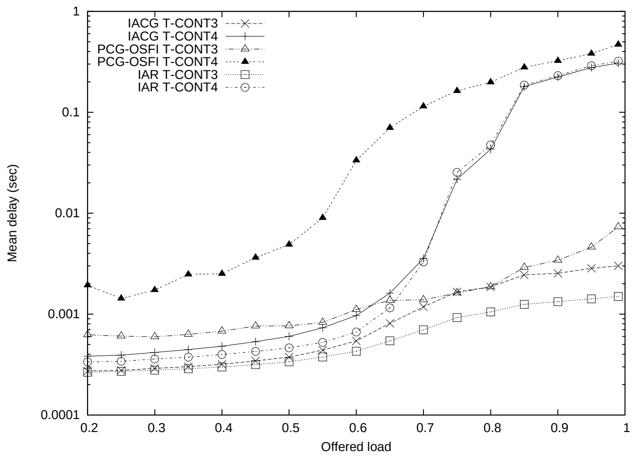

Fig. 6. Mean delay of T-CONT types 3 and 4

Figs. 7 and 8 show the frame loss rate of each algorithm. Thanks to the reallocation scheme and the additional DBRu scheme, the loss rate of IAR is zero in T-CONT type 2. However, the loss rate of IAR for T-CONT type 4 is increased compared that of IACG.

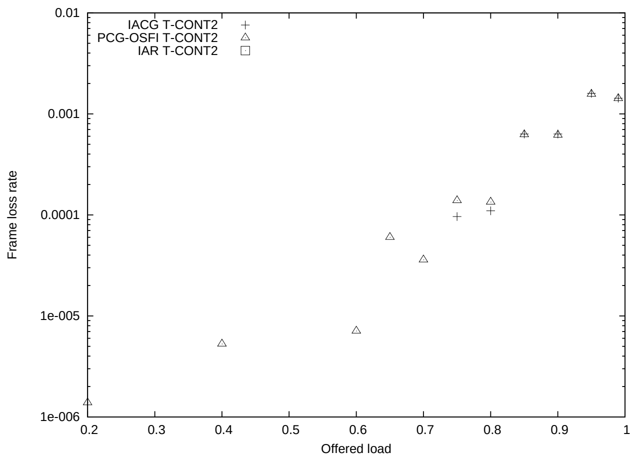

Fig. 7. Frame loss rate of T-CONT type 2

V. Conclusions

We have proposed IAR to improve the scheduling efficiency of IACG. IACG can not utilize the unused bandwidth of under-requested queues for the bandwidth allocation of over-requested queues. In order to mitigate the drawback of IACG, IAR repeats scheduling to distribute the unused bandwidth of under-requested queues to over-requested queues. In addition, to improve scheduling efficiency and to minimize the burst overhead, IAR additionally allocates polling bandwidth to a queue when the burst overhead is already allocated to the queue. Using simulations, we have shown IAR is superior to existing algorithms in mean delay and frame loss rate.

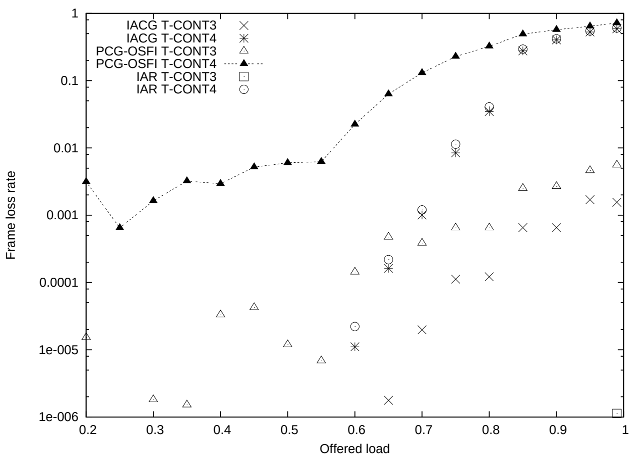

Fig. 8. Frame loss rate of T-CONT types 3 and 4

REFERENCES

[1] ITU-T Rec. G.987.3, "10-Gigabit-capable passive optical networks (XG-PON): Transmission convergence (TC) specifications," 2010.
[2] ITU-T Rec. G.987.1, "10 Gigabit-capable passive optical network (XG-PON): General requirements," 2010.
[3] ITU-T Rec. G.984.3, "Gigabit-capable passive optical networks (G-PON): Transmission convergence layer specification," 2008.
[4] ITU-T Rec. G.984.1, "Gigabit-capable passive optical networks (G-PON): General characteristics," 2008.
[5] D. Parsons, "GPON vs. EPON Costs Comparison," http://www.broadlight.com, BroadLight Inc., June 2005.
[6] J. D. Angelopoulos, H. C. Leligou, and T. Argyriou, "Prioritized multiplexing of traffic accessing an FSAN-compliant GPON," Third IFIP-TC6 Networking Conference, Arthens, Greece, 2004.
[7] M. Mcgarry, M. Reisslein, and M. Maier, "Ethernet passive optical network architectures and dynamic bandwidth allocation algorithms," IEEE Communications Surveys & Tutorials, vol. 10, no. 3, pp. 46-60, 2008.
[8] J. Zheng, and H. T. Mouftah, "A survey of dynamic bandwidth allocation algorithms for Ethernet Passive Optical Networks," Optical Switching and Networking, vol. 6, no. 3, pp. 151-162, July 2009.
[9] H. C. Leligou, Ch. Linardakis, K. Kanonakis, J. D. Angelopoulos, and Th. Orphanoudakis, "Efficient medium arbitration of FSAN-compliant GPONs," Int. J. Comm. Syst., vol. 19, no. 5, pp. 603-617, June 2006.
[10] J. D. Angelopoulos, T. Argyriou, S. Zontos, and T. Van Caenegem, "Efficient transport of packets with QoS in an FSAN-aligned GPON," IEEE Comm. Mag., vol. 42, no. 2, pp. 92-98, Feb. 2004.
[11] M.-S. Han, H. Yoo, B.-Y. Yoon, B. Kim, and J.-S. Koh, "Efficient dynamic bandwidth allocation for FSAN-Compliant GPON," OSA J. Opt. Netw., vol. 7, no. 8, pp. 783-795, Jul. 2008.
[12] G. Kramer, Ethernet passive optical networks, McGraw-Hill, 2005.
[13] K. Kanonakis and I. Tomkos, "Offset-Based Scheduling With Flexible Intervals for Evolving GPON Networks," Journal of Lightwave Technology, vol. 27, no. 15, pp. 3259-3268, Aug. 2009.
[14] C. H. Chang, N. M. Alvarez, P. Kourtessis, R. M. Lorenzo, and J. M. Senior, "Full-Service MAC Protocol for Metro-Reach GPONs," Journal of Lightwave Technology, vol. 28, no. 7, pp. 1016-1022, April 2010.