Analyzing the Performance of Centralized Polling *for Long-Reach Passive Optical Networks*

Habib Fathallah, *Member*, *IEEE*, *OSA*, Ahmed Helmy, *Student Member IEEE*
Electrical Engineering Department, and
KACST-Technology Innovation Center RFTONICS
King Saud University, Riyadh, Saudi Arabia
hfathallah@ksu.edu.sa

***Abstract*—Long-Reach passive optical networks (LR-PONs) suffer from extremely long propagation delays that degrade the performance of centralized algorithms proposed for upstream bandwidth-allocation in traditional PONs. In this paper, we analyze the performance of two centralized dynamic bandwidth allocation algorithms; so called online interleaved polling and offline multi-thread polling, recently proposed in the literature for LR-PON. We investigate their performances in terms of packet pre-transmission delay. Simulation results show that although multi-thread polling succeeds in decreasing reporting and queuing delays, interleaved polling keeps a lower grant delay and therefore has better overall delay performance.**

***Index Terms*—Dynamic bandwidth allocation (DBA); long-reach passive optical networks (LR-PONs); optical access network.**

I. Introduction

With the widespread deployment of passive optical networks (PONs) [1], research focus shifted to their scalability, with longer reaches and higher split ratios. Extending the reach of the network is thought to reduce the number of central offices (Cos) premises located near the customers allowing network operators to consolidate multiple COs and share the *optical line terminal* (OLT) optics and electronics among a larger number of users while also simplifying the fiber management [2]. The concept of a *long-reach PON* (LR-PON) was thus proposed as a more cost-effective solution for broadband optical access networks [2, 3]. By exploiting both optical amplifiers and *wavelength-division multiplexing* (WDM), LR-PON extends the coverage of PON from the traditional 20 km to 100 km and beyond while being capable of increasing the split ratio from 64 up to 1024 and more. With such extended geographic coverage, LR-PON can, at some locations, combine optical access and metro networks into a single integrated system, see Fig. 1.

This work was supported in part by the NPST program of King Saud University under project 09-ELE885-02.

LR-PONs come with many research challenges, one of which is the upstream bandwidth allocation problem under the increased propagation delays between the OLT and ONUs. Despite the various architectures of LR-PONs, the logical connection between the OLT and ONUs remains the same, as shown in Fig. 2. All downstream transmissions (from the OLT to ONUs) are done in broadcast-basis since it is a point-to-multipoint network. In the upstream, however, the network is a multipoint-to-point network; multiple ONUs transmit toward the OLT through a common passive *star coupler* (SC) also known as the splitter/combiner. Some *media access control* (MAC) mechanism is therefore required to fairly coordinate the ONUs' transmissions and avoid data collisions, especially when data from an ONU can only reach the OLT with other ONUs unable to listen due to the directional property of the combiner.

Time-division multiple access (TDMA) has been adopted in both PON standards; *Ethernet PON* (EPON) and *Gigabit PON* (GPON), to share the optical capacity among subscribers by assigning different timeslots for each user. Centralized *dynamic bandwidth allocation* (DBA) has been used [4, 5], in which the OLT at the CO arbitrates time division access to the shared upstream channel. The performance of centralized allocation depends on the *round-trip time* (RTT) since it imposes a delay on the OLT-ONUs bandwidth allocation control loop. This delay is not significant in traditional PONs as it is in LR-PONs, where the reach extension introduces challenges to the MAC-layer as the RTT may grow from today's 200μs (20 km reach) to 1ms (100 km reach). With this increase, the performance of centralized DBA is ultimately degraded.

This paper is organized as follows. In Section II we introduce two important centralized dynamic bandwidth allocation techniques including interleaved and multithread polling. Section IV presents numerical results and analysis.

II. Centralized Dynamic Bandwidth Allocation

Due to the bursty nature of web traffic, static allocation of bandwidth in a TDMA PON, by assigning fixed timeslots for each ONU, is typically inefficient. Statistical multiplexing, on the other hand, is more efficient since it adapts to instantaneous bandwidth demands. Dynamic bandwidth allocation could then be defined as providing statistical multiplexing of resources among the ONUs,

978-1-4673-1950-8/12/$31.00 ©2012 IEEE

which is essential for achieving high upstream bandwidth utilization in TDMA PON.

Centralized arbitration schemes have been adopted in traditional TDMA PONs, in which the OLT arbitrates time-division access to the shared upstream channel [8, 9]. In order for the OLT to make accurate timeslot assignments, it needs to know the exact buffer state of a given ONU. Conventionally, *polling* schemes have been used, which are based on *report* and *grant* messages defined in both EPON and GPON standards. In a polling scheme, the OLT keeps a polling table with an entry for each ONU to record both its RTT and its buffer status. Each ONU sends a report message informing the OLT of the amount of buffered data it needs to send. The OLT continuously updates its polling table, processes requests, and polls ONUs granting them transmission windows (timeslots) via grant messages. Therefore, in polling schemes, the OLT is able to know the buffer status of each ONU and allocates the ONU upstream bandwidth according to its bandwidth demand. The ONUs do not need to monitor the network state nor negotiate new parameters, which makes them simple and less expensive.

DBA involves two coexisting processes; determining the timeslot sizes (grant sizing) and determining their times/order (grant scheduling). The two processes can be carried out in two ways; by either offline or online scheduling [5]. In offline scheduling, which is also known as collective scheduling, the DBA agent has to first collect all the report messages before grant sizing and scheduling. This enables a globally optimized decision, by which any excess bandwidth from lightly loaded ONUs can be fairly distributed among heavily loaded ONUs while also being granted in an optimized order. However, using a primitive offline centralized scheduling scheme will result in an idle time of one RTT (i.e. walk time), which is the time from collecting all reports till the first grant reaches an ONU. In online scheduling, the DBA decision is made straightaway based solely on the received report. In this type of scheduling, ONUs are typically granted media access in a round-robin fashion.

A. Interleaved Polling

Interleaved polling with adaptive cycle time (IPACT) has been a pioneer online centralized algorithm for bandwidth allocation in EPON [7]. It employs a pipelined timeslot-assignment allowing the OLT to send a grant message to the next ONU before data and report messages arrive from previously polled ONUs. This is feasible since upstream and downstream channels are separated, and since the OLT maintains relevant information about each ONU in the polling table. The OLT employs *service level agreements* (SLAs) of end users by setting an upper-bound to the allocated bandwidth (grant size) of each ONU. To do so, several grant sizing schemes were investigated in [11], of which the limited service discipline was found to exhibit the best performance. In limited service, the OLT grants an ONU the number of bytes requested, but no more than a certain maximum;

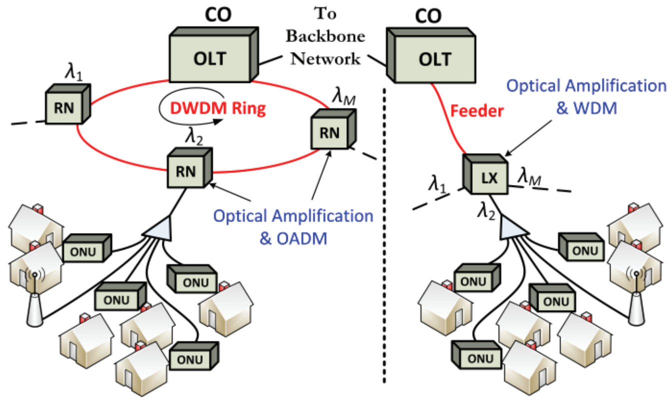

Fig. 1. Basic LR-PON architectures: ring-and-spur (left) and tree-and-branch (right).

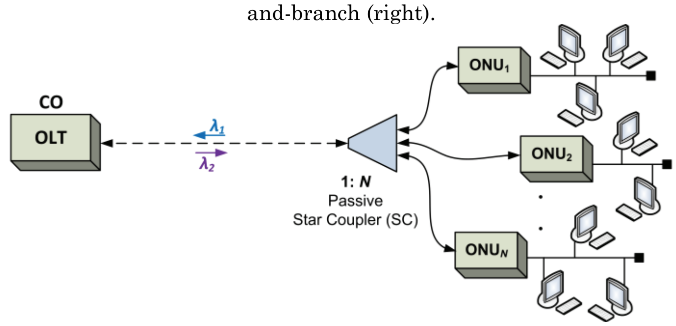

Fig. 2. Logical connection between OLT and ONUs in a LR-PON.

where R_i is the requested transmission window for ONU_i, whereas $W_{i,max}$ and W_i are its maximum allowed window and its granted one, respectively. Such a scheme is the most conservative as it assumes that no packets arrive at the ONU from the time it sends its request till it receives a grant and it therefore has the shortest cycle duration [7].

Polling protocols operate in cyclic basis with each ONU polled once per cycle and allocated a transmission window based on its demand. The maximum polling cycle C_{max} is related to the maximum transmission windows for the N ONUs by:

$$C_{\max} = \sum_{i=1}^{N} W_{i,\max} + NT_g \quad (2)$$

where T_g is a guard interval between successive ONU transmissions. The actual cycle duration however changes with the dynamics of the upstream traffic. Under light traffic loads, the cycle duration usually reduces to much less than C_{max}, which has the effect of giving ONUs in demand more bandwidth than guaranteed, and thereby reduces the average packet delay. However, the cycle cannot be less than the maximum RTT since each polling table entry should be updated within the OLT before issuing a grant to the corresponding ONU [7]. Under heavy traffic, the cycle duration reaches its maximum limit, giving ONUs their minimum guaranteed bandwidths. We therefore define the *effective cycle duration* (C_{eff}) to be the sum of the granted transmission windows within a cycle;

$$C_{eff} = \sum_{i=1}^{N} W_i + NT_g \leq C_{\max} \quad (3)$$

where W_i is the granted transmission window for ONU_i corresponding to its buffer status reported to the OLT. This expression however is ideal, as it assumes no idle intervals

exist inside the cycle other than the guard intervals. Thus the polling cycle adapts to the network load, but it is lower-bounded by the maximum RTT.

In the following we describe the different types of delays a packet may experience from the time of its arrival till the time of its transmission in a centralized polling scheme; reporting, grant, and queuing delays.

1) *Reporting Delay*: Buffered data may not be immediately reported upon entering an ONU. Instead, the report message has to be sent during the ONU's next transmission window when it is polled. This imposes a packet delay dependent upon the cycle duration. It can either be zero, if the packet arrives right before sending a report, or it can be the whole cycle duration if the packet had just missed the report. The average reporting delay is therefore half the cycle duration; $C_{eff}/2$.

2) *Grant Delay*: After the report is sent, the ONU has to wait for a grant to start transmitting. This delay cannot be less than the RTT, which makes it dependent on the OLT-ONU distance. Therefore, the longer this distance, the more significant the grant delays.

3) *Queuing Delay*: When an ONU finally receives a grant message, it starts transmitting packets one after another within its granted transmission window. Packets therefore experience queuing delays proportional to the window size.

It is worth emphasizing that these three types of delay are *pre-transmission delays*; delays that occur prior to transmission. Once packet transmission starts, two more delays are introduced; a *transmission delay*, depending on both the packet length and upstream transmission rate, and a *propagation delay*, depending on the distance between the ONUs and the OLT. Pre-transmission delays depend on the DBA algorithm used, whereas the other delays depend on the network reach, rate, and packet lengths regardless of the DBA algorithm and are therefore irreducible. Thus, the main objective is always to reduce pre-transmission delays.

It can be seen that grant delays in fact contain both transmission and propagation delays. This makes pre-transmission delays of centralized DBA also sensitive to and affected by extending the reach of the network. The increased propagation delay leads to increasing the DBA response time, which causes the average delay to increase. The performance degradation due to extending the network reach is illustrated in Fig. 3 at different network loads, for both average pre-transmission delays (Pre-D) and average packet delays (Avg-D) of IPACT. The curves are obtained from simulations with varying the distance. Average pre-transmission delays are shown to be always above 1.5RTT whereas average total delays are above twice the RTT.

B. Multi-thread Polling

In [8], a novel algorithm was proposed to handle bandwidth allocation in LR-PONs and possibly decrease high packet delays. It was suggested that the offline polling technique could be used with some modification. The idea was to utilize the idle period by creating one or more new threads of signaling between the ONUs and the OLT. Each new thread would have its own report and grant messages, thereby creating multiple offline polling processes (threads)

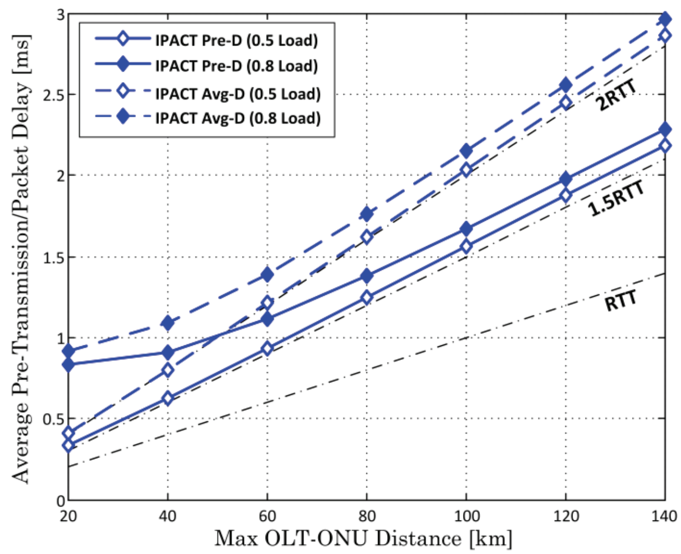

Fig. 3. Degradation of IPACT due to network extension.

running in parallel.

Comparing such a multi-thread polling with interleaved polling, i.e. IPACT, one can view multi-thread polling as interleaving entire offline threads rather than interleaving ONU polls. Although IPACT can be viewed as single thread polling [8], single thread polling is not necessarily IPACT. Single thread polling can either be online, such as IPACT, or it can be offline, which is sometimes also called interleaved polling with stop [4, 5].

It was argued in [8], that in order for single-thread polling to achieve fairness among ONUs, it had to be offline. This is not very accurate, since online polling achieves fairness among ONUs by allocating each ONU up to its maximum transmission window ($W_{i,\max}$) within each cycle, which constitutes the ONU's minimum guaranteed bandwidth. What the authors probably meant by fairness, was optimizing the bandwidth allocation by giving heavily loaded ONUs more than their guaranteed bandwidth, which can be done by distributing excess bandwidth from lightly loaded ONUs among heavily loaded ones. Indeed, this can be fairly achieved by offline polling, since it is a collective DBA scheme, where the OLT first collects all reports before allocating bandwidth with global knowledge. In that case, multi-thread polling would be better than offline single thread polling, since it utilizes the latter's long idle period.

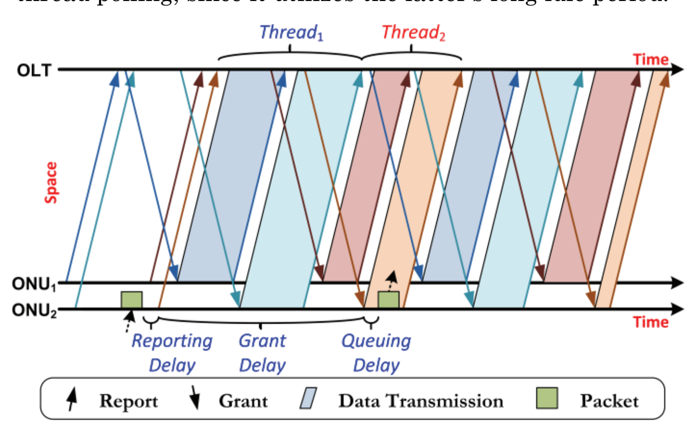

Fig. 4. Multi-thread polling.

Splitting the cycle duration into multiple threads reduces both reporting and queuing delays. This comes from the fact that when more threads are used, a packet will wait less for it to be reported. For instance, using M threads means that there are M reports within the cycle. Therefore, on average, the reporting delay would become $C_{eff}/2M$. Each thread will also have less data to transmit compared with single thread polling. Therefore, the average queuing delay would be $1/M$ of the average queuing delay of a single thread. Grant delays however are still present in multi-thread polling and cannot be less than the RTT. In fact, grant delays in any offline scheme, whether it is a single-thread or multiple, are likely to become longer than the grant delays of an online scheme such as IPACT. This is because the bandwidth allocation, in an offline scheme, is carried out at the end of the thread when all its reports have arrived, whereas, in an online scheme, an ONU is granted a timeslot straightaway, so it can transmit as soon as the preceding ONU finishes.

Although using M threads decreases reporting and queuing delays of a single-thread by a factor of M, the number of threads M cannot be increased indefinitely. This is because increasing the number of threads increases the DBA complexity, which grows linearly relative to single-thread polling. Moreover, a larger number of threads results in smaller transmission durations within a cycle, since it decreases the available bandwidth for data transmission by using more bandwidth for report messages and guard intervals. This tradeoff calls for an optimal number of threads, which was chosen in [8] to be three.

III. Performance Analysis

In this section, we first compare the pre-transmission delay components of the discussed centralized dynamic bandwidth allocation algorithms; interleaved polling and multi-thread polling. Next, we compare together their overall delay and throughput performances. Finally, we show the network extension effect on both of them.

In our study, we consider a 100 km LR-PON access network consisting of an OLT and 16 ONUs. The ONUs are placed randomly in the last 5 km of the 100 km span. They share an upstream wavelength of 1Gbps, whereas from the access side the end-users have an access rate of 100 Mbps. The system throughput is therefore less than the peak aggregated load from all ONUs. Each ONU has a finite memory buffer of 10 Mbytes. The traffic model used is self-similar Ethernet traffic with a Hurst parameter of 0.8. The traffic was generated by the model described in [9], in which multiple sub-streams of alternating Pareto-distributed on/off periods are aggregated together. In order to compare between the different schemes, the max cycle duration C_{max} is set to 5ms for all schemes, which is found to be very convenient for a 100 km span. The inter-transmission guard interval T_g is set to 5μs [7]. For multi-thread polling, we use the same parameters used in [8]. The initial interval of threads is set to be 0.5ms and 0.3ms when using 2 and 3 threads, respectively, which is max(RTT)/M [8]. Inter-thread scheduling is enabled, and thread tuning is applied with a threshold T_{tune} of 5 [8].

Figure 7 shows the average reporting delays of online interleaved polling (IPACT) and offline (collective) polling

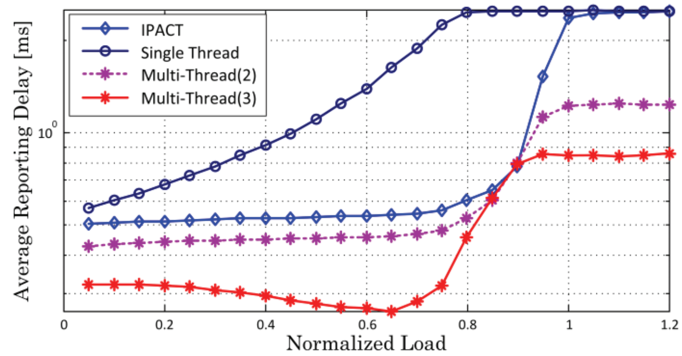

Fig. 5. Average reporting delays.

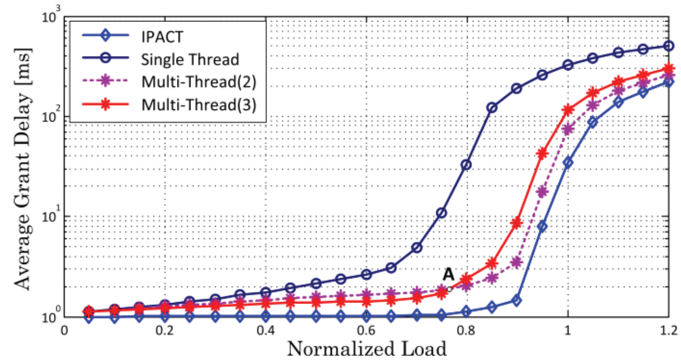

Fig. 6. Average grant delays.

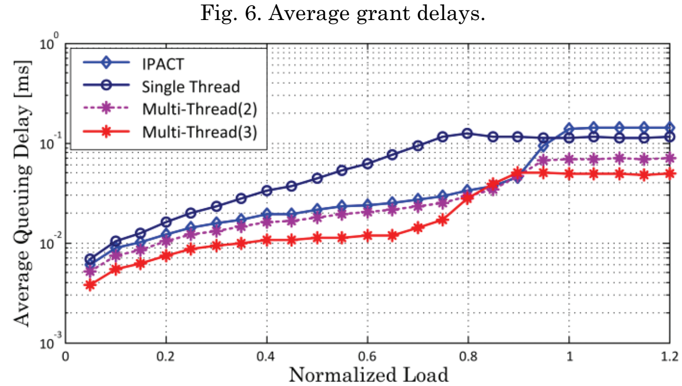

Fig. 7. Average queuing delays.

when using 1–3 threads. Using multiple threads clearly decreases the reporting delay since it increases the number of reports within the cycle. We also notice an interesting phenomenon in the reporting delays of multi-thread polling with 3 threads; as the network load increases from light to moderate, the reporting delays slightly decrease, and then they increase again under heavy loads. This can be explained by considering the effective cycle duration as the following. At light loads, the threads almost carry no data and are eventually squeezed together at the beginning of the cycle. The threads altogether will have a total duration much less than the minimum cycle duration (RTT). This means that an ONU basically ends up sending 3 consecutive reports at the beginning of each cycle and then waits for the remainder of the cycle. Therefore, packets received in this remainder will likely have long reporting delays. As the load increases, the threads begin to carry more packets and spread within the cycle, thereby distributing the reports in the cycle and decreasing the reporting delays. As the load continues to increase, the

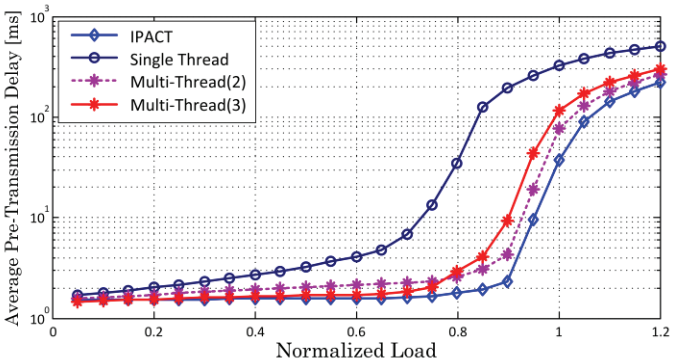

Fig. 8. Average pre-transmission delays.

threads cause the cycle to expand making the reporting delays increase again. This behavior is not significant when using only 2 threads, because there are only 2 reports per cycle, and because by the time the reports get distributed within the cycle, the cycle duration is already grown. The average reporting delays are found to be half the average effective cycle duration. Under heavy loads, the average reporting delays for all schemes is shown to be $C_{max}/2M$.

Figure 8 shows the average grant delays, which are shown to be always greater than the RTT (1ms) and are the most significant within centralized pre-transmission delays. The grant delays of the collective DBA algorithms are found to be longer than those of online interleaved polling. For multi-thread polling, using 3 threads was found to be better when the network load is below 80%, but worse than using 2 threads at higher loads (see crossover-point A). This can be explained as follows. Under normal loads, the cycle duration is less than its maximum and a packet is typically granted within one cycle. So, with 3 threads, there are more reports and grants within the cycle, which reduces grants delays. Also, recall that in an offline scheme the bandwidth allocation is carried out at the end of each thread. So, it is done relatively sooner when using 3 threads, as the threads are smaller. However, as the network load increases, the cycle duration soon reaches its maximum and a packet may have to wait multiple cycles before being granted. At this point, more bandwidth is wasted on guard intervals and report messages when using 3 threads. The performance is therefore better when using 2 threads, as more bandwidth is saved for data transmission, enabling more packets to be granted sooner.

Finally, the average queuing delays are show in Fig. 8, for which increasing the number of threads proved also to decrease queuing delays. Note also how the average queuing delays of online polling (IPACT) are greater than those of offline single thread polling for the same cycle duration. This can be misinterpreted that online polling is worse, when in fact it is quite the opposite that is true. The reason for the higher queuing delays for online polling is because it has wider transmission windows and hence longer transmission queues. Offline polling has an idle period of one RTT in each cycle, making its transmission windows smaller and transmission queues shorter.

IV. Conclusion

In this paper, we addressed the problem of bandwidth allocation in LR-PON, which suffers from long propagations delays. We studied the performance of two centralized dynamic bandwidth allocation algorithms; online interleaved polling (IPACT) and collective multi-thread polling, which was recently proposed in the literature for LR-PONs. We pointed out some inaccuracies in the multi-thread allocation algorithm and suggested some modifications to ensure that an ONU is not granted more than requested, by efficiently distributing the excess bandwidth among heavily loaded ONUs, and also to enable inter-thread scheduling to fully utilize a thread without disturbing the thread tuning. Simulation results show that multi-thread polling succeeds in decreasing reporting and queuing delays, whereas online interleaved polling has a lower grant delay and therefore achieves a better overall delay performance. Online interleaved polling also achieves a higher throughput, since multi-thread polling uses more bandwidth for report messages and guard intervals.

References

[1] F. Effenberger, J. Kani, and Y. Maeda, "Standardization trends and prospective views on the next generation of broadband optical access systems," *IEEE J. on Selected Areas in Comm.*, vol. 28, no. 6, pp. 773-780, Aug. 2010.

[2] L. Shi, S. Lee, H. Song, and B. Mukherjee, "Energy-Efficient long reach passive optical network," *J. of IEEE Systems*, vol. 4, no. 4, Dec. 2010.

[3] D. Payne and R. Davey, "Long reach access networks—Avoiding the bandwidth-price dilemma," in *Proc. Broadband Europe*, Dec. 2004.

[4] J. Zhenga and H. Mouftah, "A survey of DBA algorithms for Ethernet passive optical networks," *Optical Switching and Networking*, vol. 6, pp. 151-162, Mar. 2009.

[5] M. McGarry, M. Reisslein, and M. Maier, "Ethernet passive optical network archetectures and dynamic bandwidth allocation algorithms," *IEEE Comm. Serveys and Tutorials*, vol. 10, no. 3, pp. 151-162, Sep. 2008.

[6] S. Sherif, A. Hadjiantonis, G. Ellinas, C. Assi, and M. Ali, "A novel decentralized Ethernet-based PON access architecture for provisioning differentiated QoS," *J. of Lightwave Technology*, vol. 22, no. 11, pp. 2483-2497, Nov. 2004.

[7] G. Kramer, B. Mukherjee, and G. Pesavento, "IPACT: A dynamic protocol for EPON," *IEEE Comm. Mag.*, vol.40, no. 2, pp. 74-80, Feb. 2002.

[8] H. Song, B. Kim, and B. Mukherjee, "Multi-Thread polling: A dynamic bandwidth distribution scheme in Long-Reach PON," *IEEE Journal on Selected Areas in Communication*, vol. 27, no. 2, pp. 134-142, Feb. 2009.