Interleaved Polling Versus Multi-Thread Polling for Bandwidth Allocation in Long-Reach PONs

Ahmed Helmy, Habib Fathallah, and Hussein Mouftah

***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. This is because these algorithms are based on bandwidth negotiation messages frequently exchanged between the optical line terminal in the central office and optical network units near the users, which become seriously delayed when the network is extended causing the performance to degrade. In this paper, we review and analyze two centralized dynamic bandwidth allocation algorithms, online interleaved polling and offline multi-thread polling that was recently proposed in the literature for LR-PONs. We investigate and compare their performances together in detail, by studying and observing their elemental delays. Unexpectedly, simulation results show that, although multi-thread polling succeeds in decreasing reporting and queueing delays, interleaved polling keeps a lower grant delay and therefore has better overall delay performance. The latter also achieves better throughput compared to multi-thread polling.**

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

I. Introduction

The explosive increase in bandwidth demand has led to new access network architectures that bring the high capacity of optical fibers into access networks and closer to residential homes and small businesses. Passive optical networks (PONs) have been widely chosen for deploying optical access networks. This is because a PON uses no active elements in the field, but only employs passive optical components such as fibers and splitters, thereby saving much of the operational costs. The central office (CO) equipment is also shared by a large number of customers without deploying as much fiber as with point-to-point architectures. Compared to other access technologies, PONs provide higher bandwidths at a relatively lower cost and allow longer distances between customers and the CO, while enabling easier upgrades to higher bit rates [1].

With the widespread deployment of PONs [2], research focus has shifted to their scalability, with longer reaches and higher split ratios. Extending the reach of the network is thought to reduce the number of CO 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 [3]. The concept of a long-reach PON (LR-PON) has thus been proposed as a more cost-effective solution for broadband optical access networks [3,4]. By exploiting both optical amplifiers and wavelength-division multiplexing (WDM), an LR-PON extends the coverage of a 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, an LR-PON can, at some locations, combine optical access and metro networks into a single integrated system.

There have been many suggested architectures and demonstrations for LR-PONs in the literature [5,6]. Most architectures can be characterized under 'ring-and-spur' and 'tree-and-branch' architectures, depending on whether the CO is connected to the access zones by a long trunk fiber or by a WDM ring [7]. The tree-and-branch architecture has a feeder section of a strand of fibers (tree) that is split to multiple users (branches) at the local exchange (LX), while the ring-and-spur architecture has a feeder composed of a fiber ring and optical add–drop multiplexers (OADMs) placed in remote nodes (RNs) [7]. In both architectures, several PONs are multiplexed into the shared fiber and served together by a single CO, with each PON segment using different upstream and downstream wavelengths to communicate with the CO, as illustrated in Fig. 1. Most demonstrations put the optical network units (ONUs) within a 10 km drop section of the LX/RN.

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 the ONUs. Despite the various architectures of LR-PONs, the logical connection between the OLT and the ONUs remains the same, as shown in Fig. 2. All downstream transmissions (from the OLT to the ONUs) are made 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

Manuscript received May 2, 2011; revised November 21, 2011; accepted January 16, 2012; published February 14, 2012 (Doc. ID 146789).

Ahmed Helmy (e-mail: ahelmy1@ksu.edu.sa) and Habib Fathallah are with the Prince Sultan's Advanced Technologies Research Institute (PSATRI) and the Electrical Engineering Department, King Saud University (KSU), Riyadh 11421, Saudi Arabia.

Hussein Mouftah is with the School of Information Technology and Engineering, University of Ottawa, Ottawa, Ontario K1N 6N5, Canada.

Digital Object Identifier 10.1364/JOCN.4.000210

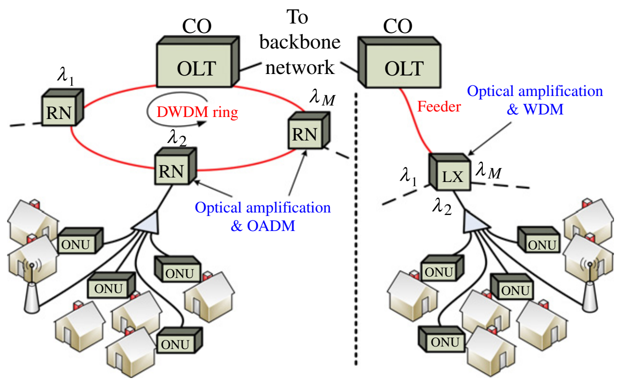


Fig. 1. (Color online) Basic LR-PON architectures: ring-and-spur (left) and tree-and-branch (right).

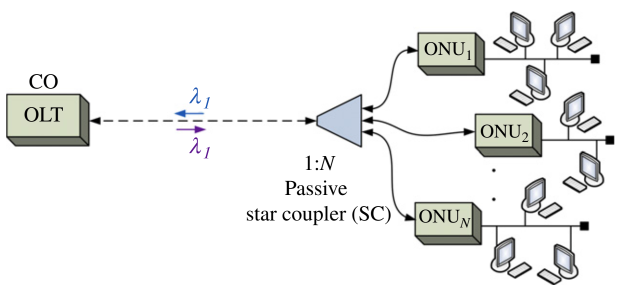


Fig. 2. (Color online) Logical connection between the OLT and the ONUs in an LR-PON.

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 [8,9], 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 as significant in traditional PONs as it is in LR-PONs, where the reach extension may cause the RTT to grow from today's 200 μs (20 km reach) to 1 ms (100 km reach). With this increase, the performance of centralized DBA is ultimately degraded.

In this paper we review and analyze multi-thread polling, which is a recently proposed bandwidth allocation algorithm for LR-PONs, and compare it with traditional interleaved polling. Unexpectedly, we find that the overall packet delay performance of the latter is better than the recently proposed multi-thread polling scheme. Our investigation highlights how multi-thread polling looks compelling for LR-PONs compared to single-thread polling only when the latter is assumed to be scheduled offline. This is because offline scheduling using a single thread leaves a long idle period that the multi-thread scheme utilizes by creating additional threads. However, while traditional interleaved polling is a single-thread scheme, it exploits online scheduling with extremely small idle times. One of the key results of this paper is the demonstration that interleaved polling with online scheduling better reduces the overall packet delay in addition to better utilizing the upstream channel. In the paper, we also point out some inaccuracies in the multi-thread algorithm, and suggest some modifications.

The rest of the paper is organized as follows. In Section II we give a brief introduction to centralized DBA. Next, we review interleaved polling in Section III, explaining how its performance is highly affected by extending the network span. In Section IV, we review multi-thread polling and discuss the different aspects of its algorithm, highlighting its major differences from interleaved polling. Section V presents illustrative numerical results, and Section VI concludes the study.

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. DBA could then be defined as providing statistical multiplexing of resources among the ONUs, which is essential for achieving high upstream bandwidth utilization in TDMA PONs.

Centralized arbitration schemes have been adopted in traditional TDMA PONs, making the OLT arbitrate 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 [9]. In offline scheduling, which is also known as collective scheduling [10], 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

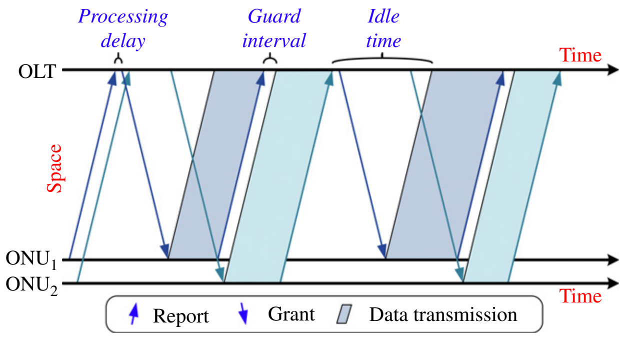


Fig. 3. (Color online) Idle time (walk time) in offline scheduling.

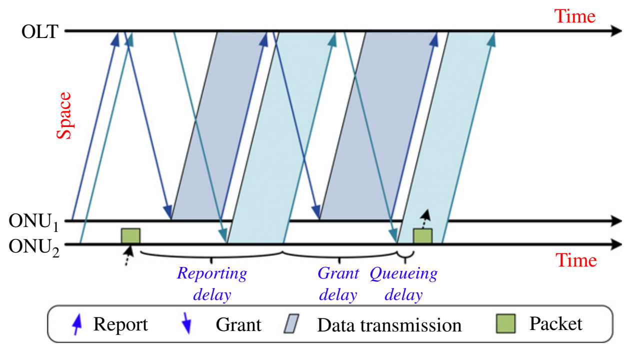


Fig. 4. (Color online) Delays of online interleaved polling.

an idle time of one RTT (i.e., walk time), which is the time from collecting all reports until the first grant reaches an ONU. This is illustrated in Fig. 3 with only two ONUs shown for simplicity, which are shown to be at different distances from the OLT. 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.

III. Interleaved Polling

Interleaved polling with adaptive cycle time (IPACT) has been a pioneering online centralized algorithm for bandwidth allocation in EPON [11]. It employs a pipelined timeslot assignment shown in Fig. 4, 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:

$$W_i = \min(R_i, W_{i,\max}), \tag{1}$$

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 until it receives a grant and it therefore has the shortest cycle duration [11].

A. Polling Cycle

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

$$C_{\max} = \sum_{i=1}^{N} W_{i,\max} + N T_g, \tag{2}$$

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 the 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 new grant to the corresponding ONU [11]. Under heavy traffic, the cycle duration reaches its maximum limit, giving the 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:

$$C_{\text{eff}} = \sum_{i=1}^{N} W_i + N T_g \le C_{\max}, \tag{3}$$

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 that 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.

B. Polling Delays

Figure 4 illustrates the different types of delay a packet may experience from the time of its arrival until the time of its transmission in a centralized polling scheme: reporting, grant, and queueing 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, $C_{\text{eff}}/2$.

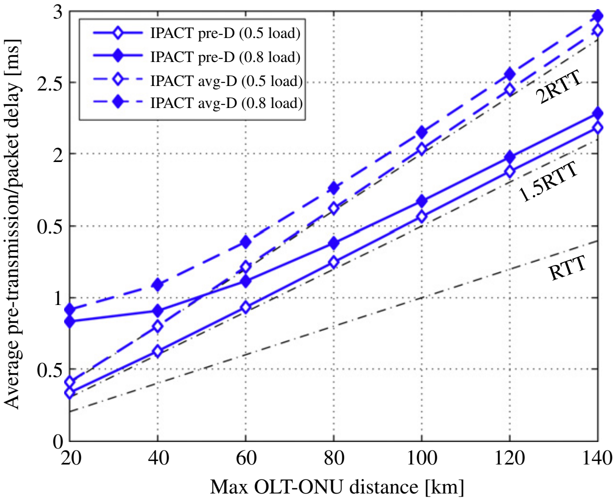


Fig. 5. (Color online) Degradation of IPACT due to network extension.

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 is, the more significant the grant delays are.

3) Queueing Delay: When an ONU finally receives a grant message, it starts transmitting packets one after another within its granted transmission window. Packets therefore experience queueing 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 the upstream transmission rate, and a propagation delay, depending on the distance between the ONU 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 the pre-transmission delays.

It can be seen that grant delays in fact contain both transmission and propagation delays. This makes the 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. 5 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 more than twice the RTT.

IV. Multi-Thread Polling

In [12], 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 shown in Fig. 3 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) running in parallel. This is illustrated in Fig. 6 with two threads, so that the OLT receives the ONU reports of one thread, processes them, and allocates bandwidth to the ONUs, while the other thread is running. This enables the OLT to distribute excess bandwidth among heavily loaded ONUs.

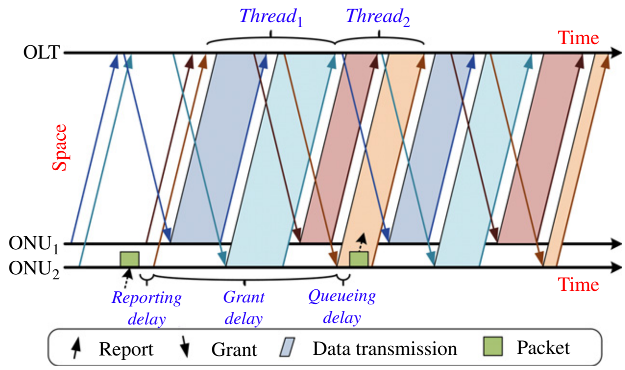


Fig. 6. (Color online) Multi-thread polling.

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 [12], single-thread polling is not necessarily IPACT. Single-thread polling can either be online, such as IPACT (Fig. 4), or it can be offline (Fig. 3), which is sometimes also called interleaved polling with stop [8,9].

It was argued in [12] 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 optimization of 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.

A. Cycle Duration and Thread Tuning

The use of multiple threads actually splits the cycle duration of a single thread. Therefore, the instantaneous durations of M threads constitute the effective cycle duration, defined in Eq. (3), as

$$C_{\text{eff}} = \sum_{i=1}^{M} T_{\text{eff},i} + MT_M \le C_{\max}, \tag{4}$$

where $T_{\text{eff},i}$ is the instantaneous duration of the i^{th} thread and T_M is a guard interval between adjacent threads. After initializing the bandwidth allocation process, the duration of each thread may fluctuate due to the dynamics of the upstream traffic. A thread may sometimes outgrow another thread causing them to act as if they were merged into a single thread. Each ONU will then practically send two consecutive reports within each cycle, and wait for the rest of the cycle, which could have a negative impact on packet delays. Therefore, it was suggested in [12] that the threads should be tuned after each cycle. That is, if the durations of the adjacent threads $n-1$ and n have a ratio greater than a certain threshold (T_{tune}),

$$T_{\text{eff},n-1}/T_{\text{eff},n} > T_{\text{tune}}, \tag{5}$$

the threads are tuned in the following cycle by resizing their durations such that

$$\frac{T_{\text{eff},n-1} - \Delta}{T_{\text{eff},n} + \Delta} < T_{\text{tune}}, \tag{6}$$

where Δ represents some bandwidth moved from the oversized thread to the undersized one. This tuning was shown to improve the performance [12].

B. Delays of Multi-thread Polling

Splitting the cycle into multiple threads reduces both reporting and queueing delays. This comes from the fact that when more threads are used, a packet will wait less time 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 $C_{\text{eff}}/2M$. Each thread will also have less data to transmit compared with single-thread polling. Therefore, the average queueing delay would be $1/M$ of the average queueing 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 the reporting and queueing 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 less data transmissions 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 [12] to be three.

C. Multi-thread Bandwidth Allocation

Unlike IPACT, multi-thread polling is an offline DBA scheme that allocates bandwidth for a thread when all this thread's requests have arrived. This enables distribution of excess bandwidth among heavily loaded ONUs. However, excess distribution is pointless when the aggregated requests are already less than the thread duration, i.e., when the network is normally loaded. Therefore, in multi-thread polling, two bandwidth allocation scenarios are used [12]. When the network is normally loaded, each ONU is granted a transmission window as requested:

$$W_{i,n} = R_{i,n} \quad \text{if } \sum_{i=1}^{N} R_{i,n} \le T_{\max,n}, \tag{7}$$

where $R_{i,n}$ and $W_{i,n}$ are the requested window and allocated window, respectively, for ONU_i in thread n, and $T_{\max,n}$ is the maximum possible duration for thread n, which may vary in each cycle. Note that an additional subscript is used here to differentiate between threads.

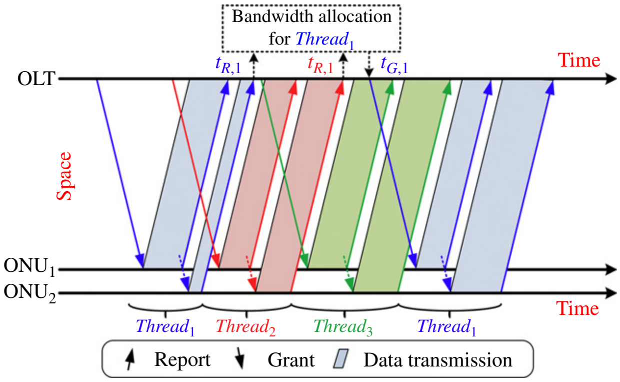
When the network is overloaded, i.e., the aggregated requests exceed the available thread duration, ONUs are partitioned into two groups: normally loaded ONUs and heavily loaded ONUs. Normally loaded ONUs are those with requests less than or equal to their maximum window sizes, that is, $R_{i,n} \le W_{i,n,\max}$. Heavily loaded ONUs are those with requests larger than their maximum grant sizes, that is, $R_{i,n} > W_{i,n,\max}$. The OLT computes the excess remaining from the set J of normally loaded ONUs by

$$W_{\text{excess}} = \sum_{i \in J} (W_{i,n,\max} - R_{i,n}). \tag{8}$$

The excess is then proportionally distributed among the heavily loaded ONUs so that the bandwidth allocation under overloaded network conditions becomes [12]

$$W_{i,n} = \begin{cases} R_{i,n}, & \text{if } R_{i,n} \le W_{i,n,\max}, \\ W_{i,n,\max} + \dfrac{R_{i,n} \times W_{\text{excess}}}{\sum_{i \notin J} R_{i,n}}, & \text{if } R_{i,n} > W_{i,n,\max}. \end{cases} \tag{9}$$

As shown from the above expression, the excess distribution used in [12] is demand-driven excess distribution [9]. This excess distribution is not very accurate, because it may give an ONU more excess than needed while giving others less than needed. As a simple example, consider three ONUs with requests 0, $1.2W_{\max}$, and $1.8W_{\max}$. The excess from the first is $W_{\max}$ (Eq. (8)), which happens to be exactly equal to the overload of the other two. However, according to the above expression, it will be distributed among the two as $1.4W_{\max}$ and $1.6W_{\max}$. Therefore, to avoid giving an ONU unneeded excess, the grant ($W_{i,n}$) can be resized afterward so that it does not exceed the request [9]. Excess saved from this resizing may then be redistributed in an iterative process [9]. Alternatively, and to avoid this inaccuracy from the start, we suggest using the following expression:



$t_{R,n}$: time at which the OLT receives all reports of thread n.

$t_{G,n}$: time at which the OLT grants the ONUs in thread n.

Fig. 7. (Color online) Inter-thread scheduling; the reports of the second thread are considered in the bandwidth allocation of the first thread.

$$W_{i,n} = \begin{cases} R_{i,n}, & \text{if } R_{i,n} \le W_{i,n,\max}, \\ W_{i,n,\max} + \dfrac{(R_{i,n} - W_{i,n,\max})W_{\text{excess}}}{\sum_{i \notin J}(R_{i,n} - W_{i,n,\max})}, & \\ \quad \text{if } R_{i,n} > W_{i,n,\max}, \end{cases} \tag{10}$$

which distributes the excess bandwidth proportionately to the overload of requests (overload driven) rather than the requests themselves (demand driven).

D. Inter-thread Scheduling

When more than two threads are used for multi-thread polling, the OLT can allocate bandwidth for a current thread not only by making use of the information carried within the received reports of this thread, but also from reports of subsequent threads. For example, consider using three threads as illustrated in Fig. 7. Before the OLT allocates bandwidth for the first thread at time $t_{G,1}$, the reports of the second thread arrive at time $t_{R,2}$, reporting the most recent information of the ONUs' packet queues. This is likely to happen when the duration of the third thread becomes long enough to make the allocation of the first thread wait until reports of the second thread arrive. The information in these newly arrived reports can therefore be counted in the bandwidth allocation of the first thread, allowing some packets reported in the second thread to be granted and transmitted in the first thread instead of waiting for the grants of the second thread. However, inter-thread scheduling is only applicable when the current thread duration can still accommodate more packets.

The pseudo-code shown in Fig. 8 shows how current requests can be updated, if their aggregation is less than the current thread duration (after tuning) and the following thread requests had arrived. The code is similar to the one provided in [12], but with a few symbolic differences. It specifies that, if all requests of current thread n and the following thread k (if they had arrived) fit within the current thread duration $T_{\max,n}$, then the current requests are updated to fully include the following requests. On the other hand, if the requests of both threads do not fit within the current thread duration, a

```
Check Δ for current thread n;
if (Σ_{i=1}^{N} R_{i,n} + Δ ≤ T_{max,n})
{
  for (k = n+1;  k MOD N ≠ n;  k++)
    {
    // Requests in thread k and after have not arrived.
    if (Σ_{i=1}^{N} R_{i,k} = 0)
    break;

    // If valid requests in thread k, then calculate and
    // update requests in thread k and n.
    if (Σ_{i=1}^{N} R_{i,n} + Σ_{i=1}^{N} R_{i,k} + Δ ≤ T_{max,n}) {
      R_{i,n} += R_{i,k};
      R_{i,k} = 0;
    }

    else{
      R_excess = Σ_{i=1}^{N} R_{i,n} + Σ_{i=1}^{N} R_{i,k} + Δ − T_{max,n};

      R_{i,n} += R_{i,k} × R_excess / Σ_{i=1}^{N} R_{i,k};

      R_{i,k} −= R_{i,k} × R_excess / Σ_{i=1}^{N} R_{i,k};

    break;

    }
  }
}
```

Fig. 8. Pseudo-code for updating requests in inter-thread scheduling.

percentage of the following requests is moved to the current requests. However, the excess expression used in the code could make updated requests in the current thread either fail to use all the thread duration when they should, or exceed the thread duration. For example, we again consider three ONUs with current requests which happen to be zeros, and the following requests of $0.45T_{\max}$, $0.63T_{\max}$, and $0.72T_{\max}$. If we assume that the thread needs no tuning ($\Delta = 0$), R_{excess} would be $0.8T_{\max}$. Current requests would then be updated to $0.2T_{\max}$, $0.28T_{\max}$, and $0.32T_{\max}$, leaving $0.2T_{\max}$ of the current thread instead of using it. If we now assume that the current requests were $0.1T_{\max}$, $0.15T_{\max}$, and $0.15T_{\max}$ instead of zeros, they would be updated to $0.4T_{\max}$, $0.57T_{\max}$, and $0.63T_{\max}$, which definitely exceed the current thread duration. It is therefore better to compute the excess as

$$R_{\text{excess}} = T_{\max,n} - \Delta - \sum_{i=1}^{N} R_{i,n}, \tag{11}$$

which guarantees that the current thread will be fully utilized without disturbing the thread tuning.

V. Numerical Results

In this section, we first compare the pre-transmission delay components of the discussed centralized DBA 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 1 Gbps, 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 Mbyte. 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 [1,13], in which multiple sub-streams of alternating Pareto-distributed on/off periods are aggregated together. In order to compare between the different schemes, the maximum cycle duration $C_{\max}$ is set to 5 ms 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 5 µs [11]. For multi-thread polling, we use the same parameters used in [12]. The initial intervals of threads are set to be 0.5 ms and 0.3 ms when using two and three threads, respectively, which represent max(RTT)/M [12]. Inter-thread scheduling is enabled, and thread tuning is applied with a threshold T_{tune} of 5 [12].

A. Pre-transmission Delays

Figure 9 shows the average reporting delays of online interleaved polling (IPACT) and offline (collective) polling when using one–three 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 three 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. Under light loads, the threads carry almost 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 three 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 threads cause the cycle to expand, making the reporting delays increase again. This behavior is not significant when using only two threads, because there are only two 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 are shown to be $C_{\max}/2M$.

Figure 10 shows the average grant delays, which are shown to be always greater than the RTT (1 ms) and are the most significant within centralized pre-transmission delays. The grant delays of the collective DBA algorithms are found

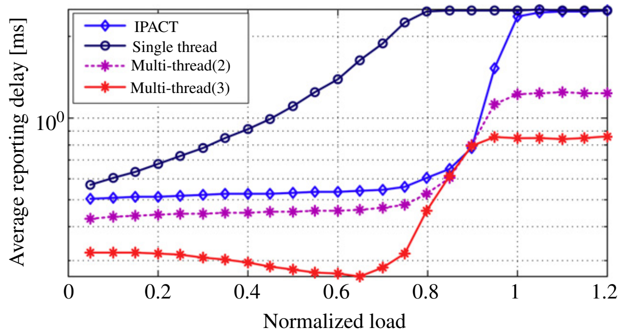


Fig. 9. (Color online) Average reporting delays.

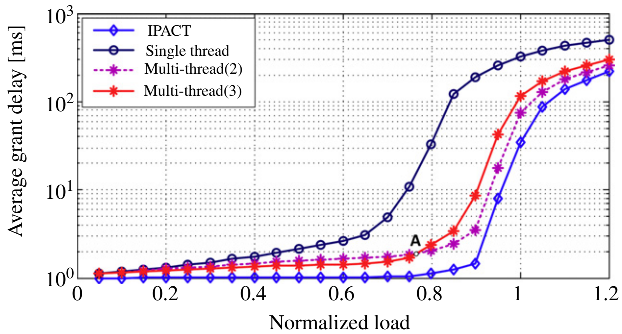


Fig. 10. (Color online) Average grant delays.

to be longer than those of online interleaved polling. For multi-thread polling, using three threads was found to be better when the network load was below 80%, but worse than using two 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 three threads, there are more reports and grants within the cycle, which reduces the grant 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 three 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 three threads. The performance is therefore better when using two threads, as more bandwidth is saved for data transmission, enabling more packets to be granted sooner.

Finally, the average queueing delays are shown in Fig. 11, for which increasing the number of threads proved also to decrease the queueing delays. Note also how the average queueing delays of online polling (IPACT) are greater than those of offline single-thread polling for the same cycle duration. This can be misinterpreted as meaning that online polling is worse, when in fact it is quite the opposite that is true. The reason for the higher queueing delays for online polling is because it has wider transmission windows and hence longer transmission queues. Offline polling has an idle

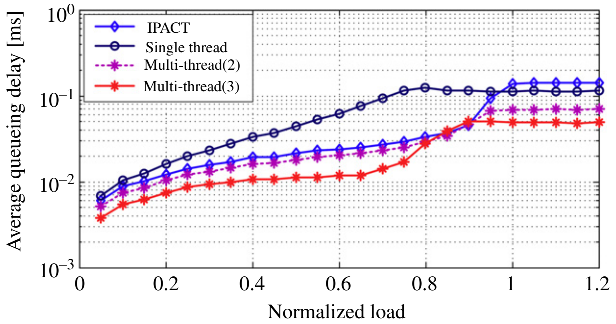


Fig. 11. (Color online) Average queueing delays.

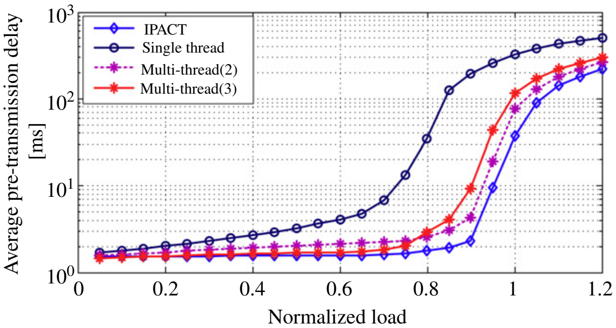


Fig. 12. (Color online) Average pre-transmission delays.

period of one RTT in each cycle, making its transmission windows smaller and transmission queues shorter.

B. Overall Performance

Figure 12 shows an overview of the pre-transmission delays of the discussed centralized schemes. Interleaved polling (IPACT) remains better than using centralized collective DBA for LR-PONs. This is because its grant delays, which constitute the dominant factor, are less than those of the collective schemes. The comparison made in [12] was between multi-thread polling and offline single-thread polling, but not with online IPACT.

Figure 13 compares the throughput of all schemes. The worst throughput was found to be that of offline polling with a single thread, since a 1 ms idle period exists in each 5 ms cycle. That makes the upper bound of throughput for single-thread polling 80%, without including guard intervals and processing delays. The best throughput is that of IPACT, achieving a maximum of 97.1%. The throughput of multi-thread polling is somewhat lower than that of IPACT, which is expected due to doubling the inter-transmission guard intervals and report messages by the number of threads.

C. Network Extension Effect

Figures 14 and 15 show the effect of extending the network span on packet pre-transmission delays and total packet delays, respectively, at both 50% and 80% network loads. The

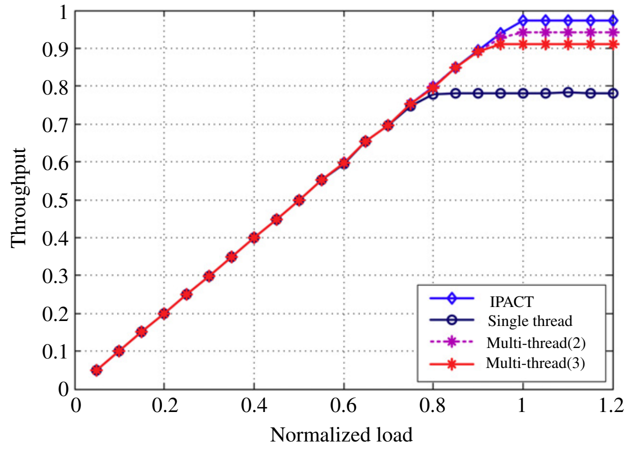


Fig. 13. (Color online) Throughput comparison.

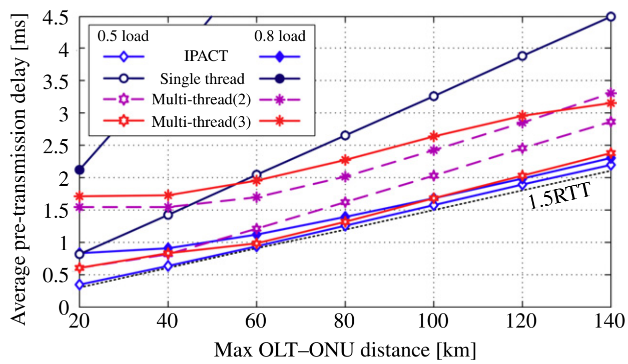


Fig. 14. (Color online) Feeder extension effect on pre-transmission delays.

ONUs are placed randomly but not further than 5 km from the SC. This is because the network extension is usually in the feeder section. The pre-transmission delays of the centralized schemes are shown to be above 1.5RTT, whereas their total delays are always more than twice the RTT. Although IPACT is shown to degrade with distance, it remains better than multi-thread polling. Offline single-thread polling is shown to be the worst, and degrades more rapidly than the other schemes.

VI. Conclusion

In this paper, we addressed the problem of bandwidth allocation in LR-PONs, which suffers from long propagation delays. We studied the performance of two centralized DBA 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

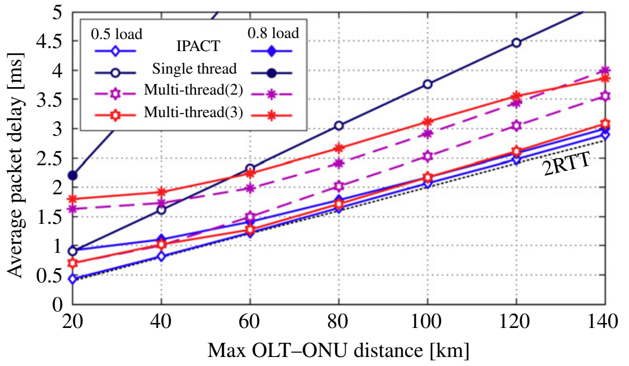


Fig. 15. (Color online) Feeder extension effect on total packet delays.

polling succeeds in decreasing reporting and queueing 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.

Acknowledgment

This work was supported in part by the NPST program of King Saud University under project 09-ELE667-02.

References

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

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

[3] L. Shi, S. Lee, H. Song, and B. Mukherjee, "Energy-efficient long reach passive optical network," *IEEE Syst. J.*, vol. 4, no. 4, pp. 449–457, Dec. 2010.

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

[5] R. Davey, D. Grossman, M. Rasztovits-Wiech, D. Payne, D. Nesset, A. Kelly, A. Rafel, S. Appathurai, and S. Yang, "Long-Reach passive optical networks," *J. Lightwave Technol.*, vol. 27, no. 3, pp. 273–291, Feb. 2009.

[6] D. Shea and J. Mitchell, "Architecture to integrate multiple PONs with long reach DWDM backhaul," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 2, pp. 126–133, Feb. 2009.

[7] H. Song, B. Kim, and B. Mukherjee, "Long-reach optical access networks: A survey," *IEEE Commun. Surv. Tutorials*, vol. 12, no. 1, pp. 112–123, Feb. 2010.

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

[9] M. McGarry, M. Reisslein, and M. Maier, "Ethernet passive optical network archetectures and dynamic bandwidth allocation algorithms," *IEEE Commun. Surv. Tutorials*, vol. 10, no. 3, pp. 151–162, Sept. 2008.

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

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

[12] H. Song, B. Kim, and B. Mukherjee, "Multi-thread polling: A dynamic bandwidth distribution scheme in long-reach PON," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 2, pp. 134–142, Feb. 2009.

[13] G. Kramer, "Synthetic traffic generation," 2007 [Online]. Available: http://glenkramer.com/ucdavis/research.html.

Ahmed H. Helmy received his B.S. degree in electrical engineering from Benha's Higher Institute of Technology (B.H.I.T.), Egypt in 2005, and received his M.S. degree from King Saud University, Riyadh, Saudi Arabia, in 2012. Since 2008, he has been an assistant researcher with Prince Sultan's Advanced Technologies Research Institute (PSATRI) in Riyadh. His current research interests include media access control, optical access networks, and next generation wireless communication systems.

Habib A. Fathallah (S'96, M'01) received his B.S.E.E. degree (with Honors) from the National Engineering School of Tunis, 1994 and his M.A. and Ph.D. degrees in electrical engineering from Laval University, Quebec, Canada, in 1997 and 2001, respectively. He initiated the use of Bragg gratings technology for all-optical/all-fiber coding/decoding in optical CDMA systems. He is the founder of Access Photonic Networks (2001–2006). He is currently with the Electrical Engineering Department, College of Engineering of King Saud University (Riyadh, KSA) and an adjunct professor with the Electrical and Computer Engineering Department of Laval University (Quebec, Canada). His research interests include optical communications systems and technologies, metro and access networks, optical CDMA, PONs and long-reach PONs, FTTH, network monitoring, and hybrid fiber wireless (FiWi) systems.

Hussein T. Mouftah received his B.Sc. and M.Sc. from Alexandria University, Egypt, in 1969 and 1972, respectively, and his Ph.D. from Laval University, Quebec, Canada, in 1975. He joined the School of Information Technology and Engineering (SITE) of the University of Ottawa in 2002 as a Tier 1 Canada Research Chair Professor, where he became a Distinguished University Professor in 2006. He was with the ECE Department at Queen's University (1979–2002), where he was prior to his departure a Full Professor and the Department Associate Head. He has six years of industrial experience, mainly at Bell Northern Research of Ottawa (now Nortel Networks). He served as Editor-in-Chief of the *IEEE Communications Magazine* (1995–97) and IEEE ComSoc Director of Magazines (1998–99), Chair of the Awards Committee (2002–03), Director of Education (2006–07), and Member of the Board of Governors (1997–99 and 2006–07). He has been a Distinguished Speaker of the IEEE Communications Society (2000–2007). He is the author or coauthor of 7 books, 50 book chapters, and more than 1000 technical papers, 12 patents and 140 industrial reports. He is the joint holder of 12 Best Paper and/or Outstanding Paper Awards. He has received numerous prestigious awards, such as the 2007 Royal Society of Canada Thomas W. Eadie Medal, the 2007–2008 University of Ottawa Award for Excellence in Research, the 2008 ORION Leadership Award of Merit, the 2006 IEEE Canada McNaughton Gold Medal, the 2006 EIC Julian Smith Medal, the 2004 IEEE ComSoc Edwin Howard Armstrong Achievement Award, the 2004 George S. Glinski Award for Excellence in Research of the U of O Faculty of Engineering, the 1989 Engineering Medal for Research and Development of the Association of Professional Engineers of Ontario (PEO), and the Ontario Distinguished Researcher Award of the Ontario Innovation Trust. Dr. Mouftah is a Fellow of the IEEE (1990), the Canadian Academy of Engineering (2003), the Engineering Institute of Canada (2005), and the Royal Society of Canada RSC Academy of Science (2008).