Dynamic Bandwidth Allocation with SLA Awareness for QoS in Ethernet Passive Optical Networks

Abhishek Dixit, Bart Lannoo, Goutam Das, Didier Colle, Mario Pickavet and Piet Demeester

Abstract— Quality-of-service (QoS) support in Ethernet passive optical networks (EPON) is a crucial concern. We propose a new dynamic bandwidth allocation (DBA) algorithm, for service differentiation that meets service level agreements (SLAs) of the users. The proposed delay aware (DA) online DBA algorithm provides constant and predictable average packet delay and reduced delay variation for the high and medium priority traffic while keeping the packet loss rate under check. We prove the effectiveness of the proposed algorithm by exhaustive simulations.

Index Terms— Dynamic bandwidth allocation; Passive Optical Networks; Quality of Service; and Service level Agreement.

I. INTRODUCTION

The tremendous growth in network applications necessitates concurrent growth in bandwidth capacities, which lead telecommunication operators to replace bottlenecked xDSL and cable modem technologies with Fiber to the X (or FTTX, where X can mean the home, curb, cabinet, or building) technologies. FTTX networks have been envisioned as a preferred solution because of their potential to provide high data rates and low energy per bit to subscribers, and passive optical networks (PONs) have been widely considered as a promising technology for implementing various FTTX solutions. Nowadays, a variety of PON systems has been proposed, with as most used one the time-division multiple access (TDMA) based PON. So far, Ethernet PON (EPON) and gigabit-capable PON (GPON) are the two variants of TDMA PON, which have

Abhishek Dixit, Bart Lannoo, Didier Colle, Mario Pickavet, and Piet Demeester are with the Department of Information Technology, Ghent University-iMinds, Gaston Crommenlaan 8/201, B-9050, Ghent, Belgium (phone: +32 9 331 49 77; fax: +32 9 331 48 99; e-mail: abhishek.dixit@intec.ugent.be).

Goutam Das was with the Department of Information Technology, Ghent University-iMinds, Gaston Crommenlaan 8/201, B-9050, Ghent, Belgium. He is now with IIT Kharagpur, India (e-mail: gdas@ece.iitkgp.ernet.in).

been used for mass rollout [1]-[2].

EPON is a tree-structured PON technology where in the downstream direction, an optical line terminal (OLT) at the central office (CO) broadcasts data to every optical network unit (ONU) at the user's end, as shown in Fig. 1. In the upstream direction, TDMA techniques are used for scheduling data transmissions from the ONUs to the OLT to avoid any collisions between the users' data. Multi-Point Control Protocol (MPCP) specified in IEEE 802. 3ah is used as the signaling protocol. GATE and REPORT messages are 64 byte Ethernet control messages specified in MPCP. The ONU sends the REPORT messages carrying bandwidth request information based on its queue size, and the OLT sends back the GATE message to the ONU informing allocated bandwidth. Several dynamic bandwidth allocation (DBA) algorithms have been proposed for efficient resource allocation in EPONs [3].
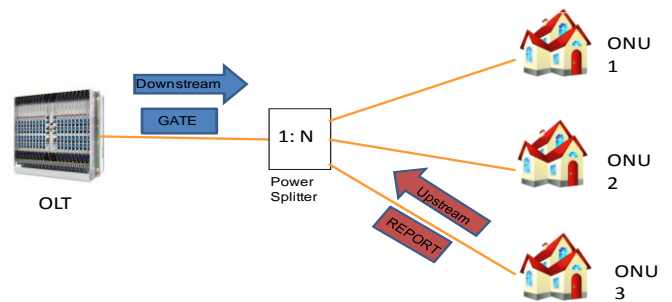

Fig. 1. A typical EPON network

The resource allocation algorithms need to address further many performance dimensions with proliferation of voice and video applications like voice-over-IP (VOIP), IPTV, network gaming, and many others. These emerging applications require a guaranteed bound on many network parameters like: bandwidth, packet delay (latency), packet delay variation (jitter), and packet-loss ratio. This parameter bound delivery of content is referred as Quality of Service (QoS). In this paper, we investigate QoS issues in optical access networks and in particular in EPONs.

An efficient QoS algorithm needs to assure many significant QoS elements and requirements (cf. section II). There are tradeoffs between the various QoS requirements, and the algorithms proposed in literature do not address all the needs of an efficient QoS algorithm. In paper [4], we proposed two delay aware (DA) algorithms, the Delay-Aware Window Sizing (DAWS) algorithm for high-priority traffic and the Delay-Aware Grant Sizing (DAGS) algorithm for medium-priority traffic. These DA algorithms combat the following problems of delivering QoS in an EPON:

- There is a tradeoff between the jitter performance of the high-priority (e.g. voice applications) traffic class and the throughput of the algorithm. Generally, the algorithms poll ONUs online (instantaneously) according to their load and thus packets have load dependent delays, which increase jitter. The online load dependent polling algorithms (also referred in literature as adaptive) achieve high throughput. On the other hand, several algorithms have been proposed to minimize jitter but fail to achieve high throughput (cf. section III). The DAWS algorithm proposed in [4], however, reduces jitter for the high-priority traffic, while keeping the delay within the bound. At the same time, DAWS achieves a high throughput of about 95.5 % [4].

- QoS algorithms have mainly focused to assure bounds for the high-priority traffic class. However, with the emergence of many medium-priority traffic class applications like online gaming and interactive video, the algorithms need to serve even the medium-priority traffic within the desired QoS parameter *(leading to the parameterized QoS control)*. The DAGS algorithm proposed in [4] helps to maintain the average packet delay according to the specified parameter and minimizes the jitter for even the medium-priority traffic (cf. section V). According to our knowledge, no other paper has focused to minimize delay and jitter for the medium-priority traffic class in PONs.

Contributions (in order of their appearance in the paper):
- We highlight important QoS parameters that an efficient QoS algorithm must fulfill (cf. section II).
- We compare various design philosophies that are adopted (e.g. online, centralized and class based) with the other most frequently used ones from the literature (e.g. offline, distributed and ONU based) and highlight important design advantages and pitfalls in designing a QoS algorithm (cf. section III). We propose the modified-DAWS (MDAWS) algorithm to reduce jitter for the high-priority traffic within an acceptable level while maintaining throughput and delay requirements. Further, we extensively describe the MDAWS and the DAGS algorithm and add mathematical formulations to make the description of the DAGS algorithm more concrete (cf. section IV.A and IV.B).
- In this paper, we extend the MDAWS algorithm and the DAGS algorithm to support the SLA of the users. In most algorithms, the users are allocated bandwidth in proportion to their SLA requirement on bandwidth but the users with different delay bound requirements are treated identically. In this paper, for a more robust SLA awareness, we propose the Differential Polling (DP) algorithm in which the users with different delay bound requirements (e.g. home and business requirements) are polled accordingly i.e. the users with a more stringent delay bound requirement are polled more frequently (cf. section IV.C).

- We further identify various factors of channel under-utilization and show how we minimize each one of them. Further, we describe the compliance of the proposed algorithms with MPCP (cf. section IV.D and IV.E respectively).

- We show the extensive performance evaluation of the MDAWS and the DAGS algorithms on delay, jitter, channel utilization, and packet loss rate (cf. section V.A).

- We show by simulation results that the DP algorithm further increases the throughput of EPONs while keeping low jitter and delay for the high and the medium-priority traffic class (cf. section V.B.1).

- Finally, we test the proposed concepts in a next generation PON scenario with 128 users, a reach of 100 km and an upstream line rate of 10 Gbps and 40 Gbps (cf. section V.B.2).

II. QoS REQUIREMENTS

QoS algorithms must meet various requirements. Figure 2 shows important QoS elements. It is elucidative that the requirements between applications may vary. Voice (Skype) and video (YouTube) applications generate traffic with characteristics that differ significantly from traffic generated by data applications (email), and they have delay and loss requirements that are more stringent. Based on the requirements of various bounds, applications are classified in different Classes of Service (CoS) (cf. section IV). We discuss some of the important QoS requirements like delay, jitter, fairness and throughput.

Keeping average/maximum delay bound – All high-priority packets such as network control messages (failure alarms, etc.) and voice traffic must have low delay bounds. The ITU-T Recommendation G.114 specified the one-way propagation delay in an access network (digital local exchange) for voice traffic as 1.5 ms for an analog subscriber line – analog junction and 0.825 ms for a digital subscriber line – digital junction. The medium-priority traffic includes interactive or streaming video applications; the interactive video applications like multimedia conferencing have a very low delay tolerance, while the streaming video applications have a more relaxed delay bound of typically less than 40 ms [5].

Keeping required delay variation (jitter) bound – The high-priority and the medium-priority traffic are also sensitive to jitter. The emerging medium priority applications like 4kTV and 8kTV have stringent jitter requirements to address high users' sensitivity [5]. The algorithm must provide low delay variation to the traffic classes. Generally, load fluctuations in the network increases jitter. Thus, the delay of various traffic classes

should be load independent. The algorithms must impart a centralized delay distribution to the traffic classes, where the delay for a complete range of load is identically distributed [6]. Also, as the delay and the jitter needed is very much dependent on the application of the user, the users must be served according to the QoS parameter needed which is referred as the Parameterized QoS control (PQoS) [7]. Paper [7] emphasizes the need of PQoS to serve multimedia applications.
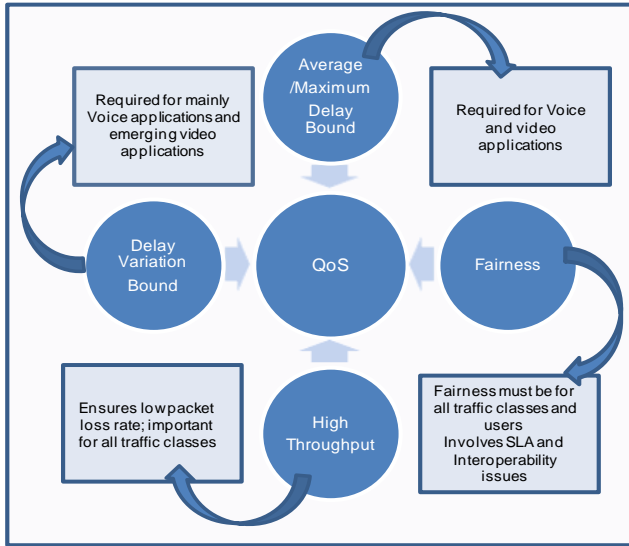

Fig. 2. Quality of Service elements and requirements

Fairness – The fairness among various traffic types (such as high- and low-priority) and the users must be maintained. Even the low-priority traffic should have a minimum guaranteed throughput. A bandwidth hungry high-priority application should not starve a low- priority traffic class. Similarly, every user must be assured a minimum guaranteed bandwidth according to the SLA of the users. Furthermore, the bandwidth allocation for the different traffic classes should be central as otherwise different ONUs may adopt a different bandwidth scheduler making the algorithm less interoperable and unfair.

Keeping high throughput – The algorithm must ensure high throughput. A high throughput improves the performance and simultaneously decreases the packet loss rate. The packet loss rate is not only important for the high-priority traffic but also for the low-priority traffic. Though there is no delivery guarantee for the low-priority packets, the packet loss has a negative impact on the performance of the whole network. It triggers packet retransmission and multiplicatively decreases the transmission control protocol (TCP) congestion window size. Because of the additive increase and multiplicative decrease nature of the TCP protocol, the available bandwidth for all CoS reduces. We can achieve high throughput by maximizing the channel utilization. There are basically four limitations to the channel utilization: idle periods, guard band overhead, unused slot remainders (USRs) and control message overhead. The bandwidth wastage in USRs formation is because variable size Ethernet packets cannot be

fragmented to pack completely the allocated cycle length.

III.   QoS Design Approaches: Pros and Cons

There has been some effort to provide QoS in Ethernet-based-networks. The IEEE 802.1Q [8] standard has identified eight CoS, based on the difference in the requirements of the bounds. The standard has also specified strict (exhaustive) priority scheduling (low-priority traffic is served only after the higher priority traffic classes are completely served) as the default scheduling algorithm to support QoS. The authors in [9] combine the strict priority scheduling algorithm with the Interleaved Polling with Adaptive Cycle Time (IPACT) [10, 11] algorithm as IPACT is an important example of an EPON DBA. However, the queuing delay for low-priority traffic increases when the network load decreases. This problem is referred to as the light-load penalty. In the same paper [9], the authors have tried to eliminate the problem by proposing a two-stage queue and a constant bit rate (CBR) credit scheme. However, both solutions are not free from problems. The CBR credit scheme does not completely alleviate the problem, and the two-stage queue scheme suffers from the problem of increased delay for the high-priority traffic. Furthermore, several other algorithms are proposed, and we discuss the advantages and disadvantages of various design philosophies.

A.   Online vs. Offline QoS Algorithms

In the online approach, upon the arrival and processing of a REPORT from an ONU, the OLT immediately decides on the bandwidth allocation for the corresponding GATE message. Online approaches such as IPACT achieve a high throughput but have a variable cycle time. The variable cycle time leads to variable packet delays or high jitter and is not suitable for the high-priority jitter-sensitive applications [6, 12].

In the offline approach, the OLT waits until it has received the reports from every ONU and then it performs some algorithm to find the best bandwidth allocation scheme for the corresponding grants. The offline algorithms improve the delay performance at low loads; however, at a high load, they lead to excessive delays for increased PON reach [13]. Moreover, the OLT waits to collect all reports before issuing the grants for each ONU, creating an idle time, which is equal to the round trip time (RTT) of the first polled ONU and the grant processing time. Because of the creation of this idle time, offline algorithms generally have low throughput [14] for a limited buffer capacity. Some authors have tried to improve the efficiency of offline scheduling algorithms by minimizing the idle time [14, 15]. The authors in [14] have tried to mitigate the idle time by immediately serving lightly loaded ONUs and predicting the high-priority traffic. This changes the service order of the ONUs in every cycle, thereby impairing the estimation of the high-priority traffic. Furthermore, to provide a constant jitter performance, several offline scheduling algorithms with either separate cycle time for each traffic class (such as Hybrid Grant Protocol (HGP) protocol [12]) or fixed cycle time (such as Hybrid Slot-Size/Rate protocol (HSSR) [16],

Cyclic-Polling-based Bandwidth Allocation with SLAs (CPBA-SLA) [6]) have been proposed. In the HGP protocol, there is a separate cycle for each traffic class and there are idle periods between each cycle, which limits the throughput. The HSSR algorithm increases the delay for low priority traffic and requires large buffers at the ONU, which makes it energy inefficient. Moreover, the HGP and the HSSR algorithm still lead to dispersed delay distribution. The CPBA-SLA algorithm imparts centralized delay distribution but the fixed cycle time limits channel utilization in the context of highly bursty traffic. Thus, an algorithm is needed which increases throughput while minimizing jitter.

B. Centralized vs. Distributed QoS Control

In distributed QoS control algorithms, the bandwidth is granted per ONU by the OLT and is distributed among each CoS in an ONU by using intra-ONU bandwidth scheduling algorithms. For service protection, the distributed QoS algorithms have to trim down the traffic of each queue (ingress shaping) to their minimum guaranteed rate [17]. Furthermore, the distributed control makes the design less interoperable as different ONUs may adopt a different intra-ONU bandwidth scheduler. In addition, in any scheme employing an intra-ONU bandwidth-scheduling algorithm, USRs of as high as 14 % are reported [17].

Thus, a QoS algorithm should have centralized control, i.e. the bandwidth allocation for each CoS of an ONU is done at the OLT. The advantage of centralized QoS control is that it requires an inexpensive ONU design. It increases the cost of an OLT, but that is trivial considering the fact that one OLT line card is shared by 32 customers or even more. Since the centralized QoS algorithms have a network view, it allows a queue of an ONU to have more traffic if the overall network load is low. Besides, the centralized algorithms do not suffer from bandwidth wastage issues of distributed QoS control. We adopt centralized bandwidth distribution mechanisms.

C. ONU vs. Class Based Bandwidth Reporting

The OLT should not be blind with regards to service needs of the traffic at an ONU. Two ONUs with the same SLA and the same buffer statistics should be allocated bandwidth from the OLT according to the service needs of the packets stored. This requires a class based bandwidth reporting in which the queue status of each class is reported to an OLT.

IV. SLA AWARE PROTOCOL FOR QoS

For supporting different CoS, we categorize the traffic into three different classes consistent with the DiffServ (Differentiated Services, RFC 2475) framework:
- *EF (Expedited Forwarding)* – high-priority, delay sensitive traffic with constant bit rate (CBR) such as voice traffic.
- *AF (Assured Forwarding)* – medium-priority, delay sensitive traffic with either interactive video applications like multimedia conferencing or streaming video applications like multimedia streaming (e.g. video

on demand), and broadcast video (e.g. broadcast TV and live events).
- *BE (Best Effort)* – low-priority traffic for non-real time data transfer such as file transfer and e-mail applications.

Note that the classification of various applications as high-priority, medium-priority and low-priority is based on the IP precedence level, cf. RFC 4594. The applications with an IP precedence level more than 4 are assumed as high-priority, 4 and 3 as medium-priority, and less than 3 as low-priority.

We consider delay as the main criterion for scheduling bandwidth for various traffic classes. In paper [4], we proposed the DAWS and the DAGS algorithm for EF and AF traffic, respectively. In this paper, we propose the modified-DAWS (MDAWS) algorithm, which further improves the jitter performance. Furthermore, to meet the SLA of different users, we propose Differential Polling (DP) mechanisms, which we combine with the MDAWS and the DAGS algorithm. In this section, we discuss the MDAWS and the DAGS algorithm, DP mechanisms for SLA awareness, maximizing channel utilization, and compliance of the algorithm with MPCP.

A. Low Delay and Jitter for EF Traffic

The traditional DBA algorithms are GATE-After-REPORT (GAR), which means that the GATE message is issued only after the packet has been reported. This defines a minimum queuing delay of one scheduling cycle, which is at least equal to the RTT of the farthest ONU. As for EF traffic, the number of granted bytes is equal to the number of reported bytes, the maximum queuing delay for a packet is $2T_{max}$ where T_{max} is the maximum cycle time. Thus, the maximum cycle time has to be half of the maximum allowed delay value (D_{EF}). In [12], the authors suggest the GATE-before–REPORT (GBR) method for EF traffic. This approach reduces the maximum delay as now EF packets are granted without first having to be reported. With the GBR method, the maximum queuing delay for EF packets is T_{max} and we can have a larger value of the cycle time for the same maximum delay bound. For example, to meet the delay bound requirement of 1.5 ms for voice traffic, the maximum cycle time can be up to 1.5 ms in the GBR approach as instead of 750 µs in the GAR approach. This reduces guard band overheads, control message overheads and USRs (cf. section IV.C), resulting in an increase in the channel utilization. The GBR method can be used only for predictable EF traffic and not for bursty AF or BE traffic.

When the GBR and/or GAR methods are combined with IPACT, it leads to a deteriorated jitter performance because of the variable cycle time of IPACT. The DAWS algorithm was proposed in [4], to reduce inter-window jitter. Though the DAWS algorithm reduces inter-window jitter, it still does not achieve load independent delay and has a dispersed delay distribution. To maintain the constant delay performance and the centralized delay distribution for EF traffic, we propose the MDAWS algorithm. Let us first understand how the variable cycle time in IPACT leads to an increase in jitter. Jitter can be divided into intra-window and inter-window jitter [12]. The intra-window jitter is

defined as the packet delay variation of two consecutively departed EF packets from the same ONU in the same transmission window. The inter-window jitter is the variation of the first departed EF packet between two consecutive transmission windows. For non-bursty EF traffic, the delay variation of the first departed EF packet between two consecutive transmission windows (inter-window jitter) maps the distribution property of the total EF delay sequence of an ONU [12]. Fluctuations in the inter-window jitter of EF traffic, makes the delay distribution dispersed. As jitter is dominated by inter-window jitter, we further focus on this part. The inter-window jitter between the ith window and the $(i+1)^{\text{th}}$ window of the jth ONU, $J_i[j]$ is given by

$$J_i[j] = D_i[j] - D_{i+1}[j] \tag{1}$$

where $D_i[j]$ is the delay of the first departed packet within the ith window [6] of the jth ONU.

In the MDAWS algorithm, we adopt the delay-aware GBR approach. In the GBR approach, all the expected EF packets are granted at the time of issuing a GATE message. In the MDAWS algorithm, the OLT delays the transmission of EF packets to the subsequent cycle, if the maximum delay of the packets does not increase beyond D_{EF}. If the packets are ungranted in the present cycle and are postponed to the next cycle, the delay of EF packets increases by the length of the next cycle of an ONU. Figure 3 depicts the relations described in a suitable timing diagram where we have considered three subsequent transmission cycles for the jth ONU. Let us consider the application of the MDAWS algorithm at the end of the ith cycle. If EF packets are delayed till the next cycle, the packet delay increases by $C_{i+1}[j]$, where $C_{i+1}[j]$ is the length of the $(i+1)^{\text{th}}$ cycle and is given by the difference in the time epoch of issuing a GATE message to the jth ONU in the $(i+2)^{\text{th}}$ and the $(i+1)^{\text{th}}$ cycle. We represent the time epoch of issuing the ith GATE message to the jth ONU as $gt_i[j]$. Thus, $C_{i+1}[j]$ is given as

$$C_{i+1}[j] = gt_{i+2}[j] - gt_{i+1}[j] \tag{2}$$

The OLT calculates the maximum delay suffered by the
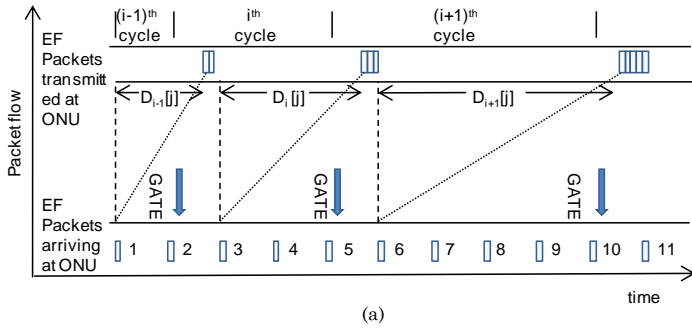
packets if they are postponed to the next cycle as

$$D_{\max} = C_i[j] + C_{i+1}[j] + U[j] \tag{3}$$

where $U[j]$ is the maximum delay of the packets that are ungranted in the previous cycles.

The OLT grants every EF packet in the present cycle if the maximum delay (D_{max}) exceeds more than D_{EF}. If the maximum delay does not increase beyond D_{EF}, then the packets are delayed to the next cycle. It is easy to see that the granted transmission slot can be formulated by

$$w_i[j] = (C_i[j] + U[j]) u[D_{\max} - D_{EF}] \tag{4}$$

where u[] represents the unit-step function and the delay for the ungranted packets is updated by

$$U[j] = U[j] + C_i[j] - w_i[j] \tag{5}$$
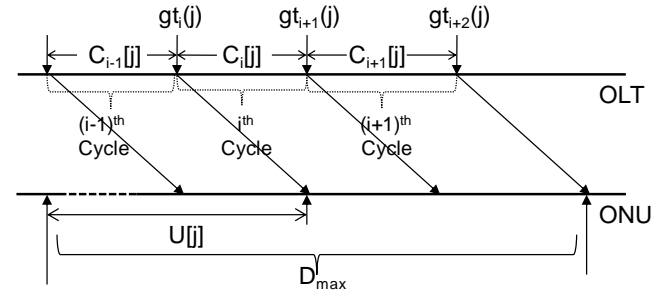
Fig. 3. Timing diagram with three subsequent transmission cycles between the jth ONU and the OLT with an indication of the different symbols used

Figure 4 (a) illustrates when the GBR method is combined with IPACT, resulting in a delay variation of the first departed EF packet. We can see that the delay of the first departed EF packet (represented by $D_{i-1}[j]$, $D_i[j]$, $D_{i+1}[j]$) is nearly equal to the cycle time. As the cycle time is variable, EF packets experience a variable delay. Figure 4 (b) shows that, by implementing the MDAWS algorithm, we achieve a constant inter-window jitter.

(a)

(b)
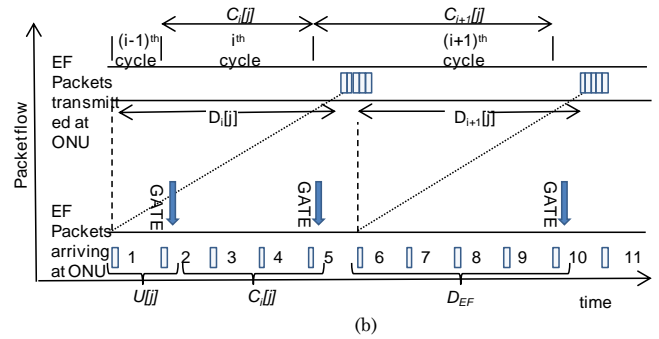
$\square^{n}$ represents nth EF packet
… connects packet arrival and transmission
┊ represents time epoch of packets arrival and transmission of the first departed packet in the cycle

Fig. 4. (a) Illustration of the variation of the first departed EF packet delay of the jth ONU because of the variable cycle time in IPACT; (b) Illustration of the constant first departed EF packet delay because of the Modified Delay-Aware Window Sizing (MDAWS) approach

The implementation of the MDAWS algorithm is not without challenges. First of all, we see that at the time of allocating bandwidth to an ONU, EF traffic is allocated according to the GBR method and later at the time of issuing a GATE message the allocated bandwidth is adjusted according to the expected time of the next GATE message for an ONU. If the OLT decides to postpone the transmission of EF packets, the earlier allocated transmission slot for EF traffic is transferred to BE traffic increasing its performance. Secondly, we see that the transmission slot for an ONU depends on the expected time duration of the next cycle. Since MDAWS (like IPACT) is an online approach, the time of issuing the GATE message of the next cycle for an ONU is not known at the time of issuing the present GATE message. Figure 5 explains it more clearly. We have assumed two ONUs for clarity. Let us assume that at time T, the OLT knows the buffer statistics of both ONUs and their RTT. Thus, at the transmission time of the first GATE G1 to ONU$_1$, the OLT can easily calculate the grant time of the next GATE message for ONU$_1$. However, at the time of issuing the second GATE message for ONU$_1$, the REPORT message from ONU$_2$ has still not arrived and thus the OLT cannot calculate the time epoch of the next GATE message for ONU$_1$. Since, in reality, the number of ONUs is large (16 or 32), the OLT can fairly predict the time epoch of the next GATE message for an ONU at the time of issuing a GATE message. To maintain QoS, every ONU is granted within a maximum transmission window (T_{max}) so that the cycle length remains within a limit and overloaded ONUs do not impact lightly loaded ONUs. The OLT can use the maximum transmission window (T_{max}) for the ONUs of which the REPORT messages have not arrived at the time of decision; however, the maximum transmission window per ONU may impair the performance of the algorithm at very low loads. Thus, the OLT assumes a transmission slot corresponding to the average transmission slot per ONU in the last polling cycle. Careful evaluation helps us to know that for an EPON consisting of N ONUs, the time of issuing the $(i+1)^{th}$ GATE message to the jth ONU ($gt_{i+1}[j]$) will depend on the $[i-1+mod(1,j)]^{th}$ REPORT message of the $[N-mod((N-j+1),N)]^{th}$ ONU; where $mod(x,y)$ is the remainder of (x/y). For example, 3rd GATE of the 4th ONU depends on the 2nd REPORT of the 3rd ONU. When the REPORT messages from an ONU arrive, we determine the grant time of the next (in cyclic order) ONU. Using the latest determined grant time of an ONU k, we can calculate the maximum time epoch at which the $(i+1)^{th}$ GATE message to the jth ONU is transmitted and is formulated by

$$gt_{i+1}[j] = gt_{(i+u[j-k])}[k] + rtt[k] - rtt[j] +$$

$$(j - k + N * (1 - u[j-k])) * \frac{(gt_{(i+u[j-k])}[k] - gt_{(i-1+u[j-k])}[k])}{N_o} \quad (6)$$

where $rtt[p]$ is the round trip time of the pth ONU.

Fig. 5. A simple EPON with an OLT and two ONUs showing REPORT (R1 and R2 for ONU1 and 2) and GATE (G1 and G2 for ONU1 and 2) messages transmission

B. Low Delay and Jitter for AF Traffic

With the emergence of new services, the QoS parameter bound for each service is different. For example, the requirements of interactive video are different from a streaming video application. Thus, the services have to be treated differently and a different share of bandwidth must be allocated to them. The intra-bandwidth scheduler [18] uses weights to distribute bandwidth among traffic classes according to their requirements but these weights are constant and there is no real insight as how to fix them according to the QoS parameter required like the average packet delay. In addition, given the bursty nature of the traffic, the distribution should depend on the instantaneous demands.

To overcome these defects, we proposed the DAGS algorithm in [4] in which we allocate bandwidth such that the parameter bound for a service is maintained and we do not over grant any application. The DAGS algorithm provides constant average delay irrespective of the load, less delay variation and more bandwidth allocation to the services that have stringent delay bound requirements. The algorithm also improves the performance of BE traffic at low load and has parameterized and instantaneous control of the bandwidth distribution.

The challenges that are to be met to provide constant delay performance to AF traffic are different than for EF traffic. AF traffic is bursty and thus the GBR method is not applicable. We have to store the REPORT messages from all ONUs. We translate each REPORT message into newly requested bytes (R_N) which depend on the present REPORT (R_P), last REPORT (R_L) and the granted bytes (G_B) in the last cycle. In addition, since AF traffic transmitted (T_B) at the ONU is different from the traffic granted at the OLT (due to USRs), the OLT has to account for over or under transmitted bytes at the ONU. When the packet size is more than the transmission slot for AF traffic but less than the combined transmission slot for AF and BE traffic, the packet is transmitted leading to over transmitted AF bytes. When the packet size is less than the combined transmission slot for AF and BE traffic, the packet is not transmitted leading to under transmitted AF bytes. These over (O T_B) or under (U T_B) transmitted bytes are also accounted as

$$R_N = R_P - R_L + G_B \pm (U/O)T_B \quad (7)$$

We store reports in a two array format. In one array we store the newly requested bytes and in the other one we assign the corresponding delay value. At the receipt of a REPORT message from an ONU, we initialize the delay field by the previous cycle length. Consecutively, at each GATE message, we update the delay field (D_N) as

$$D_N = D_N + T_L \qquad (8)$$

where T_L is the time difference between issuing the present and last GATE message to an ONU. In each cycle, we grant only those array values for which the corresponding delay field (D_N) exceeds the given threshold value (D_{AF}).

The MAC protocol is suitably adapted to account for the reporting of the over or under transmitted bytes and is further discussed in section IV.E.

C. SLA Awareness - Differential Polling (DP) Algorithm

The QOS algorithm must support the SLA requirements of the ONUs. The ONUs have different requirements of various QoS bounds like delay, bandwidth and jitter. For further discussions, we have assumed two groups of ONUs: at residential and business premises. In traditional QoS algorithms, the weights are used for the bandwidth distribution among different ONUs according to the SLA but different delay bound requirements of ONUs are ignored. The previously proposed algorithms choose the cycle time according to the most stringent delay bound and poll every ONU in a cycle, leading to a similar delay performance of every ONU. We propose the DP algorithm to differentiate ONUs based on delay bounds and the required bandwidth distribution.

Grouping the ONUs according to the EF delay bound is most practical as it poses the most stringent requirement. We consider the delay bound of the EF traffic class for business ONUs as D_{EF} and for residential ONUs as $2D_{EF}$. The GBR method allows that the cycle time is equal to the maximum allowed delay for EF traffic, i.e. D_{EF}. The previously proposed algorithms poll every ONU within a maximum cycle time of D_{EF}. The DP algorithm polls ONUs with a stringent delay bound requirement (such as business ONUs) more frequently than the ONUs with relaxed delay bound (such as residential ONUs). In the DP algorithm, the cycle length is equal to the most stringent delay bound requirement but not every ONU is polled in a cycle. This reduces the number of ONUs polled in a cycle leading to a decrease in both the guard band overhead and the bandwidth wastage due to USR formation (cf. section IV.C). In this way, the DP algorithm improves channel utilization while maintaining the QoS requirements.

To understand it better, let us assume that there are four residential and four business ONUs (cf. Fig. 6). In the first cycle, all business ONU and two residential ONUs are served and in the next cycle again all business ONUs and the remaining two residential ONUs are served. Thus, the business ONUs are polled twice as many as the residential ONUs. Generally, if there are N_a ONUs with delay bound d,

N_b ONUs with delay bound $2d$, N_c ONUs with delay bound $3d$ and so on, then the number of cycles (f_T) required to serve all ONUs is a least common multiple of {1,2,3…}. And the number of ONUs polled in a cycle (n) is

$$n = \left\lceil N_a + \frac{N_b}{2} + \frac{N_c}{3} + ... \right\rceil \qquad (9)$$

The duration of the cycle must be equal to the most stringent delay bound requirement. Furthermore, to maintain the same bandwidth share for all ONUs, the ONUs that are polled more frequently must be given a smaller maximum transmission window per cycle. For an asymmetric bandwidth share, the maximum transmission window of an ONU must be further scaled by the weight factor of the bandwidth share as

$$W_{max} = \frac{[D_{EF} - n.B].R_u.d.W}{N_o.D_{EF}} \qquad (10)$$

where B is the guard band, N_o is the total number of ONUs, d is the maximum delay bound of EF traffic of an ONU, W is the weight of the ONU according to its SLA requirements on the bandwidth share, and R_u is the upstream channel data rate. Note that if the EF delay bound (d) of an ONU is more, it is allocated a larger transmission window.
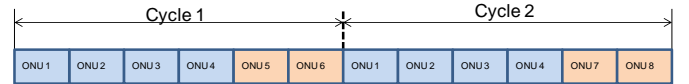

Fig. 6. Business ONUs (denoted in blue) are polled more frequently than home ONUs (denoted by orange)

D. Maximizing Channel Utilization

There are basically four limitations to channel utilization: idle periods, control message overhead, unused slot remainder, and guard band overhead.

- *Idle periods:* The MDAWS and the DAGS are online algorithms and thus there are minimal idle period formations.

- *Control message overhead:* The control message overhead is not a very significant factor of bandwidth wastage. A common GATE and REPORT message for every queue of an ONU is used. At very high load, the control message overhead is given by

$$Control\,mesage\,Overhead = \frac{S_{CM}.N_o}{T_{max}.R_u} \times 100\% \qquad (11)$$

Note that there is a channel wastage of only 0.55 % for $N_0 = 16$, $R_u = 1$ Gbps, $T_{max} = 1.5$ ms, and S_{CM} (Control Message Size) = 64 B. Alternatively, paper [17] proposes a separate GATE and REPORT message for each queue of an ONU, but this increases the control message overhead significantly.

- *Unused slot remainder:* USR is a major limiting factor

for the channel under-utilization. USR occurs due to two main reasons:

1) *Preference to the high-priority traffic* - The preference of the high-priority packets over the low priority packets changes the order in which packets are transmitted, i.e. the transmission is no more first come first serve. Thus, the transmitted packets may be different from the actual reported packets. Since the Ethernet packets cannot be fragmented, packet preemption results in an USR.

2) *Maximum Transmission window per ONU* - To maintain the cycle time within a certain bound, every ONU is granted within a maximum transmission window. If the ONU requests for a transmission slot larger than the maximum transmission window, it is granted bandwidth equal to the maximum transmission window. This makes the GATE different from the REPORT and leads to USRs. The percentage of bandwidth wastage is more if the maximum transmission window size is less. It is easy to see that the percentage of USR formation is:

$$Formation of USRs = \frac{USR N_o}{T_{max} . R_u} \times 100\% \qquad (12)$$

To minimize the USR, all the queues are served according to the granted window size for each queue. The USRs formed in all the queues are aggregated and the queues are again served according to their priority (highest priority first) until the size of the first packet in each queue is more than the difference between the sum of granted size of all queues and the transmitted bytes. This approach reduces USRs formation significantly. It is worth to mention here that finding the packet (or packets) in each queue such that they suitably fill the transmission slot is not a feasible option as it leads to packet reordering and thus has a detrimental effect on the TCP performance [19].

• *Guard band overhead:* To minimize the guard band overhead, the GBR method for EF traffic is adopted. The GBR method allows having a larger cycle time for the same maximum delay for the high-priority traffic class. The guard band overhead is given by

$$Guard band Overhead = \frac{N_o . B}{T_{max}} \times 100\% \qquad (13)$$

E. Class Based Algorithm-Compliance with MPCP

The MPCP has a generic functionality to support the reporting of bandwidth corresponding to each queue of an ONU. A GATE message can issue up to six different grants in one message (using 12 extra bytes available from the Pad field); whereas a REPORT message can report the queue size of up to eight queues. Figure 7 shows the detailed format of the GATE and the REPORT MPCP data unit (MPCPDU). The GATE message contains: DA (destination address), SA (source address), L/T (Length/Type), OC (opcode), TS (timestamp), N.Gr (number of GATE message),

Gr#N ST (Start time of the Nth Grant), Gr#N L (Length of the Nth Grant), ST (synchronization time), P (Pad bytes), FCS (frame check sequence bytes). Similarly, the REPORT message contains DA, SA, L/T, (O/U) TB (the opcode field is used to transmit under/over transmitted AF bytes), TS, N.QS (number of queue sets), RBM (Report Bitmap), Q#N (queue size of the Nth queue), P, FCS. Note the opcode field is introduced in MPCP for future applications, which we use to transmit over or under transmitted AF bytes. The readers are encouraged to refer to IEEE 802.3ah for a detailed description of functionalities of the various fields. The proposed algorithm is completely MPCP compliant.

V. SIMULATION RESULTS

We study the performance of the proposed delay aware (DA) DBA scheme by conducting a simulation of an EPON access network model using the OPNET simulation tool. Table I gives the common simulation parameters used in the scenarios discussed below for 1 G-EPON. EF priority class represents a constant bit rate service such as voice traffic. EF traffic is composed of a number of packet streams, where a stream emulates a T1 connection and is created by the generation of a 70-byte frame at every 125 µs. EF packet streams generate 20 % of the traffic as in [12], [15]. Though EF is a narrowband service, the use of 20% traffic is assumed to study the effect of the high priority traffic on other traffic classes. The remaining load is evenly distributed between AF and BE services. For AF and BE traffic, the synthetic user traffic is self-similar with a Hurst Parameter of 0.8 [12] and with a packet size varying exponentially in the form of Ethernet frames (64 to 1518 bytes). All ONUs are assumed to be symmetrically loaded. The high-priority packets displace the packets with low priority if there is not enough buffer space to store the packet. The maximum cycle time of 1.5 ms is chosen to keep the delay for EF traffic within the bound specified by the ITU-T specification G.114, as also in [20]. In the first scenario, we compare the proposed DA algorithms in 1 G-EPON without differential polling (DP). Next, we see the combined effect of the proposed DA algorithms and DP in 1G-EPON and next generation PON. The 95 % confidence interval of the simulation results gives at most 3 % variation, and is thus not shown in the figures.

TABLE I.
SIMULATION PARAMETERS

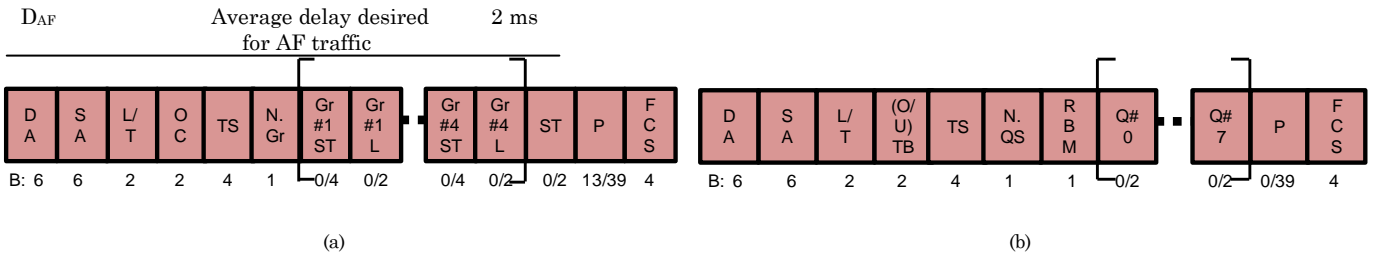| Parameter	Description	Value
N_o	Number of ONUs	16
N	Number of priority classes	3
R_L	Line rate of user-to-ONU	100 Mbit/s
R_U	Upstream bandwidth	1 Gbit/s
Q	Buffer size in ONU	10 Mbyte
L	Maximum distance between OLT and an ONU	20 Km
B	Guard time between adjacent slots	1 µs
T_{max}	Maximum cycle time	1.5 ms
D_{EF}	Maximum delay bound for EF traffic	1.5 ms

D A	S A	L/ T	O C	TS	N. Gr	Gr #1 ST	Gr #1 L		Gr #4 ST	Gr #4 L	ST	P	F C S
B: 6	6	2	2	4	1	0/4	0/2		0/4	0/2	0/2	13/39	4

(a)

D A	S A	L/ T	(O/ U) TB	TS	N. QS	R B M	Q# 0		Q# 7	P	F C S
B: 6	6	2	2	4	1	1	0/2		0/2	0/39	4

(b)

Fig. 7. (a) Format of GATE MPCPDU; (b) Format of REPORT MPCPDU

A. Scenario 1: Comparative Performance without Differential Polling and in 1 G-EPON

In the first scenario, we analyze the performance of the proposed algorithms on network parameters like delay, jitter, channel utilization, and packet loss rate. Our analysis shows that by applying the MDAWS and the DAGS algorithm, we are able to provide a constant delay performance to both EF and AF traffic.

1) Delay

The simulations are done to provide a maximum delay of 1.5 ms for EF traffic in all the schemes. For AF traffic, we have used D_{AF} = 2 ms. Figure 8 gives the average delay of EF traffic class by applying our algorithm. We compare the simulation results with the IPACT (CBR credit) and the DAWS algorithm, as proposed in [4]. The average delay of IPACT and DAWS fluctuates with the load but in the MDAWS algorithm, the delay is constant at around 0.75 ms. This constant delay is attained by the centralized delay distribution in the MDAWS algorithm, compared to the dispersed delay distribution in DAWS and IPACT. Figure 9 gives the average delay of AF and BE traffic classes in various algorithms. We see that the DAGS algorithm leads to a constant delay of 2 ms (DAF) for AF traffic for all loads. Furthermore, as the DA algorithms delay the transmission of EF and AF traffic, the transmission slot is transferred to BE traffic increasing its performance. The enhanced channel utilization of the proposed algorithms (cf. section V.A.3) also improves the performance of BE traffic. Note that in the figure, the performance is shown versus the ONU load. The maximum load of an ONU is 100 Mbps and for 16 ONUs, the ONU load of 0.6 means a total network load of 0.1×16×0.6 Gbps, i.e. 0.96 Gbps. Thus, at this point, the delay performance of the algorithms begins to saturate due to buffer overflow, leading to comparable performance of the algorithms. From these results, it is clear that the MDAWS and the DAGS algorithm provide centralized delay distribution.
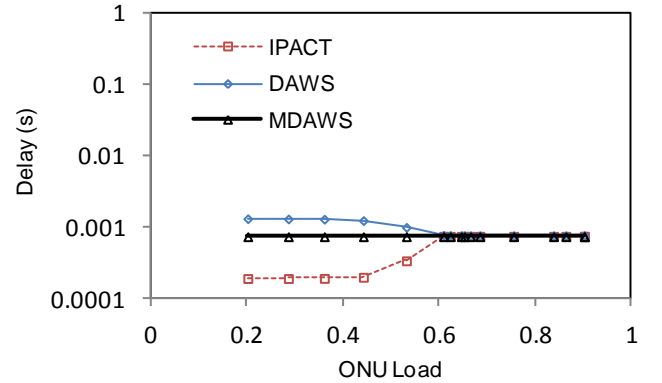
Fig. 8. Average packet delay for EF traffic when conventional DBA, DAWS and MDAWS schemes are applied
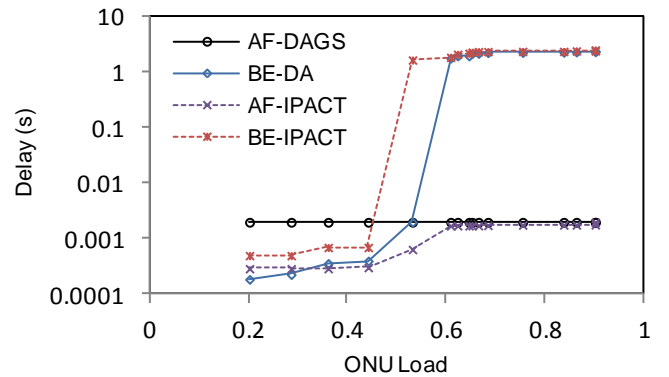
Fig. 9. Average packet delay for AF and BE traffic when conventional DBA and DAGS/MDAWS (DA) schemes are applied

2) Jitter Performance

Figure 10 shows that the inter-window jitter, which is represented by the probability of the delay of the first departed EF packet for the MDAWS algorithm at both half (0.5) and full load (0.95), is centered at the same point whereas in traditional IPACT based approaches, it varies with the load. The variance of the delay in the MDAWS approach is improved at both loads. The variance (σ^2) of the inter-window jitter for half load is 0.00267 ms², and at full load it is 0.003 ms², which shows that there is no variation in inter-window jitter with load. Note that the inter-window jitter at full load is spread more than at half load. The MDAWS has a comparable delay distribution of inter-window jitter as the DAWS algorithm. However, the DAWS algorithm still leads to a dispersed delay distribution. Figure 11 shows the probability of the delay of EF traffic for half and full load scenarios for DAWS, MDAWS and CPBA. CPBA is a fixed cycle algorithm [6], and is used to benchmark the performance of the MDAWS algorithm. We

clearly see that the MDAWS algorithm minimizes the inter-window jitter and has a centralized delay distribution comparable to CPBA, and thus is immune to fluctuations in load. In MDAWS, the variance (σ^2) of EF traffic delay for half load is 0.062 ms^2 and at full load is 0.067 ms^2. Though the MDAWS has a comparable jitter performance to the CPBA algorithm, it achieves a significantly higher throughput (cf. section V.A.3). We show in Fig. 12, that by adopting the DAGS approach, we are able to provide constant delay performance to even AF traffic with considerably reduced jitter. For AF traffic, the variance is 0.02382 ms^2 (at half load) and 0.27222 ms^2 (at full load).

To understand the jitter performance of various algorithms, for simplicity let us consider n cycles of equal length T_c (cf. Fig. 13), where:

$$n.T_c + \Delta = D_{EF} \tag{14}$$

Let us first evaluate the jitter performance of IPACT combined with the GBR method. In the GBR algorithm, the packets that arrive in a cycle are granted in the same cycle. Thus, the maximum delay of the packets is T_c and the minimum delay is 0. Thus, the delay variation (DV) is a function of T_c and 0, and is represented by

$$DV(IPACT) = f(T_c,0) \tag{15}$$

Since T_c is load dependent, IPACT combined with the GBR method leads to a load dependent delay and a dispersed delay distribution. Note that the dispersed delay distribution is due to the variable maximum component of the delay. The variable maximum component of the delay leads to inter-window jitter. The DAWS algorithm [4] tries to minimize the inter-window jitter but still leads to a dispersed delay distribution. When the DAWS algorithm is applied, it keeps on delaying the transmission of the packets to the next cycle until the maximum delay does not increase beyond D_{EF}. At the time epoch of the nth GATE message, it decides to grant the packets and chooses its transmission window [4] as

$$x = (n+1).T_c - D_{EF} \tag{16}$$

Now, the maximum delay of the packets is $n.T_c$, and the minimum delay is $n.T_c - x$. So, the DV is

$$DV = f(n.T_c, nT_c - x) \tag{17}$$

Combining equations (14), (16) and (17), we get

$$DV(DAWS) = f(D_{EF} - \Delta, D_{EF} - T_c) \tag{18}$$

We can see that the inter-window jitter (corresponding to the maximum delay) becomes fairly load independent; however, the minimum value of the delay is still load dependent. Thus though the inter-window jitter is minimized, the average value of the load is still dependent on the load. In the MDAWS algorithm, as soon as the delay

is expected to increase beyond allowable limits, all packets are granted in the same cycle leading to a zero minimum delay. Thus, the delay variation of the MDAWS algorithm can be formulated by

$$DV(MDAWS) = f(D_{EF} - \Delta, 0) \tag{19}$$

At the time of the nth GATE message, the OLT predicts the next cycle length which is denoted as T_P. Note that Δ is smaller than T_P, for the OLT to grant a transmission slot to an ONU. Based on the above explanations, we can formulate the following conditions:

$$\Delta \le T_p \tag{20}$$

$$\Delta \le T_c + \Delta_p \tag{21}$$

where Δ_P is the error in cycle prediction.

$$\Delta \le \frac{D_{EF} - \Delta}{n} + \Delta_p \tag{22}$$

$$\Delta \le \frac{D_{EF} + n.\Delta_p}{n+1} \tag{23}$$

$$D_{EF} - \Delta \ge \frac{n}{n+1}.[D_{EF} - \Delta_p] \tag{24}$$

Further, Δ_P can be formulated by

$$\Delta_p = \frac{P}{N_o}.[T_L(p) - T_L(l)] \tag{25}$$

where P is the number of ONUs for which the REPORTs are not received, and $T_L(p)$ and $T_L(l)$ are the length of the present and last cycle respectively. At a low load, the cycle length is short, which increases P and consecutively Δ_P. Note that, even though Δ_P and $\frac{n}{n+1}$ decrease with load, both factors have different effects on the maximum delay.

3) Channel Utilization

We compare the proposed DA algorithms with the IPACT, HGP and CPBA-SLA protocol. Note that the HGP protocol still leads to a dispersed delay distribution whereas CPBA-SLA gives a centralized delay distribution. Figure 14 shows that we have a much higher throughput of 96 % compared to 86 % in IPACT, 83 % in the HGP protocol and 82 % in the CPBA-SLA protocol. The algorithm achieves comparatively high channel utilization compared to IPACT because of the reduced USRs formation. The USRs formed in all the queues are combined to serve again AF and BE queues. The USRs formation at high load is reduced to about 2.3 %. The HGP protocol suffers from idle period formations and the CPBA-SLA is a fixed frame algorithm. The wastage due to control message overhead is insignificant. The wastage due to control message overhead is 5 Mbps at high load that occupies only about 0.5 % of the channel capacity. Figure 15 shows the packet loss rate of the various algorithms. For a packet loss rate of less than 1 %, the network load should not exceed 0.9.
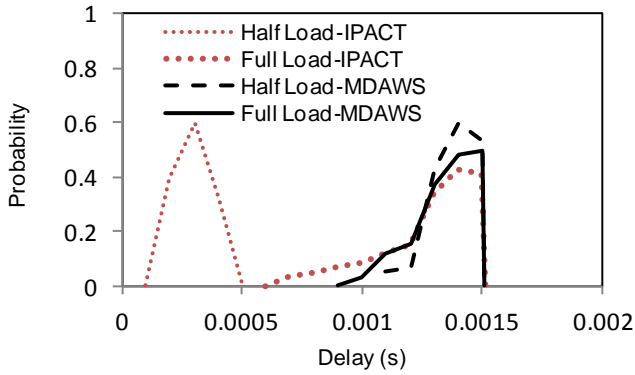
Fig. 10. Probability of the delay of the first departed EF packet for half and full load scenarios for IPACT and MDAWS
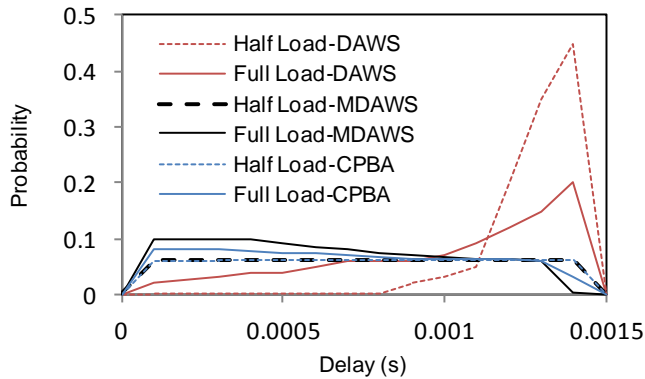

Fig. 11. Probability of the delay of EF traffic for half and full load scenarios for DAWS, MDAWS and CPBA
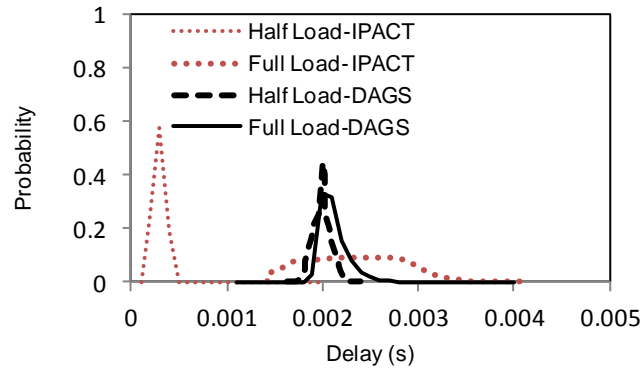

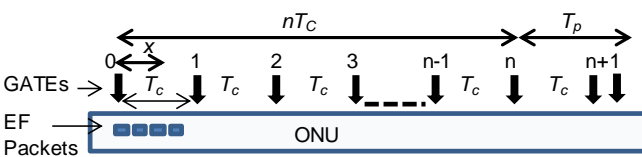Fig. 12. Probability of the delay of AF traffic for half and full load scenarios for IPACT and DAGS

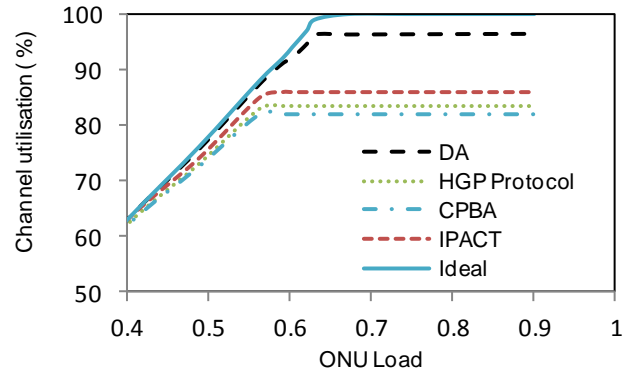Fig. 13. EF packet delay scenario at the ONU

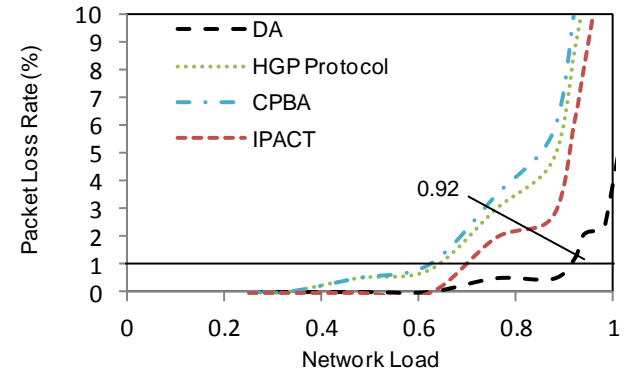Fig. 14. Comparison of the channel utilization of the proposed DA algorithm with the HGP and CPBA-SLA protocol

Fig. 15. Packet Loss Rate (%) vs. Network Load of the proposed DA algorithm and the HGP, CPBA-SLA and IPACT protocol

B. Scenario 2: Comparative Performance with Differential Polling

In this scenario, we evaluate the combined effect of DA and DP in 1 G-EPON and next-generation PON.

1) 1 G-EPON

In this scenario, we demonstrate the effectiveness of the algorithm in 1 G-EPON to provide a differential parameter requirement to different groups of ONUs. We have grouped the ONUs according to the requirement as shown in Table II and the other parameters are adopted as in Table I. According to the DP scheme, we will have four sub-cycles and eight ONUs in each sub-cycle as given by equation (9) and (10).

Figure 16 shows the average packet delay for EF and AF traffic class of four ONU groups. By simulation, we show that we have been able to satisfy different requirements of the ONUs. The ONUs which are polled less frequently show a variation in delay at particularly low load due to the increase in the cycle time prediction error. For example, the ONUs from group D and C are polled once in four cycles and they have to know the REPORTs of the ONUs which are polled in between two cycles, which makes the prediction of the next cycle time more impaired. The higher value of Δ_P (cf. equation 24) reduces the maximum and the average delay. Further, we show the effectiveness of the DP algorithm in comparison to the traditional polling scheme in Fig. 17. We compare the channel utilization in both schemes. The DP algorithm reduces the guard band

overhead and USRs and thus improves the channel utilization by 5 % at high network loads. We compare the throughput performance of various groups in Fig. 18. We can see that the throughput of Group A ONUs is slightly less than the Group B, C and D ONUs. This is because the ONUs from Group A are polled more frequently and thus are given a smaller transmission slot per GATE for fair bandwidth allocation. The smaller transmission slots per GATE leads to higher USR and reduces the throughput performance However, the performance of all ONUs is improved compared to the traditional polling schemes.

TABLE II.
DIFFERENTIAL PARAMETER REQUIREMENTS OF ONUs

Group	ONUs	Parameter required
A	ONU1-ONU4	D_{EF} =0.75 ms D_{AF} =5 ms
B	ONU5-ONU8	D_{EF} =1.5 ms D_{AF} =5 ms
C	ONU9-ONU12	D_{EF} =3 ms D_{AF} =10 ms
D	ONU13-ONU16	D_{EF} =3 ms D_{AF} =20 ms
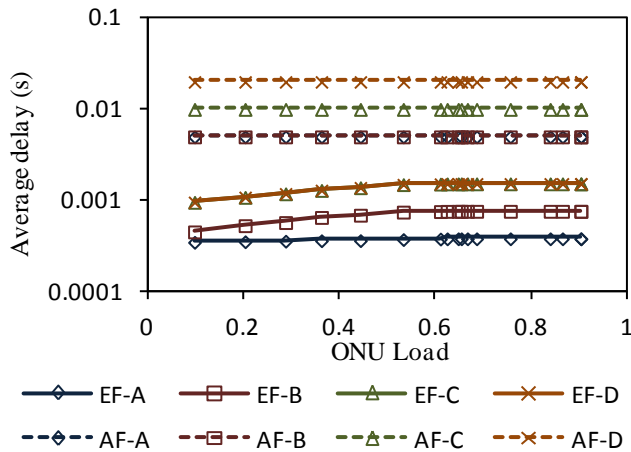
Fig. 16. Average delay of EF and AF traffic of different ONU groups (A, B, C, D) when the MDAWS and DAGS algorithms are applied
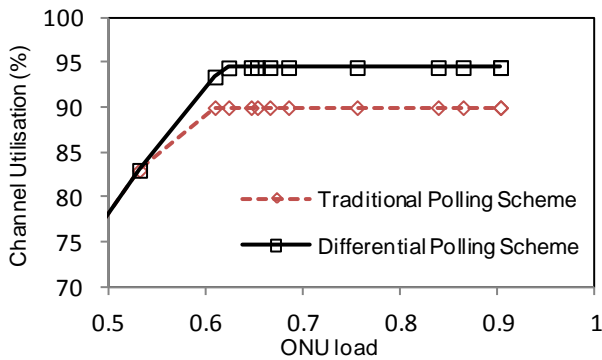
Fig. 17. Channel utilization when traditional and differential polling schemes are combined with DA algorithms
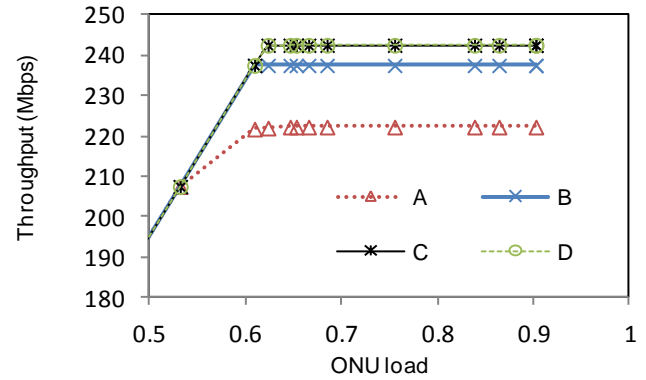

Fig. 18. Throughput of various ONU groups (A, B, C, D) when the differential polling algorithm is combined with DA algorithms

2) Next Generation PON (scalability test)

In this scenario, we evaluate the performance of DP combined with the MDAWS and the DAGS algorithm in the next generation PON scenario. The next -generation PON solutions will upgrade with respect to the PON reach, bandwidth capacity and the split ratio [5]. 10 G-EPON has been standardized in the IEEE 802.3av [21] but still has reach and fan out limitations. Several PON solutions including 10 G and 40 G TDMA PON are actively considered as a potential solution for the next generation optical access. Such a PON solution will require reach extenders in the field and is referred in the literature as Long Reach (LR)-PON [22]. For our simulation study, we first consider a next generation PON solution, with 100 km reach, upstream line rate of 10 Gbps, and a split ratio of 128. As in Table II, we consider four groups of users with the same requirements, whereas now each group consists of 32 users. The MDAWS and the DAWS algorithm keep the delay centered around the desired value and the delay is not significantly affected by the increase in the PON reach and the polling intervals. We obtain the same performance of the users as in the 1 G-EPON scenario for AF traffic. For EF traffic, there is performance improvement for even less frequently polled ONUs as there is less cycle fluctuation in LR-PON compared to 1 G-EPON, cf. Fig. 19. Note that the cycle length in LR-PON varies between 1 and 1.5 ms as compared to 200 μs to 1.5 ms in 1 G-EPON. The throughput performance of the proposed algorithm also improves as there is less USR formation in LR-PON compared to 1 G-EPON. Note that (cf. equation 12), the USR formation depends on the line rate and the number of users. Thus, both the differential and traditional polling algorithms show an improvement in the throughput. The throughput of DP increased to 97.5 % and the traditional polling algorithm to 93.5 %. Similar performance improvements are observed when the upstream line rate is further increased to 40 Gbps.
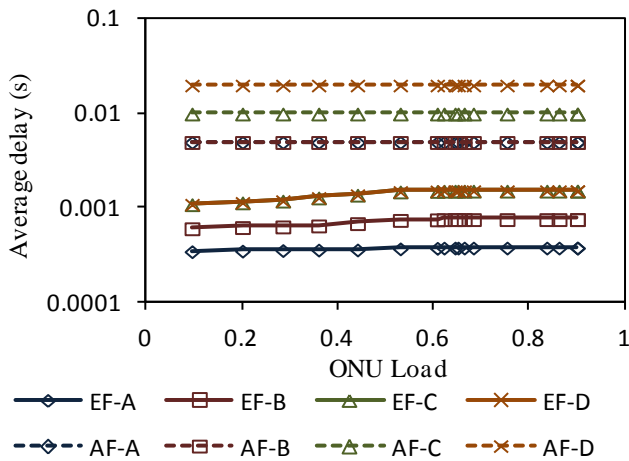
Fig. 19. Average delay of EF and AF traffic of different ONU groups (A, B, C, D) when the MDAWS and DAGS algorithms are applied in an LR-PON scenario.

VI. CONCLUSIONS

We propose the modified delay aware window sizing (MDAWS) and the delay aware grant sizing (DAGS) algorithm for the high-priority and the medium-priority traffic class, respectively, which improve the jitter performance of both the traffic classes. The proposed delay aware (DA) algorithms not only impart a high jitter performance but also achieve a high throughput. The proposed algorithms achieve a centralized delay distribution for both the high and the medium priority traffic class whereby the variance of the delay is almost load independent. The proposed DA algorithms also achieve a high throughput of 96 %. The algorithms implement centralized QoS control and thus make it more interoperable and efficient. The paper also proposes the concept of differential polling (DP) for SLA awareness where the ONUs with different delay bound requirements are polled according to their needs, accomplishing the delay requirements of the prioritized traffic classes while increasing throughput. Further, we also proved the effectiveness of the proposed concepts in long reach (LR)-PON with upstream line rates of 10 G and 40 G with a larger reach (100 km) and a higher number of users (128).

ACKNOWLEDGEMENTS

REFERENCES

[1]   ITU-T Recs. G.984, GPON.
[2]   IEEE Stds. 802.3ah, Ethernet First Mile (EFM).
[3]   M. P. McGarry, M. Maier, and M. Reisslein, "Ethernet PON Architectures and dynamic bandwidth allocation algorithms," *IEEE Commun. Surv. Tutorials*, vol. 10, no. 3, pp. 46-60, 3rd Quarter 2008.
[4]   A. Dixit, G. Das, B. Lannoo, D. Colle, M. Pickavet, and P. Demeester, "Jitter performance for QoS in Ethernet Passive Optical Networks," in *Proc. Europ. Conf. on Opt. Commun.*
(ECOC) 2011, Geneva, Switzerland, Sep. 2011.
[5]   249025-ICT OASE Project, Requirements for European Next-Generation Optical Access Networks, D2.1.
[6]   S. Choi and J. Park, "SLA-Aware Dynamic Bandwidth Allocation for QoS in EPONs," *J. Opt. Commun. Netw.*, vol. 2, no. 9, pp. 773-781, Sep. 2010.
[7]   Available at:
http://www.mocalliance.org/industry/white_papers/PQoS_White_Paper.pdf
[8]   IEEE Standard 802. IQ, 1998, "Virtual Bridged Local Area Networks".
[9]   G. Kramer, B. Mukherjee, S. Dixit, Y. Ye, and R. Hirth, "Supporting differentiated classes of Service in Ethernet Passive Optical Networks," *J. Opt. Netw.*, vol. 1, no. 8 & 9, pp. 280-298, Aug. 2002.
[10] G. Kramer, B. Mukherjee, and G. Pesavento, "IPACT: A Dynamic Protocol for an Ethernet PON (EPON)," *IEEE Commun. Mag.*, vol. 40, no. 2, pp. 74-80, Feb. 2002.
[11] B. Lannoo, L. Verslegers, D. Colle, M. Pickavet, M. Gagnaire, and P. Demeester, "Analytical model for the IPACT dynamic bandwidth allocation algorithm in EPONs," *OSA J. Opt. Netw.*, vol. 6, no. 6, pp. 677–688, May 2007.
[12] A. Shami, X. Bai, C. Assi, and N. Ghani, "Jitter performance in Ethernet passive optical networks," *J. Lightwave Technol.*, vol. 23, no. 4, pp. 1745–1753, Apr. 2005.
[13] M. R. Jason, R. Ferguson, and M. P. McGarry, "Online excess bandwidth distribution for Ethernet passive optical networks," IEEE/OSA J. Opt. Netw., vol. 8, no. 4, pp. 358–369, Apr. 2009.
[14] C. M. Assi, Y. Ye, S. Dixit, and M. A. Ali, "Dynamic Bandwidth Allocation for Quality-of-Service over Ethernet PONs," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 9, pp. 1467–77, Nov. 2003.
[15] I-Shyan Hwang, Jhong-Yue Lee, K. Robert Lai, and Andrew Tanny Liem, "Generic QoS-Aware Interleaved Dynamic Bandwidth Allocation in Scalable EPONs," *IEEE J. Opt. Commun. Netw.*, vol. 4, no. 2, pp. 99 – 107, Feb 2012.
[16] F. An, Y. Hsueh, K. Kim, I. White, and L. Kazovsky, "A New Dynamic Bandwidth Allocation Protocol with Quality of Service in Ethernet-based Passive Optical Networks," in *Proc. of Int. Conf. on Wireless and Opt. Commun. (WOC),* Banff, July 2003.
[17] G. Kramer, On Configuring Logical Links in EPON Available: http://www.ieeecommunities.org/epon.
[18] M. R. Radivojevic and P. S. Matavulj, "Implementation of Intra-ONU Scheduling for Quality of Service Support in Ethernet Passive Optical Networks," *J. Lightwave Technol.*, vol. 27, no. 18, pp. 4055-4062, Sep. 2009.
[19] G. Kramer, B. Mukherjee, and G. Pesavento, "Ethernet PON (ePON): Design and Analysis of an Optical Access Network," *Phot. Net. Commun.,* vol. 3, no 3, pp. 307–19. July 2001.
[20] T. Orphanoudakis, H. C. Leligou, E. Kosmatos and J. D. Angelopoulos, "Performance evaluation of GPON vs EPON for multi-service access," *Int. J. Commun. Syst.* vol. 22, pp. 187–202, 2009.
[21] K. Tanaka, A. Agata, and Y. Horiuchi, "IEEE 802.3av 10G-EPON Standardization and Its Research and Development Status," *J. Lightwave Technol.*, vol. 28, no. 4, Feb 15.
[22] D. P. Shea and J. E. Mitchell, "Long-reach optical access technologies," *IEEE Network*, vol. 21, no. 5, pp. 5-11, Sep. 2007.

Abhishek Dixit received the B.Tech degree (Gold Medalist) in Electronics and Communication Engineering from C.S.A University, Kanpur (India) in 2008 and M.Tech. degree in Optical Electronics and Optical Communication from the Indian Institute of Technology, Delhi (India) in 2010. He is currently working towards the Ph.D.

degree from Ghent University, Belgium. His research interests include lightwave and broadband optical access networks, and he is currently active in the European project ICT-OASE.

Bart Lannoo received an MSc degree in electro-technical engineering and a PhD degree from Ghent University (Belgium) in July 2002 and May 2008, respectively. Since August 2002, he has been working at the IBCN (Internet Based Communication Networks and Services) research group of the Department of Information Technology (INTEC) of Ghent University, where he is currently a postdoctoral researcher. As a member of the IBCN research group, he is also affiliated with the research institute iMinds.

His current research interests are in the field of fixed and wireless access networks, focusing on MAC protocols, Green ICT and techno-economics. He has been involved in various national and European research projects. Since September 2011, Bart Lannoo is coordinating the Green ICT research at IBCN. He is author or co-author of 70 national and international publications, both in journals and in proceedings of conferences.

Goutam Das has obtained his M.Tech. degree from the Indian Institute of Technology, Kharagpur, India in 2001, and his Ph.D. degree from the University of Melbourne, Australia in 2008. He has also worked as a senior research engineer in General Electric R&D center for 3.5 years from the year 2001 to 2004. He has also worked as a postdoctoral fellow at Ghent University, Belgium, from 2009-2011. Currently he is working as an Assistant Professor in the Indian Institute of Technology, Kharagpur.

His research interest is in the area of both optical as well as wireless networking. He has worked in various research topics like optical access networks, architectures and media access control protocol design, radio over fiber technology, optical packet switched networks and wireless media access protocol design for application specific requirements.

Didier Colle received a M. Sc. degree in electrotechnical engineering (option: communications) from the Ghent University in 1997. Since then, he has been working at the same university as researcher in the department of Information Technology (INTEC). He is part of the research group Internet Based Communication Networks and Services (IBCN) headed by prof: Piet Demeester. His research lead to a Ph.D degree in February 2002. He was granted a postdoctoral scholarship from the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) in the period 2003-2004. He became professor at Ghent University in October 2011. Currently, he is co-responsible for the research cluster on network modelling, design and evaluation and is coordinating the research on fixed internet architectures and optical networks.

His research deals with design and planning of communication networks. This work is focusing on optical transport networks, to support the next-generation Internet. Up till now, he has actively been involved in several IST projects (LION, NOBEL, STRONGEST, OPTIMIST, DAVID, STOLAS, LASAGNE, ECODE, EULER, SPARC and OFELIA), in the COST-action 266 and 291 and in the ITEA/IWT TBONES and CELTIC/IWT TIGER projects. His work has been published in more than 250 scientific publications in international conferences and journals.

Mario Pickavet received a M.Sc. and Ph.D. degree in electrical engineering, specialized in telecommunications, from Ghent University in 1996 and 1999, respectively. Since 2000, he is professor at Ghent University where he is teaching courses on discrete mathematics, multimedia networks and network modeling.

He is currently leading the research cluster on Network Design, Modeling and Evaluation, together with Dr. Didier Colle. In this context, he is involved in a large number of European and national research projects, as well as in the Technical Programme Committee of a dozen of international conferences.

He has published over 70 international journal articles (IEEE JSAC, IEEE Comm. Mag., Journal of Lightwave Technology, Eur. Trans. on Telecommunications, Photonic Network Communication, Journal of Heuristics, ...) and about 300 publications in international conference proceedings. He is co-author of the book 'Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS'.

Piet Demeester is professor in the faculty of Engineering at Ghent University. He is head of the research group "Internet Based Communication Networks and Services" (IBCN) that is part of the Department of Information Technology (INTEC) of Ghent University. He is also leading the Future Internet (Networks, Media and Service) Department of the the research institute iMinds. He is Fellow of the IEEE.

After finishing a PhD on Metal Organic Vapor Phase Epitaxy for photonic devices in 1988, he established a research group in this area working on different material systems (AlGaAs, InGaAsP, GaN). This research was successfully transferred to IMEC in 2002 and resulted in 12 PhDs and 300 publications in international journals and conference proceedings. In 1992 he started research on communication networks and established the IBCN research group. The group is focusing on several advanced research topics: Network Modeling, Design & Evaluation; Mobile & Wireless Networking; High Performance Multimedia Processing; Autonomic Computing & Networking; Service Engineering; Content & Search Management and Data Analysis & Machine Learning. The research of IBCN resulted in about 50 PhD's, 1250 publications in international journals and conference proceedings, 30 international awards and 4 spin-off companies.