Dynamic bandwidth allocation algorithms for Ethernet Passive Optical Networks with threshold reporting

D. Nikolova, B. Van Houdt, and C. Blondia

University of Antwerp,
Dept. Math. and Computer Science,
Performance Analysis of Telecommunication Systems Research Group,
Middelheimlaan, 1, B-2020 Antwerp, Belgium
{dessislava.nikolova, benny.vanhoudt, chris.blondia}@ua.ac.be

Abstract

In this paper we present a dynamic bandwidth allocation algorithm for Ethernet Passive Optical Networks (EPON), which makes use of the Multipoint Control Protocol (MPCP) with threshold reporting and with inter- and intra-ONU priority scheduling. Three varieties of this algorithm are compared under both symmetric and asymmetric traffic conditions, by means of a detailed simulation program, regarding average packet delay for several priorities, delay variation for constant bit rate (CBR) traffic and bandwidth utilization. Two types of intra-ONU priority scheduling algorithms are considered, being full and interval priority scheduling, called FPS and IPS, respectively. It is shown that by combining IPS with the threshold reporting mechanism one can achieve a nearly optimal bandwidth utilization (by avoiding nearly all fragmentation losses). IPS, however, causes an increased packet delay and delay variation for CBR traffic in comparison with FPS. In order to eliminate this drawback, we combine the IPS scheduling algorithm with a rate-based scheme for the highest priority (CBR) traffic. The combined IPSA algorithm provides an interesting trade-off between the efficiency, which is still near to the optimal, and the delay characteristics of time critical applications. Finally, we also demonstrate that unfairness arises with FPS under asymmetric traffic conditions, that is, ONUs with more best effort traffic are favored by FPS. Whereas asymmetric traffic conditions only slightly affect the fairness of IPS under low load conditions.

Keywords: Ethernet Passive Optical Network (EPON), dynamic bandwidth allocation, threshold reporting.

1 Introduction

A passive optical network (PON) is a subscriber access network technology that provides high bandwidth capacity over fiber. It is a point to multipoint network with a tree topology. The terminal equipment connected at the trunk of the tree is referred to as an optical line terminal (OLT) and typically resides at the service provider's facility. The OLT is connected to a passive optical splitter using an optical trunk fiber, which fans out at the splitter to multiple optical drop fibers to which Optical Network Units (ONUs) are connected. ONUs typically reside at the subscriber premises, which can be end-user locations or curbs resulting in different fiber-to-the home, business or curb (FTTx) architectures (fiber-to-the-home, fiber-to-the-business or fiber-to-the-curb).

One of the most beneficial features of a PON is that the same fiber infrastructure allows the transporting of different formats - WDM, ATM or Ethernet[1]. The Ethernet protocol is highly deployed in local area networks (LANs) and it is also becoming an emerging technology for metropolian and wide area networks with the standardisation of 10 gigabit Ethernet. Thus, Ethernet is an attractive protocol choice for the access network with its technological simplicity and customer familiarity. A PON, which provides transport of Ethernet frames is called Ethernet PON (EPON)[2]. EPON is a part of the IEEE802 standards family and therefore an implementation may make use of the wide spread Ethernet equipment.

In an EPON, all downstream (from the OLT to the ONU) Ethernet frames transmitted by the OLT, reach all ONUs. ONUs will discard frames that are not addressed to them. In the upstream direction (from the ONU to the OLT) the signal transmitted from the ONU is received only by the OLT. The OLT arbitrates the upstream transmissions from the ONUs by granting Transmission Windows (TWs), which can have variable lengths. An ONU is only allowed to transmit during the TWs allocated to it. In order to inform the OLT about its bandwidth requirements, ONUs use REPORT messages[1] that are also transmitted (along with the data) in the TW. Frames are never fragmented in EPON, therefore, the IEEE working group introduced the concept of *threshold reporting* in order to achieve a higher bandwidth efficiency (see Section 2.2.1).

There are several dynamic bandwidth allocation (DBA) algorithms proposed in the literature which are based partially or fully on the standard multi-point control protocol (MPCP). In [4] an algorithm was proposed where the length of the TWs depends on the ONU's current bandwidth requirements. This algorithm was extensively studied and improved in order to support differentiated services in [5]. Roughly speaking, this algorithm works as follows: All ONUs get a TW in a cyclic order, during each TW an ONU will transmit some data as well as a REPORT message to update the OLT's knowledge about

[1] See Section 2.2.1.

this ONUs bandwidth requirements. The length of a TW of ONU i is completely determined by the contents of the REPORT transmitted in the previous TW of ONU i, that is, the OLT grants a TW with a length equal to the minimum of the requested amount of bandwidth and a predefined maximum (plus the size of a REPORT message). Other types of DBA algorithms are cyclic-based. Such an algorithm allocates bandwidth based not only on the ONU's request but on the total bandwidth requested (by all ONUs) during a cycle. The cycle can have fixed [7] or variable length [6], [10]. The algorithms proposed in [10] are the only ones to make use of the thresholds reporting mechanism. They allocate bandwidth up to the same threshold for all requesting ONUs in the cycle. They also impose a restriction on the minimum length of the cycle which proves to solve some undesirable effects like *light-load penalty* [5] or *idle cycle periods* [6].

In this paper we extend the ideas of the DBA algorithms developed in [10] by introducing a modification to the scheduling algorithm that allows better control on the cycle length, which in turn leads to an improved efficiency. We further clarify the algorithms introduced in [10] and emphasize more on their relationship with earlier works. We also include an elaborate comparison of the performance of the different algorithms under asymmetric traffic conditions, i.e., some ONUs carry a lot of best effort traffic, whereas others do not. We demonstrate that some of the DBA algorithms treat the requesting ONUs in an unfair manner under such traffic conditions.

In Section 2 we present the current state of the standardization efforts regarding EPON. The bandwidth allocation algorithm is introduced in Section 3. The threshold assignment scheme used in this paper is discussed in Section 4, whereas Section 5 describes a way to incorporate a bandwidth allocation scheme for constant bit rate (CBR) traffic in the algorithm. Section 6 presents the simulation results, while in Section 7 conclusions are drawn.

2 EPON: a state-of-the-art

This section provides an overview of the current state of the EPON standardization efforts made by the 802.3ah working group.

2.1 General parameters, transmission window and frame formats

An EPON supports a nominal bit rate of 1000Mb/s, shared amongst the ONUs, which can be at a maximum distance of 20 km. There are two wavelengths – one for the down- and one for the upstream direction. The OLT and the ONUs transmit Ethernet frames at wire speed. In front of each frame there is a preamble of 8 bytes and between two frames there is at least a 12 byte inter-packet gap (IPG) [8]. Between the TW of two ONUs there is

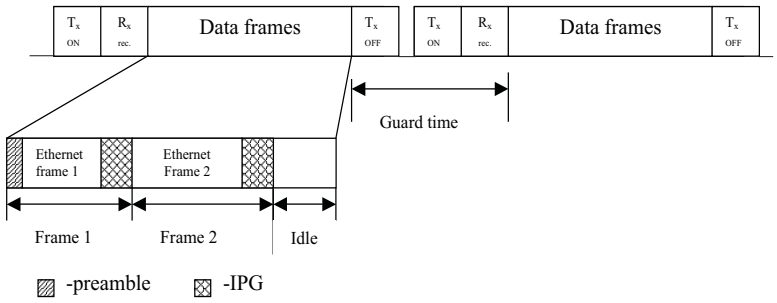


Figure 1: Transmission window and frame format

a certain guard time g needed to account for the laser on (T_{xON}) and off (T_{xOFF}) times, receiver recovery times and other optics related issues (R_{xrec}) (see Figure 1).

2.2 Multi-point control protocol (MPCP)

The Multi-point control protocol (MPCP) defines the messages used to control the data exchange between the OLT and the ONUs as well as the processing of these messages. The OLT assigns the TWs via GATE messages. Each ONU uses a set of queues to store its Ethernet frames and starts transmitting them as soon as its TW starts. An ONU can support up to 8 priority queues as defined in 802.1Q [9]. During a TW the ONU sends data and/or other management messages such as the REPORT message, the contents of which reflects the ONU's current bandwidth requirements. An ONU can also be forced to send a REPORT message within a TW. All MPCP messages are transmitted as Ethernet frames. During a TW, an ONU is free to transmit its Ethernet frames according to an internal scheduling algorithm. Ethernet frames are not fragmented, causing idle periods in the TWs. For example, if an ONU was granted a TW of 1000 bytes and it has 10 frames ready for transmission, each with a length of 101 bytes (including preamble and IPG), it will send only 9 frames in this TW, which leaves 1000-9*101=91 bytes unused. Also, as the order of Ethernet frames must be retained, it is not possible to transmit another frame (from the same queue) that fits in the remainder of the TW. To deal with this drawback the threshold reporting concept was introduced (see Section 2.2.1).

2.2.1 REPORT messages

An ONU transmits its current bandwidth requirements to the OLT by means of a REPORT message. These requirements are indicated by the number of bytes waiting in each queue where the granularity of the reported queue value is 2 bytes. This would imply that there is only one value for each queue in a REPORT message, being the current queue length. However, within MPCP, there can be several Queue Reports (QRs) for one queue in a

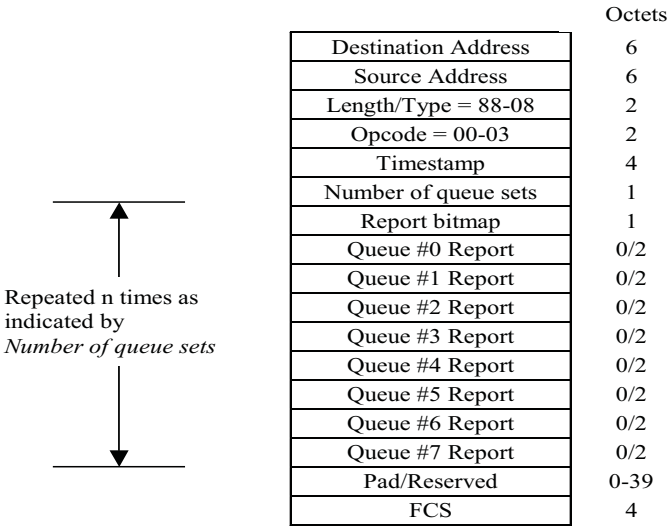


Figure 2: Format of a REPORT message

single REPORT message (allowing an ONU to provide some information on the frame bounds as well). As stated earlier, REPORT messages are transmitted as Ethernet frames as can been seen from the format of the REPORT message (Figure 2).

The Report bitmap identifies the queues for which QRs follow, e.g., 10011000, indicates that 3 QRs, one for the priority $0, 3$ and 4 queue, follow the report bitmap. The time stamp indicates the local time when the message is transmitted by the ONU. In MPCP, ONU i can have several threshold values $\tau^i_{j,l}$ for queue j (with $l = 1, \ldots 13$). These thresholds are used by the ONU to determine the values of the QR fields (see Figure 3). Denote $\beta^i_j(n)$ as the total size of the first n packets waiting in queue j at ONU i. ONU i is said to use the threshold $\tau^i_{j,l}$ if it includes $\beta^i_j(n)$ as a QR in the REPORT message, where $\beta^i_j(n) < \tau^i_{j,l} < \beta^i_j(n+1)$. Infinity can also be a threshold value, meaning that an ONU will report all the bytes waiting in this queue. The length of the REPORT message is 64 bytes and when we add the IPG and the preamble we find that a REPORT message has a length of 84 bytes. From these at most 40 bytes are used to report the bandwidth requirements of an ONU (see Figure 2).

2.2.2 GATE messages

The GATE message contains the starting time and the length of a TW, taking the guard time into account. For example, if the OLT wants to grant a transmission window for 1000 bytes to ONU i it actually grants $1000 + g$ bytes where g is the guard time. GATE messages are transmitted as Ethernet frames of 64 bytes.

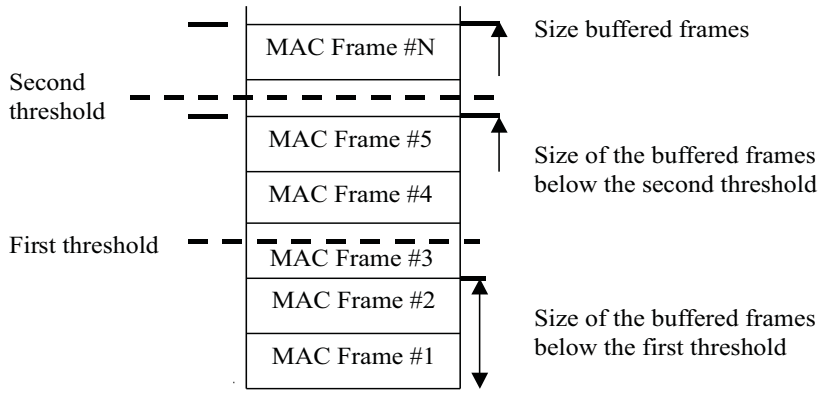


Figure 3: Threshold Reporting: the threshold values are used by the ONU to determine the values of the Queue Report (QR) fields

3 The upstream bandwidth allocation algorithm

This section introduces our bandwidth allocation algorithm. The proposed algorithm is cycle based, where a cycle is defined as the time that elapses between 2 "executions" of the scheduling algorithm. A cycle has a variable length confined within certain lower and upper bounds, which we denote as T_{min} and T_{max} (sec), meaning that the algorithm schedules between B_{min} and B_{max} (bytes) at a time, where B_i is found by multiplying T_i by the line rate. During each cycle each ONU is granted exactly one TW and each registered ONU is forced to send a REPORT message during its TW, thus, even if an ONU reported nothing to the OLT, it is granted a TW by the OLT that is sufficiently large for one REPORT message. Thus, the number of bytes that the OLT needs to schedule is bounded by $\hat{B}_{min} = B_{min} - N(84 + g)$ and $\hat{B}_{max} = B_{max} - N(84 + g)$ bytes (recall, a REPORT requires 84 bytes), where N is the number of registered ONUs and g the guard time. An execution of the scheduling algorithm produces a set of ONU assignments a_i, where a_i indicates the number of bytes that ONU i is allowed to transmit in its TW during the next cycle (see Section 3.4). The length of the TW for ONU i is set to $w_i = a_i + 84 + g$ (bytes).

The REPORT messages used by the OLT to schedule cycle $n+1$ are exactly those that were received during cycle n. Now, due to the distance between the OLT and the ONUs, it should be clear that the REPORT message of some ONUs might not reach the OLT before it executes its algorithm. Indeed, there should be enough time left for the GATE messages that result from executing the algorithm, to reach the most distant ONU before the start of cycle $n + 1$ (which coincides with the end of cycle n). After all,the first TW of cycle $n + 1$ could be assigned to this most distant ONU. This in its turn implies that the ONUs scheduled at the end of a cycle are somewhat disadvantaged. Therefore, we have decided to make the ONU order within a cycle random. Numerical experiments have shown that

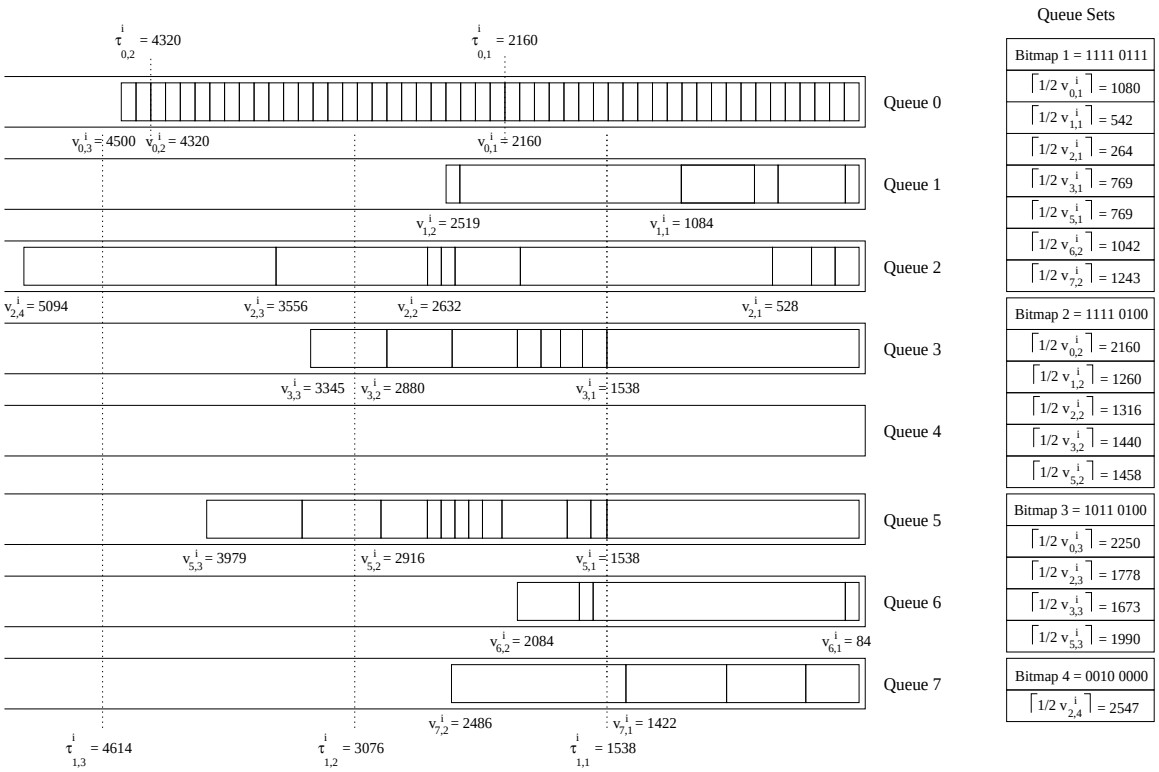


Figure 4: Example: Constructing a REPORT message, left: the contents of the priority queues residing at ONU i, right: the resulting Queue Sets and Queue Reports in the REPORT message

this causes somewhat higher delays, but does guarantee fairness among all ONUs.

The starting time s_i of the TW of the i-th ONU in cycle $n+1$, for $i = 1, \ldots, N$, is found as $s_{i-1} + w_{i-1}/(\text{line rate})$, where s_0 represents the end of cycle n. Before setting the starting time of the GATE messages, the s_i values are recalculated based on the knowledge of the round-trip times of each ONU to represent their local time.

3.1 REPORT message generation at the ONU

Several thresholds, denoted as $\tau_{j,l}^i$ for $l = 1, \ldots 13$, are associated to each queue j of ONU i. We assume that the condition $\tau_{j,l}^i < \tau_{j,l+1}^i$ is satisfied. Recall that a REPORT message uses 39 bytes for the QRs and their associated bitmap fields, as a result there can be at most 13 QRs for the same queue j of ONU i in a REPORT message[2].

In the current algorithm the last threshold for each queue equals infinity (see Section

[2]This can happen if queue j of ONU i is the only non-empty queue. In this case the REPORT message would contain 13 bitmap fields with a single bit set to one, where each bitmap is followed by a single QR; hence, 39 bytes are used to report the state of queue j.

4), to allow reporting of the total number of bytes waiting in a queue. The ONU includes in each REPORT message at least one QR for each queue $j = 0, \ldots, P-1$ that has a length different from 0. Instead of describing the general procedure to generate REPORT messages, let us start with the example presented in Figure 4. In this example we have 7 non-empty queues. The threshold values of queue 0 are multiples of 2160 bytes, while for all other queues the l-th threshold equals $1538l$. The values $v^i_{j,l}$ added below each queue represent the total number of bytes that can be transmitted in a window of length $\tau^i_{j,l}$ without fragmenting packets. The QRs for the queue with the highest priority queue (priority 0) are created first. To report the state of queue 0 we need 3 bitmap fields, whose leftmost bit is set equal to 1, and 3 QRs. Thus, 9 bytes are needed to report the state of queue 0: 1 byte for each of the three bitmap fields and 2 bytes for each QR. The values of the 3 QRs equal 1080, 2160 and 2250. For instance, the total number of bytes that we can transmit in a window of length $\tau^i_{0,1} = 2160$ without fragmenting packets is exactly 2160. Due to the 2 byte granularity of the reported values we find a QR value of $\lceil 2160/2 \rceil = 1080$. For queue 1 we need to add two QRs, whose values equal 542 and 1260 (we make use of the first 2 bitmap fields, introduced for queue 0). Hence, a total of 13 bytes is used to report the values of queue 0 and 1. Similarly, adding the QRs for queue 2 requires 9 bytes, being 4 QRs and 1 additional bitmap field. The same procedure is applied to queues 3, 4 and 5. When we arrive at queue 6 we would like to add 2 QRs. However, a total of 34 bytes was used to report the contents of the first 6 queues, therefore, only 5 bytes are left (as the total size of all queue sets should be at most 39). Two of the 5 remaining bytes are reserved for queue 7, because there has to be at least one QR in the REPORT message for each non-empty queue. As a result, we can only add a single QR, whose value equals 1042 for queue 6. Finally, the remaining 3 bytes allow us to include a single QR with a value equal to 1243 to represent the contents of queue 7.

The procedure used in the previous example can be generalized to the following 3 step algorithm to generate a REPORT messages:

- STEP 1: ONU i generates, for each queue j, a set of 13 values $v^i_{j,l} = \max_n\{\beta^i_j(n) | \beta^i_j(n) < \tau^i_{j,l}\}$. Next, define the set V^i_j as $\{v^i_{j,l} |\ 0 \neq v^i_{j,l} \neq v^i_{j,l-1}\}$ and let $\theta^i_j = 1$ if V^i_j is not empty and let $\theta^i_j = 0$ otherwise. For instance, in our example $V^i_2 = \{528, 2632, 3556, 5094\}$. Ideally, ONU i would like to include a QR for each value $v^i_{j,l}$ in V^i_j for all queues j. However, due to the limited size of a REPORT message, this is often impossible. As a result, a selection has to be made.

- STEP 2: Keeping in mind that ONU i has to include at least one QR for each queue j for which the set V^i_j is not empty, we find that ONU i includes at most $x = \lfloor (39 - 2\sum_{j>0} \theta^i_j)/3 \rfloor$ QRs for queue 0 (2 bytes are used for the QR, 1 for the corresponding bitmap). Hence, ONU i will include $n^i_0 = \min(x, |V^i_0|)$ QRs for queue

0, where $|V|$ is used to denote the number of elements in a set V.

- STEP 3: Denote n_j^i as the number of QRs that are included for queue j (by ONU i). After creating the QRs for the queues $0, \ldots, j-1$ we have at most

$$y = 39 - 2\sum_{k<j} n_k^i - \max_{k<j} n_k^i - 2\sum_{k>j} \theta_k^i$$

bytes left for queue j. Indeed, the first sum represents the size of all the QRs (for queue 0 to $j-1$), the maximum the number of bitmaps used so far and the second sum the amount of bytes reserved for the remaining queues. For instance, in our example we have $y = 39 - 30 - 4 - 2 = 3$ bytes for queue $j = 6$. If $z = \min(|V_j^i|, \lfloor y/2 \rfloor) \leq \max_{k<j} n_k^i$, meaning that there is no additional bitmap required for the QRs of queue j, then ONU i will generate $n_j^i = z$ QRs for queue j. Otherwise, it generates

$$n_j^i = \max_{k<j} n_k^i + \min(|V_j^i| - \max_{k<j} n_k^i, \lfloor (y - 2\max_{k<j} n_k^i)/3 \rfloor)$$

QRs for queue j (the minimum in this term reflects the number of new bitmap fields required).

Finally, the n_j^i QRs for queue j included by ONU i hold the $n_j^i - 1$ smallest values in the set V_j^i and the maximum value in this set. For instance, in our example we included the values 1042 for queue 6 and 1243 for queue 7 in the QRs and not 42 and 711, respectively.

ONUs transmit their REPORT message using the last 84 bytes of the TW, as such we suppose that the reported values represent the state of the queues at the end of the TW, i.e., we do not take into account the necessary time for the ONU to construct such a report. We assume that the ONU takes the IPG and the preamble for each packet into account when reporting (this is also the case in our example).

3.2 Scheduling at the ONU

Two types of scheduling at the ONU are considered in this paper:
1. Full priority scheduling(FPS): With this scheduling algorithm we mean the normal priority scheduling scheme, i.e., the packets are sent according to their priority in a TW. The disadvantage of this scheme is that a substantial amount of time elapses between the transmission of the REPORT message and the start of the corresponding TW, meaning that the contents of the queues is likely to change during this time interval. For instance, if some priority p packets arrived during this interval, these arrivals will destroy the usefulness of the threshold reporting for all the priority $q > p$ traffic (that is, the lower priority traffic). Moreover, higher priority packets may consume some of the bandwidth requested by low

priority traffic, thereby postponing its transmission. This is especially true for light loaded ONUs. This effect has been identified as the light-load penalty (see [5]).

2. Interval priority scheduling(IPS): In this scheduling scheme the ONU remembers the total number of bytes (per queue) that it reported in last REPORT message, i.e., the REPORT transmitted during the last cycle, and it transmits the reported data first. Thus, if higher priority traffic arrived meanwhile, it has to wait until the reported lower priority traffic is transmitted. If the TW is larger than the reported queues' content it continues sending packets according to the FPS scheme. This means that up to certain granularity the ONU respects also the arrival time of the packets. It can be considered as corresponding to the colored grants in an ATM PON (APON) where all the scheduling is done at the OLT site and where the ONU indicates the priority for which a given cell is destined. The IPS scheme coincides with the two stage buffer scheme proposed in [5], where it was mainly introduced to combat the light-load penalty. We however are mostly interested in its interaction with the threshold reporting mechanism. Results presented in Section 6 demonstrate that combining IPS with the thresholds can result in substantial efficiency gains.

3.3 Processing of the REPORT messages at the OLT

The OLT maintains a table with information about the state of the queues at each ONU. This table contains the fields $r^i_{j,l}$, for ONU $i = 1, \ldots, N$, queue $j = 0, \ldots, P-1$ and $l = 1, \ldots, 13$ (recall, we use 13 thresholds, as at most 13 QRs for the same queue j can be supported by a REPORT message): These fields are updated whenever the OLT receives a REPORT message (from ONU i) as follows:

- STEP 1: The OLT starts by clearing the $13P$ fields $r^i_{j,l}$. Next, it scans the REPORT message and whenever it encounters a QR that corresponds to queue j, it enters twice[3] the value v found in this QR in $r^i_{j,x}$, where x is the smallest index such that $v \leq \tau^i_{j,x}$.

- STEP 2a: If the field $r^i_{j,13}$, corresponding to the infinity threshold, is non-empty and if $y < 13$ is the largest index y for which $r^i_{j,y}$ is non-empty, then we set $r^i_{j,l} = \tau^i_{j,l}$ for all $y < l < 13$. This rule is meant to improve the fairness of the algorithm and might cause some minor bandwidth losses due to fragmentation.

- STEP 2b: If the field $r^i_{j,13}$ is empty and if y is the largest index y for which $r^i_{j,y}$ is non-empty, then we set $r^i_{j,l} = r^i_{j,y}$ for all $y < l \leq 13$.

[3]Due to the granularity of the QR values

- STEP 3: Finally, the OLT will increase all $13(P-1)$ entries $r^i_{j,l}$, for $j = 1, \ldots, P-1$ and $l = 1, \ldots, 13$, by $\sum_{k<j} r^i_{k,13}$.

This procedure guarantees that $r^i_{j,l}$ contains the size of all the packets that were reported in the last REPORT message of ONU i for the queues $k < j$ as well as the first n packets of queue j, where n is the maximum number for which $\beta^i_j(n) \leq \tau^i_{j,l}$. Recall $\beta^i_j(n)$ was defined as the size of the first n packets in queue j of ONU i. Thus, $r^i_{P-1,13}$ represents the total number of bytes reported by ONU i. For instance, in the example of Figure 4 we have $r^i_{2,3} = 2 * (2250 + 1260 + 1778) = 10576$ bytes.

3.4 Scheduling at the OLT

The scheduling at the OLT is based on the table with the $r^i_{j,l}$ fields (see Section 3.3). The OLT constructs the GATE messages for cycle $n+1$ as follows. First, if the REPORT message of ONU i (transmitted in cycle n) did not reach the OLT before the execution time of the algorithm (because its TW was located near the end of cycle n, see Section 3), then $r^i_{j,l} = 0$ for all j and l. Next, the OLT computes the following sums:

$$R_{j,l} = \sum_i r^i_{j,l}, \tag{1}$$

for all j and l. Notice, $R_{j,l}$ represents the amount of bandwidth requested by all ONUs for priority $p < j$ traffic and priority j traffic up to threshold l. Thus, $R_{j,l} \leq R_{k,m}$ if $j < k$ or if $j = k$ and $l \leq m$. Let $R_{tot} = R_{P-1,13}$, then the amount of bandwidth a_i allocated to ONU i depends in the following manner on R_{tot}:

1. $R_{tot} < \hat{B}_{min}$: In this case the assignment lengths a_i of the ONUs are the amount they have requested (i.e., $r^i_{P-1,13}$) plus a fair share of the remaining amount of bandwidth up to $\hat{B}_{min}$ (i.e., $(\hat{B}_{min} - R_{tot})/N$).

2. $\hat{B}_{min} \leq R_{tot} \leq \hat{B}_{max}$: In this case the ONUs are assigned exactly the amount of bytes they have requested, $a_i = r^i_{P-1,13}$.

3. $R_{tot} > \hat{B}_{max}$: The scheduler now has to find the largest index l and queue j for which $R_{j,l} < \hat{B}_{max}$ starting from the queue with the highest priority (notice, $l < 13$ as $R_{tot} > \hat{B}_{max}$).

   (i) If $l + 1 \neq 13$, we start by setting $A = \sum_i a_i$, where $a_i = r^i_{j,l}$. Next, the OLT considers each of the values $r^i_{j,l+1}$ for all ONUs i in a random order and sets $a_i = r^i_{j,l+1}$ if $A' = A + (r^i_{j,l+1} - r^i_{j,l}) \leq \hat{B}_{max}$ in which case A is replace by A'. The fairness between the ONUs is guaranteed by the random order. An appropriate choice of the threshold values $\tau^i_{j,l}$ guarantees that $A > \hat{B}_{min}$ (see Section 4).

(ii) If, on the other hand, $l + 1 = 13$, we start by setting $a_i = r^i_{j,l}$ and $A = \sum_i a_i$. Next, we increment a_i in an iterative manner as long as $A \leq \hat{B}_{max}$ as follows. Let $x_i = r^i_{j,13} - a_i$, then increment a_i by $\min(x_i, FS)$, where the fair share FS equals $(\hat{B}_{max} - A)/N_r$ and N_r equals the number of ONUs for which $x_i > 0$. This simple iteration distributes the remaining bandwidth $\hat{B}_{max} - R_{j,l}$ in a fair manner between the ONUs that requested more than $r^i_{j,l}$ bytes in such a way that none get more than they requested, i.e., $a_i \leq r^i_{j,13}$.

The distinction between case 3(i) and 3(ii) is made because $\tau^i_{j,12}$ could be much smaller than the buffer size of queue j (setting $\tau^i_{j,13}$ equal to the buffer size is obviously equivalent to having an infinite threshold value).

4 Threshold assignment

There is no standard way to convey the thresholds to the ONUs. There are several possibilities: at the registration of an ONU, this would result in static thresholds, or in some of the Operation and Management (OAM) messages, providing the possibility for dynamic thresholds, e.g., threshold values that dependent on the number of currently registered ONUs in the network. The number of the thresholds per queue may influence the way in which the thresholds are conveyed to the ONUs, e.g., if an ONU has 8 priority queues each having 13 thresholds that require regular updates, several Mbit/s are needed to keep the thresholds up to date. Also, with the scheduling algorithm proposed in Section 3, dynamically changing the threshold values could result in a temporary unfairness caused by the simultaneous use of both old and new threshold values.

Our proposal is that for each queue j of ONU i only one threshold is assigned (being $\tau^i_{j,1}$) and all others are derived from it. As such, the assignment can be static or dynamic without requiring a lot of bandwidth. A simple and logical way is to use linear dependence

$$\tau^i_{j,l} = l\ \tau^i_{j,1}, \tag{2}$$

for $l < 13$. As explained in Section 3, the maximum number of QR values in a single REPORT message is 13 so there is no point in having more than 13 thresholds for a queue. In order to provide the OLT with a complete view of the bandwidth requirements of an ONU, $\tau^i_{j,13}$ equals infinity (which is equivalent to setting $\tau^i_{j,13}$ equal to the ONU buffer size). Finally, setting $\hat{B}_{max} - \hat{B}_{min} > 2\tau^i_{j,1}$ guarantees that case 3(i) of the scheduling algorithm presented in Section 3.4 generates a cycle length between B_{min} and B_{max}.

5 CBR applications

To achieve a better performance for time critical applications that have a constant bit rate (CBR), e.g., voice, it would be preferable to assign the CBR bandwidth to the ONUs according to the rate of these applications, avoiding the need for the ONUs to report the status of the highest priority queue (cfr. [11]). Such a mechanism requires the possibility in EPON to establish "a connection", however, Ethernet is a connectionless protocol. Nevertheless, given the importance of such a mechanism we believe that a way to report or estimate the rate of a CBR application should and will be standardized. Efforts in this direction are currently being discussed by the Metro Ethernet Forum.

We will explore what the performance of a CBR application would be if the rate r_{CBR}, expressed in packets/s, and packet size s_{CBR} of each CBR application is known by the OLT and we propose a way to incorporate this information into the scheduling algorithm. A rate allocation scheme, named the CBR credit scheme, that also assumes that the OLT is aware of the rates and packet sizes of the existing CBR applications, was proposed and studied in [5].

5.1 Rate-Based Scheduling

The idea behind rate-based scheduling is to predict the number of packets n_{CBR} that arrive at the ONU (from each CBR application) during the time interval spanned by two consecutive TWs W_1 and W_2. For doing so, we can make use of the formula provided in [5, Equation 5]:

$$n_{CBR} = \left\lceil \frac{t_s + h/LR - t_r}{p_{CBR} - s_{CBR}/LR} \right\rceil, \tag{3}$$

where p_{CBR} is the period of the CBR application[4], h is the amount of bytes in W_2 allocated to the non CBR traffic, LR is the line rate, t_s the starting time of W_2 and t_r the time stamp in the REPORT message transmitted during W_1 (see Figure 5).

Knowing n_{CBR} for each CBR application of ONU i, the OLT computes the number of bytes a_i allocated to ONU i as h plus the total amount of bytes b^i_{CBR} required for the CBR traffic of ONU i. The value b^i_{CBR} clearly depends on the TW's starting time t_s, which is not known before executing the scheduling algorithm (except for the first ONU in the cycle). Moreover, $\sum_i a_i$ should be smaller than or equal to $\hat{B}_{max}$. To account for this, we first reduce the maximum number of bytes $\hat{B}_{max}$ to be scheduled to $\bar{B}_{max} = \hat{B}_{max} - B_{CBR}$, where $B_{CBR} = \sum_i b^{i,max}_{CBR}$ and $b^{i,max}_{CBR}$ equals the amount of bytes required for ONU i assuming that the distance between the two consecutive TWs is maximal, i.e., $2T_{max}$.

[4] p_{CBR} can be obtained as the reciprocal of the rate expressed in packets per second.

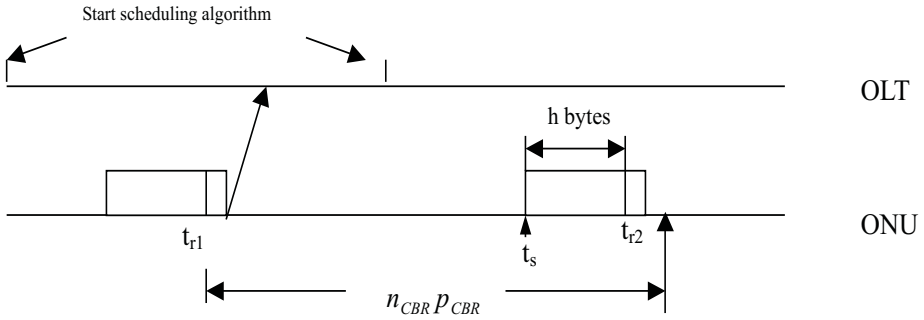


Figure 5: Calculation of the number of packets n_{CBR} that arrive during an interval spanned by two TWs

In conclusion, we first evoke the algorithm with $\bar{B}_{max}$ instead of $\hat{B}_{max}$ to find, for each ONU, the amount of bandwidth allocated to the non CBR traffic, afterwards we add b^i_{CBR} to find a_i starting with the first ONU in the cycle. Indeed, the end of the TW of the $i-1$-th ONU in a cycle determines the starting time t_s of the i-th ONU.

It is important to keep in mind that the CBR traffic is always transmitted first in a TW when the rate-based scheme is used, independent of the scheduling algorithm implemented at the ONU. Thus, if an ONU uses IPS scheduling, it will first transmit the CBR traffic, followed by (a part of) the reported data and finally (if there is still some room left) by some unreported data.

6 Simulation Results

A variety of simulation results for both symmetric and asymmetric traffic scenarios is presented within this section. Part of the results in Section 6.1, for a slightly different scheduling algorithm, were presented in [10]. The differences between the results in Section 6.1 and [10] mainly exist in the high load regions.

Our aim is to compare the performance of three scheduling algorithms, which are all based on the MAC algorithm described in Sections 3-5. The algorithms differ in the scheduling algorithm used at the ONU (FPS/IPS) and the OLT policy regarding CBR traffic. To do so, we simulate an EPON system with $N = 32$ ONUs each at a randomly chosen distance between 0.5 and 20 km. Also, each ONU supports 3 priority queues, the size of which is 1 Mbyte. The line rate LR between the OLT and the ONUs is 1000Mb/s and the rate at which Ethernet packets (IPG included) are generated at the ONUs is 100Mb/s. The guard time g between two consecutive TWs is 1μs. The time required to execute the algorithm (as well as to generate the GATE messages from the results of the execution) is assumed to be 0.1 msec.

The cycle length, which was defined as the time that elapses between two executions of the scheduling algorithm, varies between $T_{min} = 0.5$ ms and $T_{max} = 1.5$ ms, meaning that $B_{min} = 62500$ bytes, $\hat{B}_{min} = 55812$, $B_{max} = 187500$ bytes, $\hat{B}_{max} = 180812$ and $\bar{B}_{max} = 118380$ bytes, unless otherwise stated (see Sections 3 and 5 for definitions). The thresholds are chosen as follows $\tau^i_{0,1} = 2160$ and $\tau^i_{1,1} = \tau^i_{2,1} = 1538$ bytes for all i, unless otherwise stated. The other thresholds $\tau^i_{j,l}$ are obtained from these as explained in Section 4. Let us describe the three scheduling algorithms:

- R-FPSA (Rate-Based FPSA): In this setup, we use the rate-based scheduling algorithm for the CBR traffic, i.e., highest priority traffic, and the FPS scheduling algorithm at the ONU.
- R-IPSA (Rate-Based IPSA): In this case, we combine the rate-based scheduling algorithm for CBR traffic with IPS scheduling at the ONU.
- IPSA: We do not use the rate-based scheduling algorithm for CBR traffic and IPS scheduling is used at the ONU. The maximum cycle length $T_{max} = 1$ ms, meaning that $B_{max} = 125000$ bytes. We have chosen a smaller maximum cycle length T_{max} in order to restrict the maximum delay for the priority 0 traffic, i.e., CBR traffic. $\tau^i_{0,1} = 1440$ bytes.

Recall, even though R-IPSA uses IPS scheduling at the ONU, it will first transmit all the CBR traffic in a TW before transmitting the reported low priority data (see Section 5).

6.1 A Symmetric Traffic Scenario

In this section, the total data load ρ_d is varied from 0.16 up to 0.96 of the line rate LR which is calculated only on the base of the Ethernet frames (without preamble and IPG). Notice, due to the preamble, IPG and guard time g, the actual load on the uplink channel ρ will be considerably higher. The load ρ_d is equally distributed between all ONUs, which results in an ONU data rate between 5 and 30 Mb/s. All ONUs have equal traffic parameters. The traffic for priority 0 is simulated as a CBR stream with a rate of 8000 packets/s and packet size s_{CBR} of 70 bytes. It is chosen so as to emulate a T1 connection with a UDP/IP/Ethernet protocol stack. The amount of CBR traffic is identical for all simulations. Thus, in this section increasing the load ρ_d means adding more priority 1 and 2 traffic, while keeping the amount of CBR traffic fixed. For the traffic source of the two other priority classes, being priority 1 and 2, we use a 2-state Conditioned Markov-Modulated Bernoulli Process (C-MMBP) as described in [12]. The arrival rate in state 1 depends on the load ρ_d and is 5 times as high as in state 2. Also, the mean sojourn time

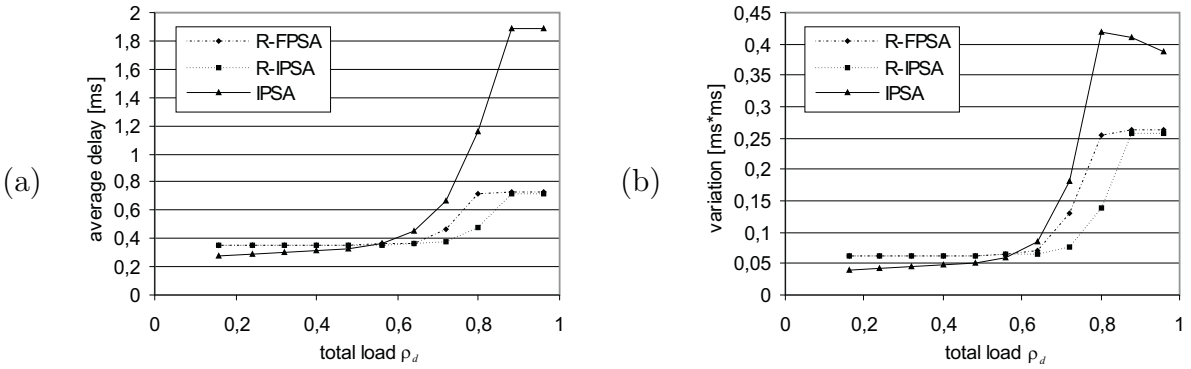


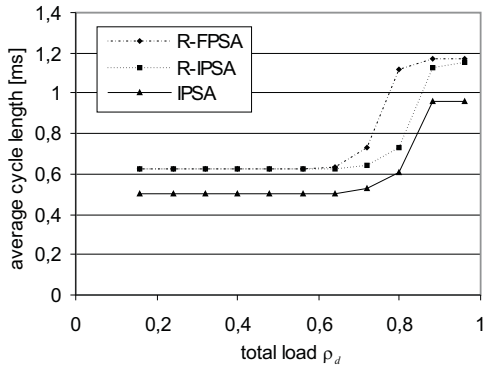
Figure 6: Symmetric Traffic: (a) the average queueing delay and (b) the delay variation of the priority 0 traffic as a function of ρ_d.

in state 1 and 2 depends on the load ρ_d and lies within $[9.4, 417.3]$ msec and $[188.5, 8346]$ msec, respectively. Thus, the background sources can be considered as bursty and strongly correlated. The packet size distribution is based on real data traces from the Passive Network and Analysis (PNA) project conducted by the National Laboratory for Applied Network Research (NLANR). The mean Ethernet frame size of the distribution used in the simulation is 455.7 bytes. Finally, the data load for the priority 1 and 2 traffic is chosen to be identical.

The average queueing delay and the delay variation of the priority 0, i.e., CBR traffic, as a function of the data load ρ_d are presented in Figure 6 for each of the three algorithms. Let us discuss these results in detail, starting with R-FPSA and R-IPSA, i.e., the algorithms that use the rate-based scheduling algorithm. First, the average delay and the delay variation remain nearly constant as $\rho_d < 0.6$. This is easily explained by Figure 7a that indicates that the cycle length is constant and equal to the minimum length[5]. Therefore, the average delay is approximately half the cycle length. Both the mean delay and the variation start to increase around $\rho_d = 0.65$ and 0.7 for R-FPSA and R-IPSA, respectively. This is exactly the point where the mean cycle length starts to increase. The fact that this happens at a higher load for R-IPSA is explained by the data throughput results shown in Figure 7b. This figure indicates that the fragmentation losses, that is, the amount of bandwidth that is lost due to not fragmenting the frames, are much smaller for R-IPSA (due to the combination of the threshold reporting mechanism and IPS scheduling), meaning that more data fits into a minimal length cycle. As the load ρ_d further increases (beyond 0.8 and 0.9, resp.), the mean delay stabilizes, which is in correspondence with the cycle length behavior, that is, the cycle length has reached its maximum size (see Figure

[5]The minimum is more than 0.5 because $\hat{B}_{min}$ bytes are allocated to the priority $p > 0$ traffic on top of the bandwidth computed by the rate-based scheme for the CBR traffic.

(a)



(b)

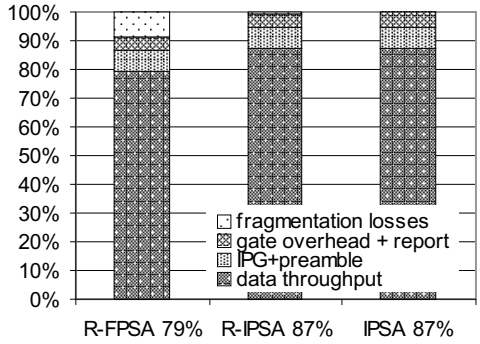


Figure 7: Symmetric traffic: (a) the average cycle length, (b) the bandwidth utilization during overload situations (simulated with $\rho_d = 0.96$).

7a). Again, due to the better efficiency of R-IPSA compared to R-FPSA, more data ρ_d is needed to reach the maximum cycle length. The higher variation for ρ_d large is caused by the burstiness of the low priority traffic (that determines the boundaries of the TWs).

For IPSA we can see a slight increment in the average queueing delay for ρ_d in the $[0, 0.6]$ area, although the cycle length is also minimal in this region. The gradual growth of the mean delay in this region can be explained as an effect caused by the IPS scheduling algorithm as follows. In low load situations, an ONU generally gets a TW that is larger than the amount of bandwidth requested. As a result, some (or all) of the data that arrived since the transmission of the last REPORT message, can also be transmitted in the TW. For very low loads this causes ONUs to report 0 bytes for all the queues (as reporting happens at the end of the TW), therefore the IPS scheduling reduces to FPS (see Section 3.2) and all the CBR traffic is transmitted first. However, as the load ρ_d increases some of the newly arrived low priority traffic will no longer be able to use the TW and therefore the ONU will report some low priority traffic. The IPS scheduling algorithm will transmit this reported low priority data before the CBR traffic; hence, while increasing ρ_d, the CBR traffic gradually shifts to the back of the TW. This explains the slight increment in the mean delay. Also, due to the burstiness of the low priority traffic (that precedes the CBR traffic in the TW), we get a slight increment in the delay variation. If we further increase ρ_d (beyond 0.6), some of the CBR traffic will no longer be able to use the TW, causing an additional delay of one cycle. Thus, more and more CBR traffic will suffer this additional delay until eventually all CBR traffic suffers a delay of approximately 1.5 cycles (explaining the strong increase in the mean delay and the delay variation). The peak in the delay variation at 0.8 corresponds to the situation where about half of the CBR traffic uses the first TW (after being generated) and the other half uses the second TW (this can be seen from the mean delay curve: the data point for $\rho_d = 0.8$ is located halfway between

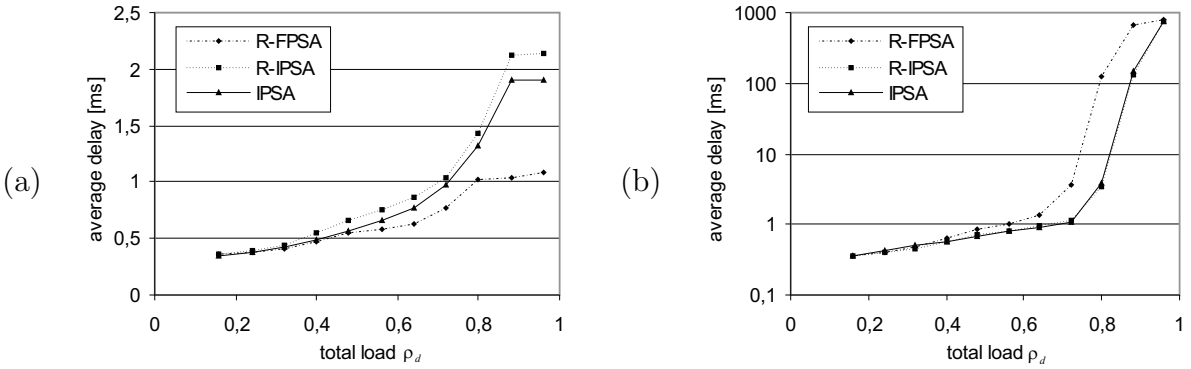


Figure 8: Symmetric Traffic: the average queueing delay for (a) the priority 1 and (b) the priority 2 traffic.

the two high load plateaus). Finally, the efficiency obtained by IPSA should be even higher compared to R-IPSA, because with R-IPSA one has only a good estimation of the CBR traffic present in the ONU and not its exact value. It is clear that IPSA pays a high price in terms of the delay to achieve this minor efficiency improvement. Notice, in Figure 7b the maximum cycle length $T_{max} = 1$ ms for ISPA (compared to 1.5 ms for R-ISPA), which cancels the slight efficiency gain that IPSA has on R-ISPA.

Before we proceed with the priority 1 and 2 results, let us discuss the impact of the threshold reporting mechanism on the efficiency results in Figure 7b. If we turn off the threshold reporting mechanism, we find nearly the same throughput results for R-FPSA (in overload situations: 79.4%). This confirms the idea that threshold reporting is useless when the ONUs use FPS, because as soon as some new priority p traffic arrives at the ONU between two TWs, all priority $q > p$ packet boundaries that where reported with the threshold mechanism in the first TW become worthless. For (R)-IPSA however we found that implementing a threshold mechanism can be worthwhile, especially in overload situations. For instance, if we turn off the threshold mechanism (and thus only report the complete queue lengths) the data throughput decreases from 87% to 79% with $\rho_d = 0.96$. Indeed, without the thresholds an ONU typically reports large bandwidth requirements (close to 1 Mbyte, in severe overload periods), which cannot be supported by a single TW, meaning that we get fragmentation losses in the majority of the TWs.

The average queueing delay for priority 1 and 2 traffic is represented in Figure 8. The mean delay of the priority 1 traffic is substantially higher for R-IPSA and IPSA compared to R-FPSA. This is an obvious consequence of the IPS scheduling, which transmits reported priority 2 traffic before unreported priority 1 traffic. Moreover, in high load situations, IPS forces some priority 1 traffic to wait an additional cycle due to earlier reported priority 2 traffic. The priority 1 delays of IPSA are slightly better than those of R-IPSA because

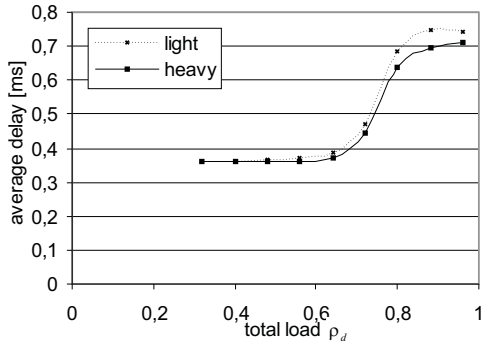
IPSA may transmit priority 1 traffic before priority 0 traffic, which is never the case with R-IPSA (see Section 5). With respect to the priority 2 traffic, we obviously find that R-FPSA achieves the worst results. The strong delay increment for ρ_d in the $[0.6, 0.8]$ region is caused by the fact that the uplink channel becomes saturated. Indeed the actual load ρ on the channel is higher than the data load ρ_d due to the overhead caused by the preamble, the IPG, guard time g and a possible idle period at the end of each TW. A less efficient algorithm causes channel saturation at a lower data load ρ_d. This is confirmed by comparing Figure 7b and Figure 8b. Finally, the nearly stable delay for ρ_d beyond 0.8 is caused by the finite buffers.

6.2 An Asymmetric Traffic Scenario

The aim of this section is to compare the performance of two types of ONUs: ONUs with a lot of best effort traffic, called *heavily loaded* ONUs, and ONUs with considerably less best effort traffic, called *lightly loaded* ONUs. We vary the load ρ_d between 0.32 and 0.96, but as opposed to the previous section we only change the amount of priority 2 traffic when doing so. For priority 0 we use exactly the same traffic source as in Section 6.1, which emulates a T1 connection with a UDP/IP/Ethernet protocol stack. For priority 1 and 2 we use a 2-state Conditioned Markov-Modulated Bernoulli Process (C-MMBP) with an arrival rate in state 1 five times as high as in state 2, but with different sojourn times. For priority 1 the sojourn time in state 1 is 21.7 ms and in state 2 it is 434 ms for both the heavy- and lightly loaded ONUs and this for all loads considered. For priority 2 the mean sojourn time in state 1 and 2 depends on the load ρ_d and lies within $[7.9, 313]$ msec and $[158.6, 6259.6]$ msec for the lightly loaded ONUs and 3 times less for the heavily loaded ONUs, respectively. As such the priority 2 traffic, i.e., best effort traffic, for the heavily loaded ONUs is 3 times as high as for the lightly loaded ones. The packet size distribution is the same as in the previous section. Given the performance results for the priority 0 traffic of IPSA in the previous section, we restrict ourselves to R-IPSA and R-FPSA. Moreover, experiments not reported here have indicated that the performance characteristics of heavily loaded and lightly loaded ONUs are very close to one another when using IPSA.

Figure 9a presents the delay of priority 0 traffic for the R-FPSA algorithm. At low loads up to $\rho_d = 0.4$ the two types of ONUs experience the same delay. This is the region where $R_{tot} \ll B_{min}$ and the allocated bandwidth, i.e., TW, for the ONUs is always larger than the requested bandwidth. As the total load gradually increases the asymmetric nature of the priority 2 traffic becomes visible. In the region $0.4 < \rho_d \leq 0.9$ the lightly loaded ONUs suffer a sightly higher (priority 0) delay in comparison with the heavily loaded ones. This stems from the fact that while all TWs are more or less of the same length when $\rho_d < 0.4$, this is no longer the case in the $[0.4, 0.9]$ area, meaning that heavily loaded ONUs tend

(a)



(b)

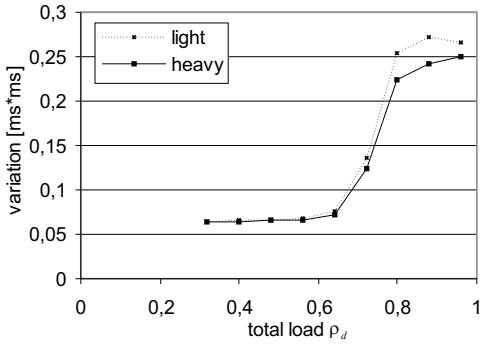


Figure 9: Asymmetric Traffic: R-FPSA (a) the average queueing delay and (b) the delay variation of the priority 0 traffic as a function of ρ_d.

to get larger TWs. Priority 0 traffic that arrives while the TW is ongoing will interject the transmission of lower priority traffic (as soon as the transmission in progress ends). Therefore, the priority 0 traffic of heavily loaded ONUs is favored in comparison with lightly loaded ONUs. Also, the mean distance between two consecutive TWs (measured from the end of the first until the start of the second) is less for heavily loaded ONUs, which implies that the mean time until the next TW is less for a packet that arrives at the ONU outside a TW. As the load increases in the $[0.4, 0.9]$ area, so does the difference between the average length of a TW assigned to a heavily loaded ONU and a lightly loaded one. Thus, the difference in delay between both types of ONUs grows. In the $\rho_d > 0.9$ region the system becomes severely saturated. Meaning that the heavily loaded ONUs will drop large amounts of best effort traffic. Thus, the throughput ratio for priority 2 traffic between heavy- and lightly loaded ONUs, which equaled 3 at low loads, begins to decrease. For instance, at $\rho_d = 0.96$ the ratio equals 1.7. Hence, the ratio between the length of the TWs allocated to heavy- and lightly loaded ONUs decreases causing the delays for priority 0 to converge to the same value. As is to be expected, the packet delay variation for priority 0 (see fig 9b) behaves similarly to the average delay.

The delay for priority 1 traffic, when using R-FPSA, behaves similarly to the priority 0 traffic, but here the difference between the delays experienced by both types of ONUs is substantially larger. The unfairness is again caused by the asymmetry in the TW sizes as explained for the priority 0 traffic (because, priority 1 traffic is allowed to interject priority 2 traffic when using (R)-FPSA). We believe that this type of unfairness is a serious drawback of any FPSA scheme. The amount of best effort traffic should not affect the performance of higher priority traffic, therefore, we advise against using an FPSA scheme. As with the priority 0 traffic, the delay of both types of ONUs at severe overload conditions converge to the same value. As can be seen from Figure 10, the delay of priority 2 traffic for the

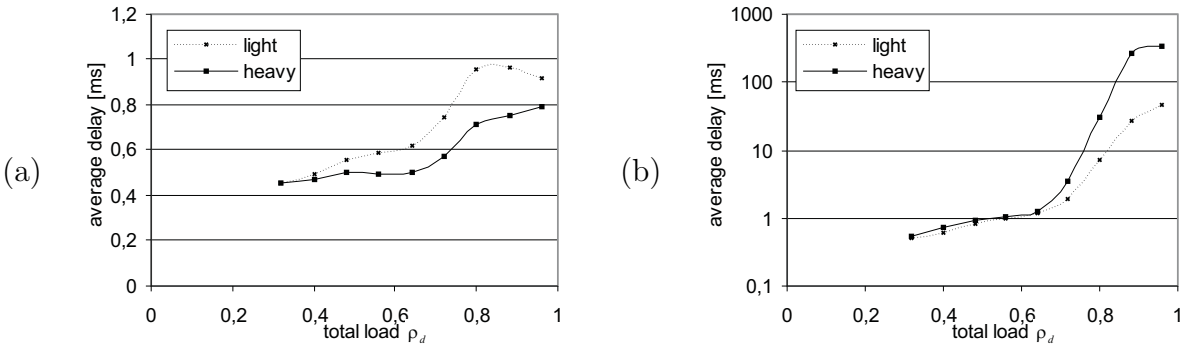


Figure 10: Asymmetric Traffic: R-FPSA (a) the average queueing delay for priority 1 (b) the average queueing delay for priority 2 as a function of ρ_d.

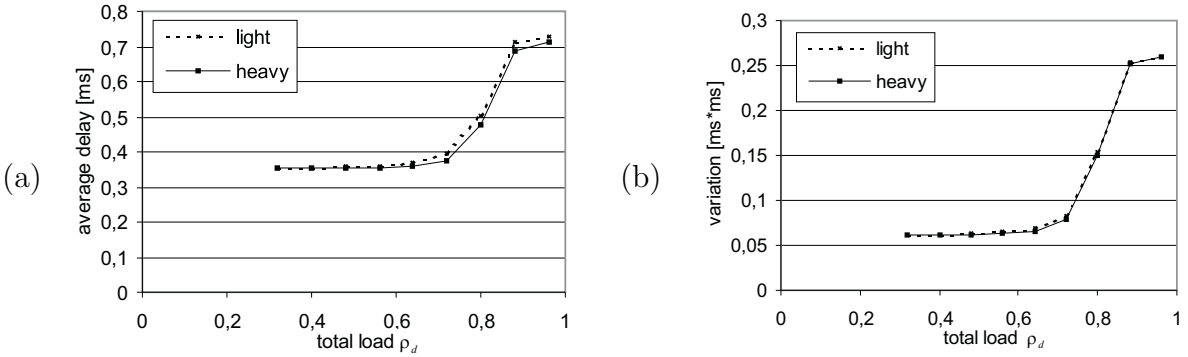


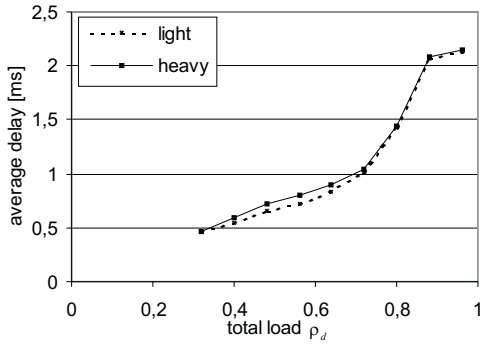
Figure 11: R-IPSA (a) the average queueing delay and (b) the delay variation of the priority 0 traffic as a function of ρ_d.

lightly loaded ONUs is obviously less than the one for the heavily loaded ONUs for all loads. The difference between both types of ONUs starts to increase significantly as soon as the system reaches its saturation point.

The average packet delay for priority 0 when using R-IPSA is presented in Figure 11a. The delay for the two types of ONUs are close to each other. Recall that with R-IPSA the packets from priority 0 are still transmitted before the reported lower priority traffic. Thus, as far as the policy to transmit priority 0 traffic is concerned, there is no difference between R-FPSA and R-IPSA. The reason for the higher delays for lightly loaded ONUs is thus the same as with R-FPSA. The packet delay variation for priority 0 of R-IPSA presented in Figure 11b, it is fair to state, based on these figures, that the asymmetry does not affect priority 0 traffic in a significant way.

The average packet delay for priority 1 is presented in Figure 12a. Due to the IPSA scheme, priority 1 traffic is no longer able to interject reported priority 2 data. Thus, one would expect the same results for heavy- and lightly loaded ONUs for all data loads ρ_d.

(a)



(b)

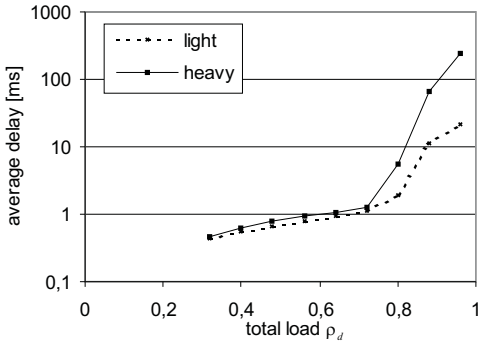


Figure 12: R-IPSA (a) the average queueing delay for priority 1 (b) the average queueing delay for priority 2 as a function of ρ_d.

Somewhat surprisingly, R-IPSA slightly favors lightly loaded ONUs in the $[0.3, 0.7]$ region. In this region the TW of all ONUs is composed of two parts: a first that corresponds to the reported (mostly priority 2) data and a second being some fair share which is identical for all ONUs (in order to reach a cycle of length B_{min}). With (R)-IPSA, newly arrived priority 1 traffic can only make use of the fair share of the TW. Now, any priority 1 traffic that arrives during the fair share is transmitted immediately[6] unless a transmission of a (priority 2) packet is ongoing (in which case it has to wait until this transmission is completed). Such ongoing transmissions occur more frequently for ONUs with more priority 2 traffic. Thus, lightly loaded priority 1 traffic has a slight advantage on the heavily loaded ONUs. Notice, when an ongoing transmission of a (priority 2) packet ends there might not be enough bandwidth left in the TW to transmit the priority 1 packet, causing an additional delay of one cycle. At a load $\rho_d = 0.32$ both curves coincide as there is hardly any priority 2 traffic generated in either type of ONU. As the load increases the priority 1 packets of lightly loaded ONUs are more likely to make use of the fair share. The mean size of the fair share however decreases as a function of the load (at $\rho_d = 0.7$ its size is zero). Thus, the load at which the advantage of the lightly loaded ONUs is maximal is about halfway between 0.32 and 0.7. Finally, Figure 12(b) represents the priority 2 delays, these are obviously higher for heavily loaded ONUs.

7 Conclusions

In this paper we proposed a dynamic bandwidth allocation algorithm for EPON that uses threshold reporting and supports different priorities. Three varieties of this algorithm, being IPSA, R-FPSA and R-IPSA, were compared under symmetric traffic conditions regard-

[6]Provided that no other priority 0 or 1 traffic is pending

ing the packet delay, the delay variation and the bandwidth efficiency. The three algorithms differ in their CBR traffic allocation method and their intra-ONU priority scheduling algorithm (FPS/IPS). We have demonstrated that with IPSA it is possible to reduce the bandwidth losses, caused by not fragmenting Ethernet frames, to almost zero. The drawback of IPSA is a strong increase of the average queueing delay and the delay variation for the highest priority traffic (CBR traffic) at high data loads. By combining IPSA with rate based scheduling for CBR traffic (R-IPSA), one can significantly reduce this high load delay increment. With respect to the bandwidth efficiency and the delay of the lower priority traffic, IPSA was shown to perform slightly better than R-IPSA. Combining FPS at the ONU with the rate based scheduling for CBR traffic (R-FPSA) results in a priority 0 performance similar to R-IPSA, but improves the average queueing delay characteristics of the priority 1 traffic. The drawback of R-FPSA compared to R-IPSA is a significant reduction in the bandwidth efficiency under high load conditions and more importantly, it significantly favors ONUs with substantial amounts of best effort traffic, causing unfairness between the end users.

Acknowledgments

The second author is a Postdoctoral Fellow of the FWO-Flanders, the first and third author were supported by Alcatel-Belgium. We would like to thank E. Six, T. Van Caenegem, C. Dessauvages and E. Ringoot of Alcatel-Belgium for their valuable comments and suggestions on this work as well as the associated editor and the reviewers for improving the presentation of the paper.

References

[1] K. Kim, "On the evolution of PON-based FTTH solutions", *Information sciences* **149** 21–30, 2003.

[2] G. Kramer, and G. Pessavento, "Ethernet Passive Optical Network (EPON): Building a Next-generation optical access network", *IEEE Communications Magazine*, February 2002

[3] G. Kramer, B. Mukherjee, and G. Pessavento, "Ethernet PON (ePON): Design and Analysis of an optical access network", *Photonic Network Comm.* **3**, pp. 307–319, July 2001.

[4] G. Kramer, B. Mukherjee, and G. Pessavento, "Interleaved Polling with Adaptive Cycle Time (IPACT): a dynamic bandwidth distribution scheme in an Optical access network", *Photonic Network Comm.* **4**, pp. 89–107, Jan 2002.

[5] G. Kramer, B. Mukherjee, S. Dixit, Y. Ye, and R. Hirth, "Supporting differentiated classes of service in Ethernet passive optical networks", *Journal of Optical Networking* **1**(8,9), pp. 280–298, 2002.

[6] C. Assi, Y. Ye, S. Dixit, and M. Ali, "Dynamic Bandwidth Allocation for Quality-of-Service Over Ethernet PONs", *IEEE Journal on Selected Areas in Communications* **21**(9), pp. 1467-1477, November 2003.

[7] S. Choi, "Cyclic Polling-Based Dynamic Bandwidth Allocation for Differentiated Classes of Service in Ethernet Passive Optical Networks", *Photonic Network Comm.* **7**(1), 87-96, 2004

[8] "Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications",ANSI/IEEE Std. 802.3, Part3, 2002

[9] "Virtual Bridged Local Area Networks", IEEE 802.1q, 1998.

[10] D. Nikolova, B. Van Houdt and C. Blondia, "Dynamic bandwidth allocation algorithms in EPON: a simulation study, in *Proc. of OptiComm*, pp. 369-380, Dallas, USA, 2003.

[11] E. Ringoot, N. Janssens, M. Tassent, J. Angeloupoulos, C. Blondia, and P. Vetter, "Demostration of Dynamic Medium Access Control for APON and superPON", in *Proc. of Globecom*, San Antonio, Texas, USA, 2001.

[12] B. Van Houdt, C. Blondia, O. Casals, and J. García", "Performance Evaluation of a MAC protocol for broadband wireless ATM networks with QoS provisioning", *Journal of Interconnection Networks (JOIN)* **2**(1), pp. 103–130, 2001.