

# Dynamic bandwidth allocation algorithms in EPON: a simulation study

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

## ABSTRACT

In this paper we present a dynamic bandwidth allocation algorithm for 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, 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. We show that by introducing a specific intra-ONU priority scheduling algorithm, which takes the reported values into account, the bandwidth can be fully utilized. However, this scheduling algorithm causes an increased packet delay and delay variation for CBR traffic. In order to eliminate this drawback, we combine this scheduling algorithm with a rate-based scheme for the highest priority (CBR) traffic. This combined algorithm provides an interesting tradeoff between the efficiency, which is still near to the optimal, and the delay characteristics of time critical applications. Finally, we also include a comparison with a standard intra-ONU priority scheme.

**Keywords:** EPON, passive optical networks, FTTH, Multipoint Control Protocol (MPCP), Dynamic Bandwidth Allocation (DBA) algorithm, threshold reporting

## 1. INTRODUCTION

A passive optical network (PON) is a subscriber access network technology that provides high bandwidth capacity. 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).

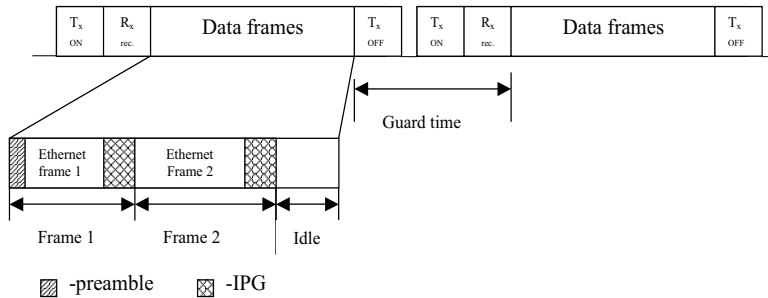
Two standardization bodies are currently working on PONs. The International Telecommunication Union has already a family of standards regarding PON<sup>1</sup> with Asynchronous Transfer Mode (ATM) as the data-link layer and is currently developing a second one with a new, still to be standardized, data-link layer. Within the IEEE there is a working group (802.3ah) standardizing Ethernet based PONs (EPON).<sup>2</sup> The Ethernet protocol is highly deployed in local area networks (LANs) and it is also becoming an emerging technology for wide area networks (cfr. Metro Ethernet). Therefore, it is natural that the subscriber access network also offers an Ethernet solution.

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 allocating Transmission Windows (TWs), which can have variable lengths. An ONU is only allowed to transmit during the TWs allocated itself. In order to inform the OLT about its bandwidth requirements, ONUs use REPORT messages\* that are also transmitted (along with the data) in the TW. Frames are never fragmented in EPON, therefore, the IEEE working group also introduced the concept of reporting with "thresholds" in order to achieve a higher bandwidth efficiency (see Section 2.2.1).

---

{dessislava.nikolova, benny.vanhoudt, chris.blondia}@ua.ac.be

\*See Section 2.2.1.



**Figure 1.** Transmission window and frame format

A time division multiple access scheme with fixed length TWs assigned to each ONU was analyzed in.<sup>3</sup> The disadvantage of this scheme is, obviously, that no statistical multiplexing is possible. Another, more efficient, algorithm where the length of the TWs depends on the ONU's current bandwidth requirements, was proposed in<sup>4</sup> and extensively studied and improved in.<sup>5</sup> 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 this ONU's 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). This basic scheduling algorithm was also enriched with a two-stage queuing system at the ONU and a CBR credit scheme (see<sup>5</sup> for details). Our work differentiates itself from this prior work by studying the impact of the threshold reporting on the delay and efficiency of the system. Also, the scheduler introduced in this paper should provide better QoS guarantees because it realizes both intra- and inter-ONU priority scheduling.

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. In section 6 the simulation results are presented, while in Section 7 conclusions are drawn and future work is discussed.

## 2. EPON: A STATE-OF-THE-ART

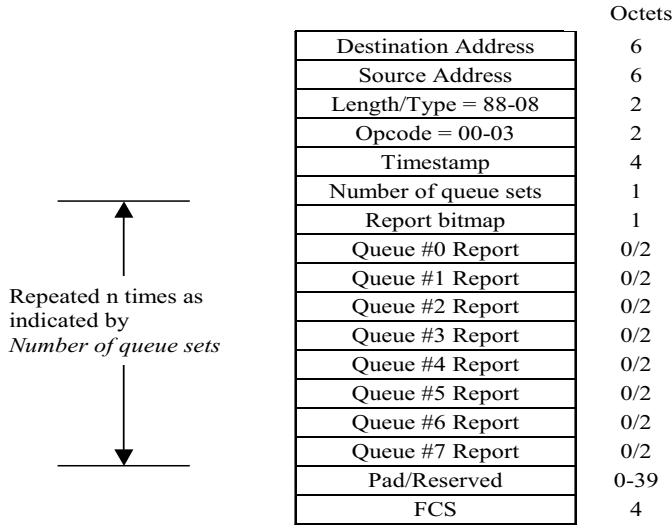
This section provides an overview of the current state of the EPON standardization efforts made by the 802.3ah working group.

### 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).<sup>6</sup> Between the TW of two ONUs there is 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 ( $R_{rec}$ ) (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.<sup>7</sup> 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. The ONU can also be forced to send a REPORT message within a TW. All MPCP



**Figure 2.** Format of a REPORT message

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 of with a length of 101 bytes (including preamble and IPG), it will send only 9 frames in this TW, which leaves  $1000 - 9 \cdot 101 = 91$  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 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 be 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 timestamp indicates the local time when the message is transmitted by the ONU. In MPCP, ONU  $i$  can have several threshold values  $\tau_{j,l}^i$  for queue  $j$  (with  $l = 1, \dots, 13$ ). These thresholds are used by the ONU to determine the values of the QR fields (see Figure 3). Denote  $\beta_j^i(n)$  as the total size of the first  $n$  packets waiting in queue  $j$  at ONU  $i$ . ONU  $i$  is said to use the threshold  $\tau_{j,l}^i$  if it includes  $\beta_j^i(n)$  as a QR in the REPORT message, where  $\beta_j^i(n) < \tau_{j,l}^i < \beta_j^i(n+1)$ . 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.

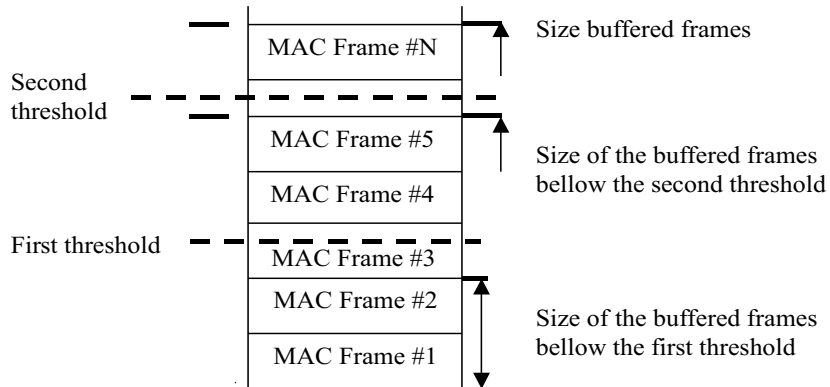


Figure 3. Threshold Reporting

### 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. An execution of the scheduling algorithm produces a set of ONU assignments  $a_i$ , where  $a_i$  indicates the amount of bytes that an ONU 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 are produced using the results of an execution, to reach the most distant ONU before the start of cycle  $n + 1$  (which coincides with the end of cycle  $n$ ) because the first TW of cycle  $n + 1$  could be assigned to this ONU. This in its turn implies that the ONUs scheduled at the end of a cycle are somewhat disadvantaged. Therefore, we decided to make the ONU order within a cycle random.

The starting time  $s_i$  of the TW of the  $i$ -th ONU in cycle  $n + 1$ , for  $i = 1, \dots, N$ , 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.<sup>2</sup>

#### 3.1. REPORT message generation at the ONU

Several thresholds, denoted as  $\tau_{j,l}^i$  for  $l = 1, \dots, 13$ , 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<sup>†</sup>.

In the current algorithm the last threshold for each queue equals infinity (see Section 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  $j = 0, \dots, P - 1$  that has a length different from 0. Also, the QRs for the queue with the highest priority (priority 0) are created first. The following 3 step algorithm is used to generate a REPORT message:

<sup>†</sup>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  $j$ .

- STEP 1: ONU  $i$  generates, for each queue  $j$ , a set of 13 values  $v_{j,l}^i = \max_n \{\beta_j^i(n) \mid \beta_j^i(n) < \tau_{j,l}^i\}$ . Next, define the set  $V_j^i$  as  $\{v_{j,l}^i \mid 0 \neq v_{j,l}^i \neq v_{j,l-1}^i\}$  and let  $\theta_j^i = 1$  if  $V_j^i$  is not empty and let  $\theta_j^i = 0$  otherwise. Ideally, ONU  $i$  would like to include a QR for each value  $v_{j,l}^i$  in  $V_j^i$  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  $V_j^i$  is not empty, we find that ONU  $i$  includes at most  $x = \lfloor (39 - 2 \sum_{j>0} \theta_j^i) / 3 \rfloor$  QRs for queue 0 (2 bytes are used for the QR, 1 for the corresponding bitmap). Hence, ONU  $i$  will include  $n_0^i = \min(x, |V_0^i|)$  QRs for queue 0.
- 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  $0, \dots, j-1$  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. 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.

The 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.

### 3.2. Scheduling at the ONU

Two types of scheduling at the ONU are considered in this paper.

**Full priority scheduling(FPS):** With this scheduling algorithm we mean the normal priority scheduling scheme, i.e., the packets are send 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 are likely to have changed. For instance, if some priority  $p$  packets arrived during this interval, then these arrivals will destroy the usefulness of the threshold reporting for all the priority  $q > p$  traffic (that is, the lower priority traffic).

**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 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.<sup>1</sup> The IPS scheme coincides with the two stage buffer scheme proposed in.<sup>5</sup>

### 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 following fields  $r_{j,l}^i$ , for ONU  $i = 1, \dots, N$ , queue  $j = 0, \dots, P-1$  and  $l = 1, \dots, 13$ : 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  $r_{j,l}^i$ . Next, it scans the REPORT message and whenever it encounters a QR that corresponds to queue  $j$ , it enters the value  $v$  found in this QR in  $r_{j,x}^i$ , where  $x$  is the smallest index such that  $v \leq \tau_{j,x}^i$ .
- STEP 2a: If the field  $r_{j,13}^i$  is non-empty and if  $y < 13$  is the largest index  $y$  for which  $r_{j,y}^i$  is non-empty, then we set  $r_{j,l}^i = \tau_{j,l}^i$  for all  $y < l < 13$ .
- STEP 2b: If the field  $r_{j,13}^i$  is empty and if  $y$  is the largest index  $y$  for which  $r_{j,y}^i$  is non-empty, then we set  $r_{j,l}^i = r_{j,y}^i$  for all  $y < l \leq 13$ .
- STEP 3: Finally, the OLT will increase all  $13(P-1)$  entries  $r_{j,l}^i$ , for  $j = 1, \dots, P-1$  and  $l = 1, \dots, 13$ , by  $\sum_{k < j} r_{k,13}^i$ .

This procedure guarantees that  $r_{j,l}^i$  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  $\beta_j^i(n) \leq \tau_{j,l}^i$ . Recall  $\beta_j^i(n)$  was defined as the size of the first  $n$  packets in queue  $j$  of ONU  $i$ . Thus,  $r_{P-1,13}^i$  represents the total number of bytes reported by ONU  $i$ .

### 3.4. Scheduling at the OLT

The scheduling at the OLT is based on the table with the  $r_{j,l}^i$  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  $r_{j,l}^i = 0$  for all  $j$  and  $l$ . Next, the OLT computes the following sums:

$$R_{j,l} = \sum_i r_{j,l}^i, \quad (1)$$

for all  $j$  and  $l$ . Notice,  $R_{j,l} \leq R_{k,m}$  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.,  $r_{P-1,13}^i$ ) 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,  $a_i = r_{P-1,13}^i$ .
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. (i) If  $l+1 \neq 13$ , the assignments for the ONUs are equal to what they have reported up to this queue and threshold, i.e.,  $a_i = r_{j,l}^i$ . An appropriate choice of the threshold values  $\tau_{j,l}^i$  can in this particular case guarantee that  $R_{j,l} > \hat{B}_{min}$ . (ii) If, on the other hand,  $l+1 = 13$ , we start by setting  $a_i = r_{j,l}^i$  and  $A = \sum_i a_i$ . Next, we increment  $a_i$  in an iterative manner as long as  $A < \hat{B}_{max}$  as follows. Let  $x_i = r_{j,13}^i - a_i$ , 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  $r_{j,l}^i$  bytes in such a way that  $a_i \leq r_{j,13}^i$ .

The distinction between case 3(i) and 3(ii) is made because  $\tau_{j,12}^i$  will, in general, be much smaller than the buffer size of queue  $j$ .

## 4. THRESHOLD ASSIGNMENT

At the current stage of the development of the EPON standard it is still not decided how the thresholds will be conveyed 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. Another possibility to create dynamic thresholds is to include them in the GATE messages. The number of the thresholds per queue will 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, you could end up spending several Mbit/s to keep the thresholds up to date.

Our proposal is that for each queue  $j$  of ONU  $i$  only one threshold is assigned (being  $\tau_{j,1}^i$ ) and all others are derived from it. The assignment can be static or dynamic. A simple and logical way is to use linear dependence

$$\tau_{j,l}^i = l \tau_{j,1}^i, \quad (2)$$

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. Finally, in order for the OLT to have a complete view of the bandwidth requirements of an ONU,  $\tau_{j,13}^i$  equals infinity.

## 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. APON,<sup>18</sup>). 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.<sup>5</sup>

### 5.1. Rate-Based Scheduling

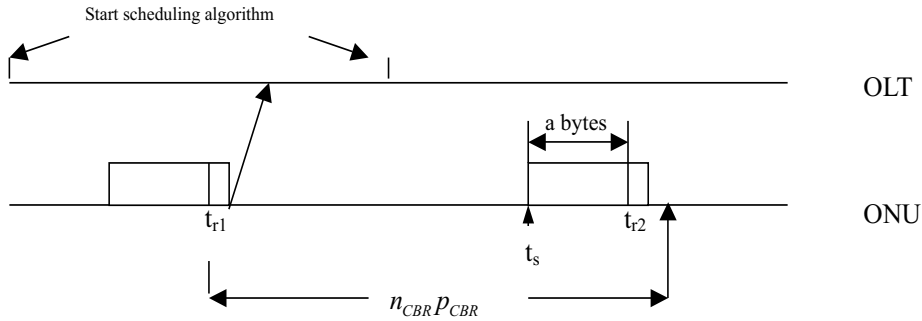
The idea behind rate-based scheduling is to predict the number of packets  $n_{CBR}$  that arrive at the ONU (from each CBR application) during the time interval spanned by two consecutive TWs  $W_1$  and  $W_2$ . For doing so, we can make use of the formula provided in<sup>5</sup>:

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

where  $p_{CBR}$  is the period of the CBR application<sup>‡</sup>,  $h$  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 4).

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  $b_{CBR}^i$  required for the CBR traffic of ONU  $i$ . The value  $b_{CBR}^i$  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

<sup>‡</sup> $p_{CBR}$  can be obtained as the reciprocal of the rate expressed in packets per second.



**Figure 4.** The rate allocation scheme

$B_{CBR} = \sum_i b_{CBR}^{i,max}$  and  $b_{CBR}^{i,max}$  equals the amount of bytes required for ONU  $i$  assuming that the distance between the two consecutive TWs is maximal, i.e.,  $2T_{max}$ .

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  $b_{CBR}^i$  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.

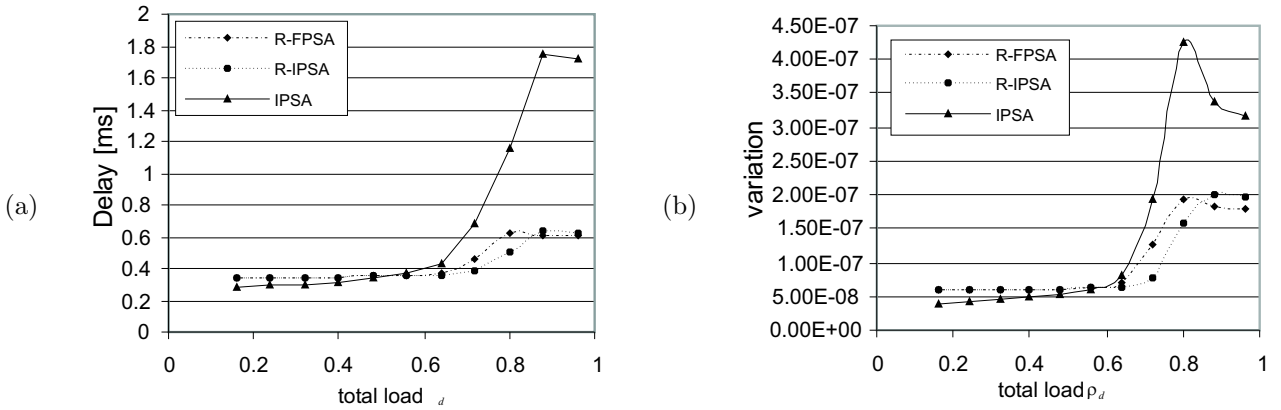
## 6. SIMULATION RESULTS

### 6.1. Simulation Setup

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 8 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  $1\mu s$ . 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 total data load  $\rho_d$  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  $\rho$  will be considerably higher. The load  $\rho_d$  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, the load  $\rho_d$  is increased by 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.<sup>9</sup> The arrival rate in state 1 depends on the load  $\rho_d$  and is 5 times as high as in state 2. Also, the mean sojourn time in state 1 and 2 depends on the load  $\rho_d$  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. The data load for the priority 1 and 2 traffic is chosen to be identical.





**Figure 5.** (a) the average queuing delay and (b) the delay variation of the priority 0 traffic as a function of  $\rho_d$ .

The cycle length 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  $\tau_{0,1}^i = 2160$  and  $\tau_{1,1}^i = \tau_{2,1}^i = 1538$  bytes for all  $i$ , unless otherwise stated. The other thresholds  $\tau_{j,l}^i$  are obtained from these as explained in Section 4. Let us describe the three scheduling algorithms:

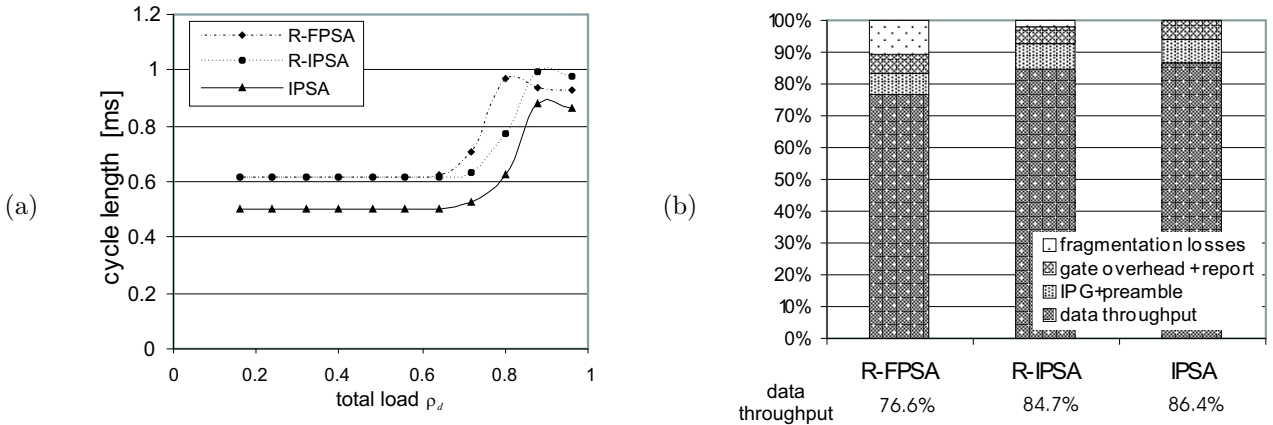
- R-FPSA: In this setup, we use the rate-based scheduling algorithm for the CBR traffic, i.e., highest priority traffic, and the FPS scheduling algorithm at the ONU.
- R-IPSA: In this case, we combine the rate-based scheduling algorithm for CBR traffic with IPS scheduling at the ONU.
- IPSA: We do not 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.  $\tau_{0,1}^i = 1440$  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).

## 6.2. Numerical Results

The average queuing delay and the delay variation of the priority 0, i.e., CBR traffic, as a function of the data load  $\rho_d$  are presented in Figure 5 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 6a that indicates that the cycle length is constant and equal to the minimum length<sup>§</sup>. 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 6b. 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 and IPS scheduling), meaning that more data fits into a minimal length cycle. As the load  $\rho_d$  further increases (beyond 0.8 and 0.9, resp.), we observe a slight decrement in the delay and delay variation, which is in correspondence with the cycle length behavior (see Figure 6b). In this load region,  $R_{tot} > \hat{B}_{max}$  (see Section

<sup>§</sup>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.

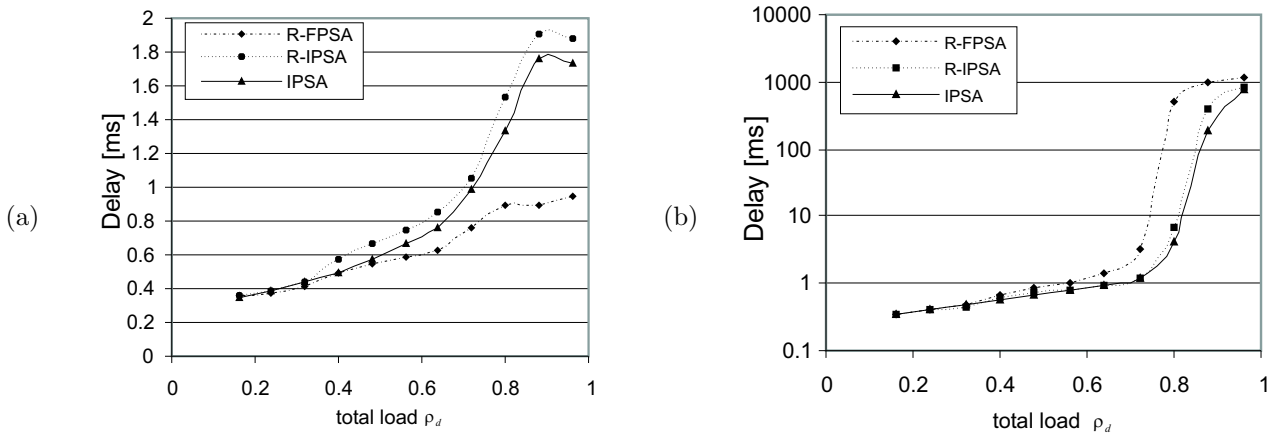


**Figure 6.** (a) the average cycle length, (b) the bandwidth utilization at high loads  $\rho_d$ .

3.4), while with  $\rho$  increasing the mean number of ONUs that request bandwidth (of priority  $p > 0$ ) also increases, causing a shorter mean cycle length. Again, due to the better efficiency of R-IPSA compared to R-FPSA, we need a higher data load  $\rho_d$  to reach the maximum cycle length. The higher variation for  $\rho_d$  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  $\rho_p$  in the  $[0, 0.6]$  area, although the cycle length is also minimal in this region. The increasing delay 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 actually 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  $\rho_d$  increases some of the newly arrived low priority traffic will not be able to use the TW anymore 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, the CBR traffic shifts to the back of the TW as the load increases. This explains the slight increment in the mean delay. Also, due to the burstiness of the low priority traffic, we get a slight increment in the delay variation. If we further increase  $\rho_d$  (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, because the data point for  $\rho_d = 0.8$  is located halfway between the two high load plateaus). Finally, the efficiency obtained by IPSA is even higher compared to R-IPSA, because with R-IPSA you only have a good estimation of the CBR traffic present in the ONU and not the exact value. It is clear that IPSA pays a high prize in terms of the delay to achieve this minor efficiency improvement.

The average queueing delay for priority 1 and 2 traffic is represented in Figure 7. 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  $\rho_d$  in the  $[0.6, 0.8]$  region is caused by the fact that the uplink channel becomes saturated. Indeed the actual load  $\rho$  on the channel is higher than the data load  $\rho_d$  due to the overhead caused by the preamble, the IPG, guard time  $g$  and possible the idle period at the end of a TW. A less efficient algorithm



**Figure 7.** the average queueing delay for (a) the priority 1 and (b) the priority 2 traffic.

causes channel saturation at a lower data load  $\rho_d$ . This is confirmed by comparing Figure 6b and Figure 7b. Finally, the nearly stable delay for  $\rho_d$  beyond 0.8 is caused by the finite buffers.

## 7. CONCLUSIONS AND FUTURE WORK

In this paper we proposed a new 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 regarding 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 and, to a lesser extent, an increment in the priority 2 traffic delay characteristics.

Future work will consist of further clarifying the following issues: what is the tradeoff between implementing dynamic or fixed length cycles, what is the exact impact of the threshold assignment scheme on the performance and is it worth to implement threshold reporting without IPS scheduling at the ONU? Finally, we are exploring the possibility to replace the analytical C-MMBP input traffic for the priority 1 and 2 traffic by a trace-driven input source.

## Acknowledgements

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.

## REFERENCES

1. "A Broadband optical access system with increased service capabilities using dynamic bandwidth assignment", ITU-T G.983.4, Nov 2001.
2. "Media Access Control Parameters, Physical Layers and Management Parameters for subscriber access networks", IEEE Draft 802.3ah/D1.414, March 2003.

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. "Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications", ANSI/IEEE Std. 802.3, Part3, 2002
7. "Virtual Bridged Local Area Networks", IEEE 802.1q, 1998.
8. E. Ringoot, N. Janssens, M. Tassent, J. Angeloupoulos, C. Blondia, and P. Vetter, "Demonstration of Dynamic Medium Access Control for APON and superPON", in *Proc. of Globecom*, San Antonio, Texas, USA, 2001.
9. 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.