QoS issues in EPON

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

One of the key issues for the Ethernet passive optical network (EPON) is the ability of the Multi-point control protocol (MPCP) to support quality of service (QoS). In[3] a variety of dynamic bandwidth allocation algorithms, which make use of the threshold reporting, were compared with respect to their QoS support and efficiency. Two of these algorithms, referred to as R-FPSA and R-IPSA, are studied in more detail in this paper by considering both symmetric and asymmetric traffic conditions. Moreover, a slight improvement has been made to the scheduling algorithm to further improve its performance. We demonstrate that the QoS support of high priority traffic is not influenced by the presence of best effort traffic when using R-IPSA, as opposed to R-FPSA that favors ONUs with lots of best effort traffic. While slightly higher delays for some priorities are observed with R-IPSA, it realizes better fairness and efficiency when compared to R-FPSA.

1. INTRODUCTION

An Ethernet passive optical network (EPON) is a subscriber access network using the Ethernet protocol as the data-link layer. Generally PONs are point to multipoint networks 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.

EPON is currently being standardized by an IEEE working group (802.3ah).[1] 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. The OLT assigns the TWs via so called GATE messages. An ONU is only allowed to transmit during the TWs allocated to itself. 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.[2] 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 multi-point control protocol (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. In combination with EPON two types of scheduling algorithms are discussed in the literature.[3,4] One is the standard or full priority scheduling algorithm (FPSA), which is also described in.[2] The other one is the interval priority scheduling algorithm (IPSA),[3] which is very close to the two stage buffer scheme described in.[4] IPSA generally outperforms FPSA qua efficiency but the packet delay for time critical applications is somewhat higher. In order for EPON to be able to offer QoS guarantees toward such time critical applications (especially for constant bit rate (CBR) traffic) rate-based scheduling has been discussed.[3,4] It provides QoS guarantees and when combined with IPSA (referred to as R-IPSA) it realizes an interesting tradeoff between the efficiency, which is still near optimal and the delay characteristics of time critical applications. The nearly optimal efficiency can be realized by means of the "threshold reporting" mechanism, as demonstrated in.[3] The fact that the OLT cannot control the scheduling at the ONUs creates a premises for ONUs with more total bandwidth requirements to be privileged.

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

Our aim in this article is to study and compare the QoS properties, such as the mean delay and the delay variation, of ONUs with different bandwidth requirements. We shall consider two types of intra-priority ONU scheduling combined with a rate-based DBA algorithm at the OLT. We use a slightly modified version of the upstream bandwidth allocation algorithm proposed in.[3] This modification affects the performance at high loads by reducing the mean and variation of the cycle length (and consequently of some QoS parameters).

The paper is structured as follows. In Section 2 we give an overview of the DBA algorithms by describing the general principles, the "threshold reporting" and the modified scheduling algorithm that operates at the OLT. In Section 3 we demonstrate the effect of the proposed modification when compared with the algorithm in.[3] To demonstrate the QoS issues arising from the choice of the intra-priority ONU scheduling, we present in Section 4 the results of simulations of the same algorithm but with different traffic profiles in the ONUs. Finally, in Section 5 some conclusions are drawn.

2. DBA ALGORITHM WITH THRESHOLDS REPORTING

This section introduces the modified bandwidth allocation algorithm. The 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, where N is the number of registered ONUs (because a REPORT requires 84 bytes). For the rate-based scheduling as proposed in[3] a certain amount of bandwidth B_{CBR} is reserved for the rate-based assignments. To account for this, we further reduce the maximum number of bytes $\hat{B}_{max}$ to be scheduled to $\bar{B}_{max} = \hat{B}_{max} - B_{CBR}$, details on how B_{CBR} is computed can be found in.[3]

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) or $w_i = a_i + 84 + g + b^i_{CBR}$ for the rate-based scheduling, where b^i_{CBR} is the amount of bandwidth allocated to the CBR traffic of ONU i (see,[3] Section 5). A GATE message will consist of the start time and the length w_i of the TW.

2.1. Reporting with thresholds

The ONUs inform the OLT about their bandwidth requirements using REPORT messages. These messages can contain up to 13 Queue Reports (QRs), where a QR basically holds a statement about one of the ONU queues (e.g., its total length or its length up to a certain threshold). The MPCP protocol allows several QRs for 1 queue in one REPORT message, which makes "threshold reporting" possible. A detailed description for generating and processing REPORT messages at the ONU and the OLT is given in.[3] A summary is included here for reasons of completeness. Several thresholds, denoted as $\tau^i_{j,l}$ for $l = 1, \dots 13$, are associated to each queue j of ONU i, where the condition $\tau^i_{j,l} < \tau^i_{j,l+1}$ is satisfied. The last threshold $\tau^i_{j,13}$ for each queue j equals infinity to allow reporting of the total number of bytes waiting in a queue. ONU i is said to use the threshold $\tau^i_{j,l}$ if it includes the total size of the first n packets waiting in its queue j, denoted with $\beta^i_j(n)$, as a QR in the REPORT message, where $\beta^i_j(n) < \tau^i_{j,l} < \beta^i_j(n+1)$ (see. Figure 1). Ideally, the ONU would like to use as many thresholds as required for each queue j, giving the OLT a detailed description of its queue state. However, a REPORT can only hold 13 QRs, therefore some selection has to be made. First, the ONU includes in each REPORT message at least one QR for each queue $j = 0, \dots, P-1$ that has a non zero contents, where P is the maximum number of supported queues. Next, the ONU continues by creating the QRs for the queue with the highest priority (priority 0), until either all QRs have been filled or it has included all the QRs required to reflect the contents of the priority 0 queue. In the second case, it continues by creating QRs for the priority 1 queue and so on. The number of QRs for a given queue is restricted (≤ 13) and can be derived knowing the number of non empty queues (see[3]).

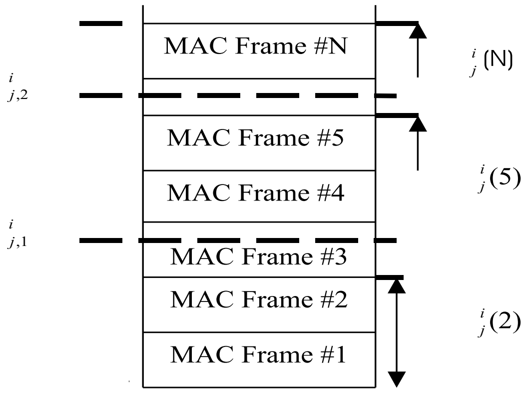


Figure 1. Threshold Reporting

The OLT maintains a table, based on which it calculates the bandwidth assignments a_i. The entries in this table are filled in upon receiving a REPORT message from an ONU. For each ONU i, queue j, and possible threshold $\tau^i_{j,l}, l = 1, \ldots, 13$, it keeps a record $r^i_{j,l}$ corresponding to the QR for this threshold. An ONU does hardly ever uses 13 thresholds when reporting for a single queue due to lack of QRs (or if a threshold is less than a packet size then it does not use a QR to report a 0 value). If a threshold $\tau^i_{j,L}$ is not used, the OLT assumes that the QR equals 0 unless the next QR included in the REPORT message is the infinity threshold in which case the QR is assumed to be $\tau^i_{j,L}$. The OLT fills the record $r^i_{j,l}$ in the table with the corresponding QRs for $j = 0$ while for $j > 0$, it enters the QR value plus $r^i_{j-1,13}$. In this way $r^i_{j,l+1} \geq r^i_{j',l}$ and $r^i_{j,0} \geq r^i_{j+1,13}$, where $j = 0, \ldots P-1$ and P is the maximum number of allowed priority queues (being 8). Notice, $r^i_{P-1,13}$ equals the total number of bytes waiting in all queues of ONU i.

2.2. scheduling at the OLT

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, $R_{j,l} \leq R_{k,m}$ 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}:

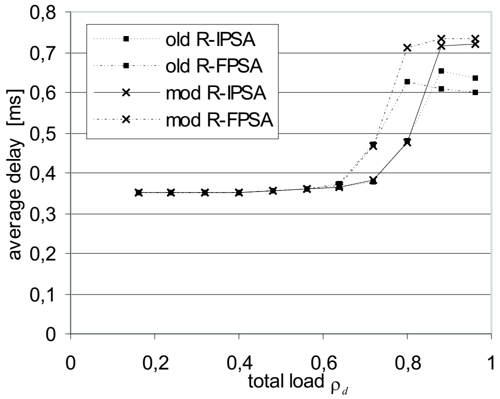
1. $R_{tot} < \hat{B}_{min}$: In this case the assignment lengths a_i of the ONUs are the amount they have requested (i.e., $r^i_{P-1,13}$) plus a fair share of the remaining amount of bandwidth up to $\hat{B}_{min}$ (i.e., $(\hat{B}_{min} - R_{tot})/N$).

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.

   (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.

   (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

(a) 

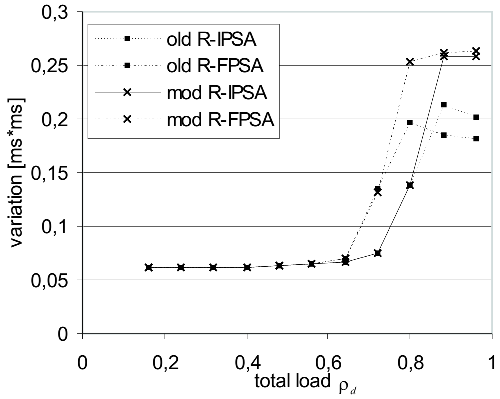
(b) 

Figure 2. (a) the average queuing delay and (b) the delay variation of the priority 0 traffic as a function of ρ_d.

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 $a_i \leq r^i_{j,13}$.

The distinction between the algorithm presented in[3] and the above presented is in 3(i). In the presented algorithm the maximum difference $max(\bar{B}_{max} - \sum_i a_i) < max(\tau^i_{j,l} - \tau^i_{j,l-1})$, while in[3] $max(\bar{B}_{max} - \sum_i a_i) < (\sum_{i=1}^N \tau^i_{j,l} - \sum_{i=1}^N \tau^i_{j,l-1})$.

3. PERFORMANCE COMPARISON

In this section we present the results from the algorithm proposed in[3] and its the modified version described in 2.2. For the simulations we have used the same setup and traffic profile as the one described in.[3] For reasons of completeness a brief description is given below. 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 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 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 $\bar{B}_{max} = 111992$ bytes, unless otherwise stated (see Section 2 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. The other thresholds $\tau^i_{j,l}$, for $l > 1$, are obtained from $\tau^i_{j,1}$ as follows: $\tau^i_{j,l} = l\tau^i_{j,1}$ for all i, j. We consider two DBA algorithms: R-FPSA and R-IPSA. Both these algorithms use the rate-based scheduling for CBR traffic, but make use of a different scheduling algorithms at the ONU. R-FPSA uses full priority scheduling (FPS), whereas R-IPSA makes use of interval priority scheduling (IPS). It should be noted that 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.

Figures 2 and 3 present both the mean delay and the delay variation of the priority 0 traffic as well as the mean and variation of the cycle length. Due to the modification made to the scheduling algorithm as described in 3(i) of Section 2.2, the maximum difference between B_{max} and the cycle length C is, under high load conditions, reduced from $1538N$ to $\max_{i,j}(\tau^i_{j,1})$ or 1538 bytes. We do not consider $\tau^i_{0,1}$ because on one hand as we are simulating rate-based scheduling algorithms normally for the traffic for priority 0 is already accounted for and on the other hand the reported traffic for this priority 0 never exceeds B_{max}. As a result, the mean cycle length of the modified algorithm is larger compared to the old scheme, while its variation is less (under high load conditions). This causes a higher mean delay and delay variation for the priority 0 traffic, while the increased cycle length obviously results in a better efficiency. For R-FPSA it rises from 76.7% to 79.4% and for R-IPSA from 86.3% to 87.2%.

(a) (b)

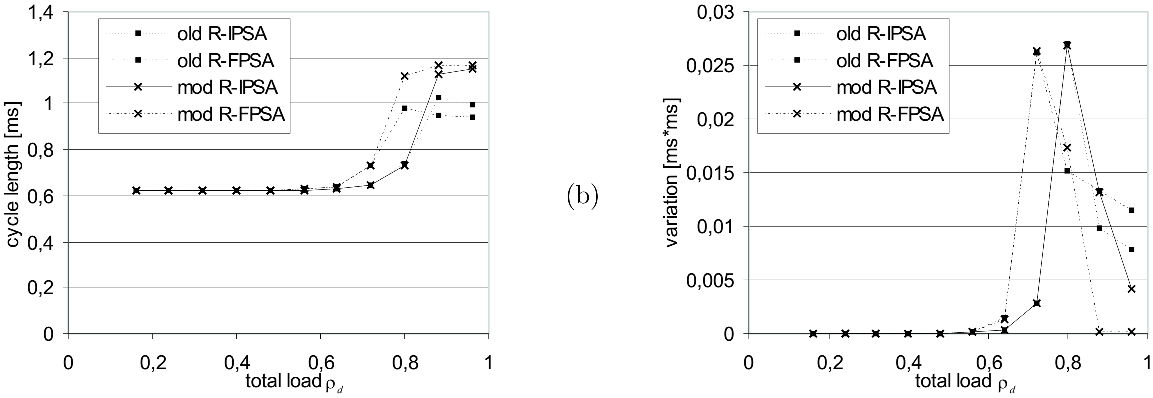


Figure 3. (a) average cycle length and (b) cycle length variation as a function of ρ_d.

Further, with the old algorithm we found that in the high load area (beyond $\rho > 0.8$, resp. 0.9) the mean delay slowly decreased as a function of the load ρ. This was caused by the fact that higher loads imply that more ONUs are actively competing for bandwidth within a cycle. Thus, the difference between B_{max} and the cycle length C under high load conditions grew as a function of ρ, creating shorter cycles and therefore a small reduction in the mean delay of the priority 0 traffic. With the modified scheme the number of active ONUs in a cycle has no real impact on the difference between B_{max} and C (under high load conditions). Therefore, the minor drop in the average delay is not present.

4. PERFORMANCE EVALUATION

In this section we present and discuss the results obtained from simulating the modified DBA algorithm for ONUs with two types of traffic load. Type 1 has a lot of best effort traffic and is referred to as “heavy-loaded” ONUs, while type 2 contain 3 times less best effort traffic and are called “light-loaded” ONUs. We vary the load ρ between 0.32 and 0.96 by only varying the amount of best effort traffic, that is priority 2 traffic. For priority 0 we use exactly the same traffic source as in,[3] which emulates a T1 connection with a UDP/IP/Ethernet protocol stack. For priority 1 and 2 we make use of a 2-state Conditioned Markov-Modulated Bernoulli Process (C-MMBP) as proposed in[5] with 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 all ONUs for all loads. 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 “light-loaded” ONUs. The “heavy-loaded” ONUs have similar properties except that their mean sojourn times are 3 times shorter. This means that for all loads ρ the amount of priority 2 traffic for the “heavy-loaded” ONUs is 3 as high as the “light-loaded” ones. 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.

4.1. Numerical results

Recall that the traffic for priority 0 and 1 doesn’t change as the load varies. So the changes in the delay are solely due to the influence of priority 2 traffic and the scheduling in the ONUs and OLT.

Figure 4a presents the delay of priority 0 traffic for the R-FPSA algorithm. At low loads up to $\rho = 0.4$ the two types of ONUs experience the same delay. This is the region where $R_{tot} << B_{max}$ and the allocated bandwidth, i.e., TW, for the ONUs is always larger than the requested. As the total load increases the influence of having more priority 2 traffic becomes more visible. In the region $0.4 < \rho_d \leq 0.9$ the delay (for priority 0 traffic) of the light loaded ONUs becomes slightly higher compared to the heavy loaded ones. This stems from the fact that while all TWs are more or less of the same length when $\rho < 0.4$, this is no longer the case in the $[0.4, 0.9]$ area, meaning that heavy loaded ONUs tend to get larger TWs, therefore the mean distance between

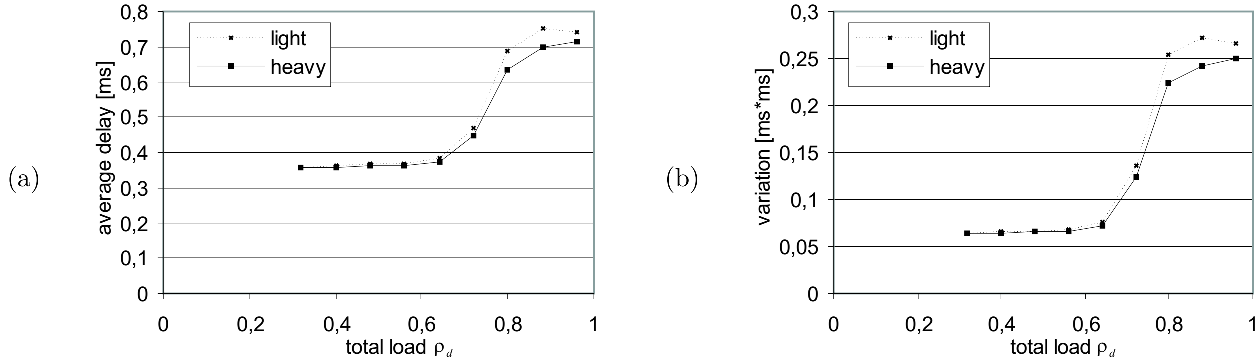


Figure 4. R-FPSA (a) the average queuing delay and (b) the delay variation of the priority 0 traffic as a function of ρ_d.

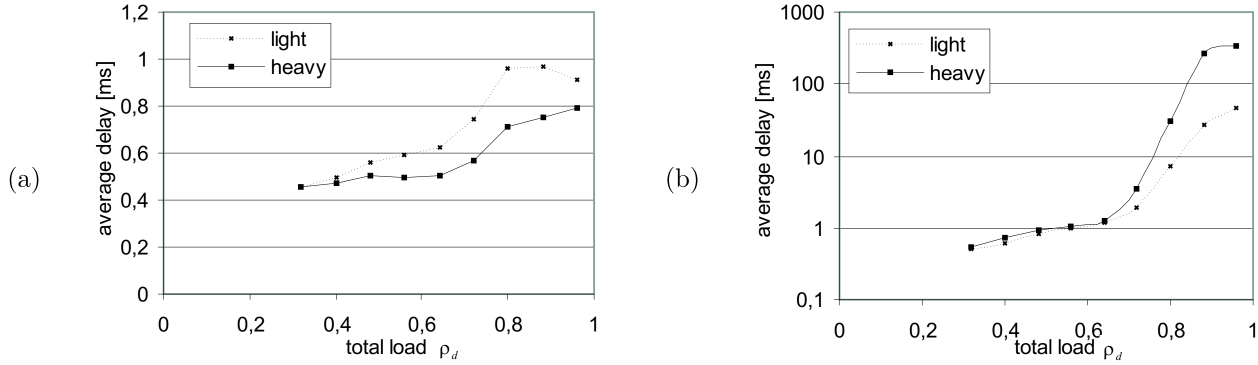


Figure 5. R-FPSA (a) the average queuing delay for priority 1 (b) the average queuing delay for priority 2 as a function of ρ_d.

two consecutive TWs is less compared to the light loaded ones. As the load increases in this area so does the difference between the average length of a TW assigned to a heavy loaded ONU and a light 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 and therefore the heavy loaded ONUs start to drop large amount of best effort traffic. Therefore, the throughput ratio for priority 2 traffic between heavy and light loaded ONUs, which equaled 3 at low loads, begins to decrease, for instance, at $\rho = 0.96$ the ratio equals 1.7. Hence, the ratio between the TWs allocated to heavy and light 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 Figure 4b) follows the behavior of the average delay.

The delay for priority 1 traffic has almost the same dynamic when varying the load, but there the difference between the delays for experienced by both types of ONUs is larger. Recall, the amount of traffic generated for the priority 1 traffic stays the same for all loads. When an ONU has a TW it starts to transmit the packets according to their priority and if the TW is large enough to support the transmission of all priority 0 and 1 traffic, it reports only the state of queue 2 at the end of the TW. Consequently for the next cycle it will be allocated the reported value plus some fair share. For low loads this reported value is close to zero and thus all ONUs get more or less the same TW. As ρ increases the reported value will grow and cause an unfairness between both types of ONUs. Indeed, light loaded ONUs will get significantly less bandwidth and therefore some of the newly arrived priority 1 traffic of the light loaded ONUs might be postponed for one cycle whereas the new priority 1 traffic of the heavy loaded ONUs can make use of the bandwidth reserved by the best effort traffic. As with priority 0 the delay of both types of ONUs at severe overload conditions converge to the same value.

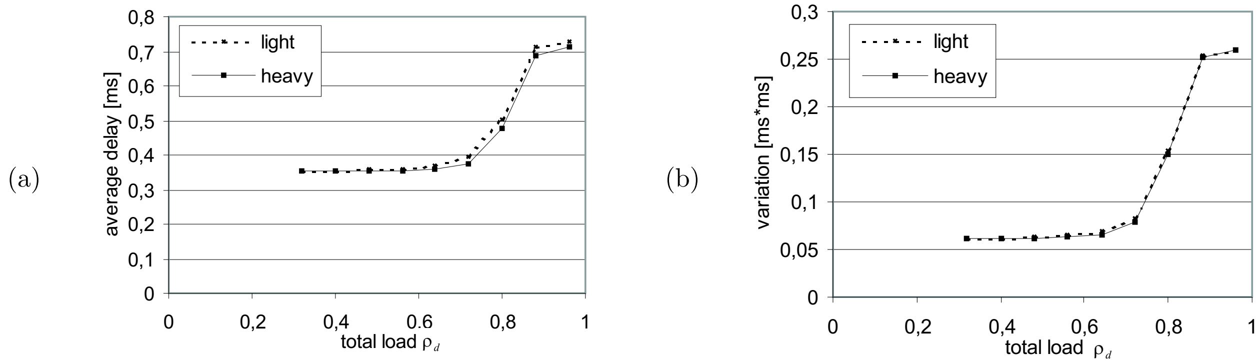


Figure 6. R-IPSA (a) the average queuing delay and (b) the delay variation of the priority 0 traffic as a function of ρ_d.

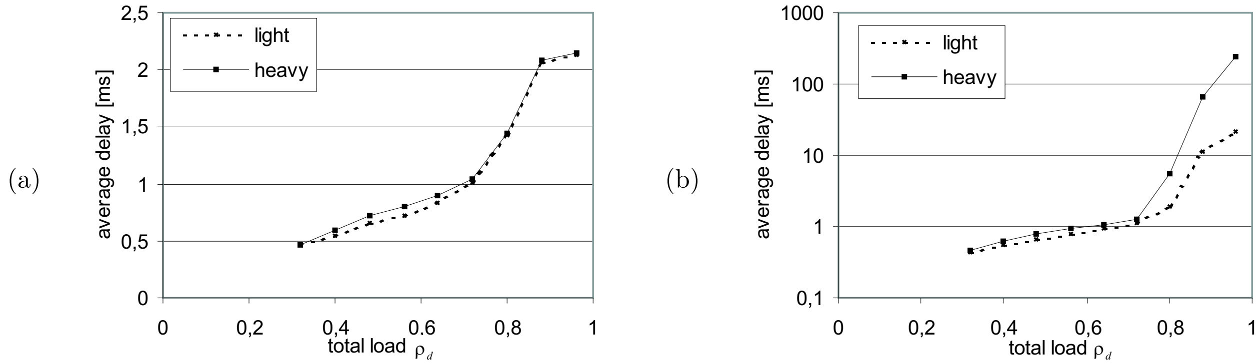


Figure 7. R-IPSA (a) the average queuing delay for priority 1 (b) the average queuing delay for priority 2 as a function of ρ_d.

The delay of priority 2 traffic for the light loaded ONUs is less than the one for the heavy loaded ONUs for all loads as seen from Figure 5. For loads $\rho \leq 0.64$ the difference is small. This is the region where the average cycle length has constant value, which means that the ratio of the allocated bandwidth corresponds to the one of the total amount of generated traffic. From this load up packets for priority 2 at the heavy loaded ONUs are being discarded. At loads $\rho > 0.8$ the delay for this priority for the heavy loaded ONUs becomes constant because of the finite size buffer at the ONUs.

The average packet delay for priority 0 when using R-IPSA is presented in Figure 6a. The delay for the two types of ONUs are very close to each other. The maximum difference reached at load $\rho_d = 0.88$ is 0.026 ms. Recall that with R-IPSA the packets from priority 0 are still transmitted before the reported lower priority traffic. The heavy loaded ONUs request more bandwidth and more is allocated to them. Consequently the odds are higher that a packet will arrive during TW and will be transmitted without delay, hence the average delay is smaller than for the light loaded ONUs. At overloaded conditions there is the effect of convergence of the delays as for R-FPSA. The packet delay variation for priority 0 of R-IPSA presented in Figure 6b and has the same behaviour as the average packet delay though the difference in the variation for the two types of ONUs is minimal. The maximum value of this difference is 0.004 ms^2.

The average packet delay for priority 1 is presented in Figure 7a. As the traffic for priority 1 for the two types of ONUs is the same we should expect the same delay. However, we have demonstrated that for the R-FPSA the ONUs with more low priority traffic are favoured, while for R-IPSA the ONUs with less low priority traffic have a better delay performance for this priority at loads $\rho < 0.6$. This is due to the fact that with R-IPSA the ONU first transmits the reported traffic and thus it may occur that traffic from priority 2 is transmitted before the one for priority 1. In this way it seems as if the ONUs with more low priority traffic have to pay some penalty.

The delay in this load region increases due to the fact that the load of priority 2 increases, which means that packets from priority 1 have to be more delayed.

With the augmentation of the total load more and more packets are being discarded from the priority 2 queue of the heavy loaded ONUs (as can be seen from the sharp increase of the average delay for this priority given in Figure 7b). This implies that the bandwidth allocated for the two types ONUs converges and so does the average delay for priority 1.

5. CONCLUSIONS

In this paper we compared the QoS support for packet delay and delay variation of two types of ONUs, which differ in the amount of best effort traffic, for two types of intra-priority ONU scheduling. For the two types of algorithms, we have simulated modified versions of the rate-based DBA algorithm at the OLT, namely, R-FPSA and R-IPSA. We demonstrated that the QoS support of high priority traffic is not influenced by the presence of best effort traffic when using R-IPSA, as opposed to R-FPSA that favors ONUs with lots of best effort traffic. While slightly higher delays for some priorities are observed with R-IPSA, it realizes better fairness and efficiency when compared to R-FPSA.

REFERENCES

1. "Media Access Control Parameters, Physical Layers and Management Parameters for subscriber access networks", IEEE Draft 802.3ah/D1.414, March 2003.
2. "Virtual Bridged Local Area Networks", IEEE 802.1q, 1998.
3. D. Nikolova, B. Van Houdt, and C. Blondia, "Dynamic bandwidth allocation algorithms in EPON: a simulation study", to appear in *Proc. of Opticomm'03*.
4. 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.
5. 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.