On the Use of the Doze Mode to Reduce Power Consumption in EPON Systems

Sergio Herrería-Alonso, Miguel Rodríguez-Pérez, Manuel Fernández-Veiga, Senior Member, IEEE, and Cándido López-García

Abstract—Current networking equipment usually provides several energy profiles with different performance capabilities. For example, in EPON systems, ONUs are able to switch their transmitters off when there is no data to transmit. This low power mode, known as doze mode, can significantly reduce power consumption in EPONs if wisely used. In this paper we evaluate through simulation the potential power savings that can be obtained using this mode in many different scenarios. In particular, we analyze the impact of the DBA algorithm and the mode governor on both energy efficiency and frame delay with different implementations of the doze mode.

Index Terms—Energy efficiency, EPON, doze mode.

I. INTRODUCTION

The importance of a more efficient usage of power resources in the IT field has been recently gaining mind share not only for improving environmental sustainability but also for reducing the high operating costs associated with networking equipment. In this paper we focus on reducing power consumption of Ethernet passive optical network (EPON) systems. In spite of their inherent energy-efficient operation, EPON systems will involve a significant part of the energy budget of IT equipment in the near future since they are being widely adopted to support broadband services. Moreover, although EPONs are quite power efficient compared to other access technologies when operating at 1 Gb/s [1], in the next generation they will likely be upgraded to 10 Gb/s capacity increasing their energy needs significantly [2].

Essentially, an EPON consists of one optical line terminal (OLT), located at the provider central office, connected via optical fibers and optical passive splitters to multiple optical network units (ONUs) located at the users’ premises. For downstream transmission, data are broadcasted from the OLT to all the connected ONUs, so each ONU must filter out those packets not directed to itself. In the upstream direction, a common upstream channel is shared among all the ONUs using time division multiple access (TDMA) and MPCP (Multi-Point Control Protocol) as the MAC algorithm to emulate a dedicated point-to-point channel from each ONU to the OLT. The OLT allocates the appropriate share of upstream bandwidth to each ONU with the help of a dynamic bandwidth allocation (DBA) algorithm that takes into account their different needs.

Without any doubt, the most important energy efficiency issue in EPON systems is the power consumption of ONUs since, although they are considerably more energy-efficient than OLTs [3], each OLT is usually connected to 16–32 ONUs, so ONUs outnumber OLTs significantly. In fact, it is estimated that up to 60% of the power required in an access provider network would be consumed in the ONUs [4].

The ITU-T has proposed two different power saving modes to reduce energy consumption in ONUs: the doze and the sleep mode [5], [6]. In the doze mode, the transmitter of the ONU is put to sleep when there is no upstream traffic while the receiver is kept awake to catch incoming downstream traffic. The sleep mode makes both the transmitter and the receiver sleep when no traffic is observed, so the ONU cannot detect the arrival of downstream traffic and cannot be accessed by the OLT for link synchronization.

Although the sleep mode consumes less energy than the doze mode, the latter can be considered the most promising to achieve significant energy savings when coupled with just a simple mode governor. Note that, if ONUs stop receiving downstream traffic, they lose clock synchronization and thus require a long clock recovery time when they wake up [7].

In addition, using the sleep mode requires some modifications to the MPCP protocol and the cooperation of the OLT since the downstream traffic for all the sleeping ONUs must be queued in this terminal until they wake up [8], [9], [10], [11], [12]. In any case, although the sleep mode switches both the transmitter and the receiver off, power savings in the transmitter are more important than in the receiver because it is the optical transmission circuitry that demands the larger part of the power requirements of these units [13].

In this paper we evaluate the potential power savings that can be obtained using the doze mode in many different scenarios. We analyze, for example, how the DBA algorithm used in the EPON affects energy efficiency comparing the power consumed in ONUs when different DBA algorithms are applied. In addition, the doze mode can be implemented in many different ways over diverse optical transmitters. When implementing a doze mode, there always exists a tradeoff between the power consumed in it and the time required to power the transmitter up. In this paper we evaluate two ends of the spectrum of available doze modes, that is, a relatively high energy demanding doze mode requiring a very short transition time to reactivate the transmitter, and a low-power doze mode with a quite larger transition time.

Another factor that significantly influences energy efficiency is the control policy applied to manage the doze mode. Probably, the most natural way to govern this mode is entering it immediately after sending the last buffered frame and restoring normal operation as soon as a new frame is ready for transmission. Unfortunately, this policy is not very efficient for high loads or non bursty traffic, so the waking up of the
dozed ONUs can be delayed until a given number of frames are queued in their upstream buffers. The most appropriate value for the queue threshold varies with traffic load, so in this paper we present a new dynamic algorithm able to adapt this parameter to real time traffic characteristics with the goal of maintaining average frame delay around a given target value while keeping energy consumption low enough.

The rest of this paper is organized as follows. Section II presents the fundamentals of EPON systems. Then, the main control policies for managing the doze mode and our new dynamic algorithm are described in Section III. Section IV shows some performance results obtained through simulation. Finally, the main conclusions are laid out in Section V.

II. EPON OPERATIONS

A. The DBA Cycle

Under MPCP the upstream channel is divided in periods that ONUs employ to transmit their traffic to the OLT. These periods correspond with DBA cycles. Every DBA cycle each ONU reports to the OLT its current bandwidth requirements by sending a traffic report (TR). In these reports, each ONU notifies the OLT of the amount of data stored in its upstream queue. Then, based on the TRs received, the OLT assigns non-overlapping transmission slots to the ONUs using a DBA algorithm and sends a gate message (GM) to each of them informing of the start time and length of their allocated transmission slots in the next DBA cycle. Clearly, the ONUs must only send upstream data frames in their corresponding transmission slots to avoid multiple uplink transmissions simultaneously. In addition, an appropriate guard time is intercalated between the transmission slots of each pair of ONUs in order to reduce the risk of collision due to potential time synchronization problems between the OLT and the ONUs.

B. The DBA Algorithm

Many different DBA algorithms have been proposed to allocate upstream bandwidth to ONUs [14], [15], [16], [17]. In this study, we focus on the most popular low-complexity DBA algorithms, so a number of assumptions have been made. Firstly, we only consider single-thread DBA algorithms, i.e., DBA algorithms in which each ONU is only polled once per DBA cycle. Secondly, we assume that the DBA algorithm execution is triggered when the TRs from all the ONUs have been received at the OLT. Finally, we also consider that the DBA algorithm schedules a transmission slot for every connected ONU in the EPON.

One of the main tasks of the DBA algorithm is to determine the length of transmission slots for each ONU in the next DBA cycle. The following are among the most interesting sizing policies proposed in the literature:

- Fixed allocation: with this policy, all the transmission slots have the same length irrespective of ONU requirements. This simple scheme is clearly inefficient in terms of performance since some ONUs receive too much bandwidth, while others receive too little.
- Gated IPACT (Interleaved Polling with Adaptive Cycle Time): when using this policy, the OLT allocates to each ONU a transmission slot with the exact length required to send all the data stored in its upstream queue [14].
- Limited IPACT: in order to upper bound the duration of a DBA cycle, this policy specifies a predefined maximum length for the transmission slots. So the transmission slots of those overloaded ONUs requesting excessively large slots are initially constricted to this limit. However, the excess upstream bandwidth accumulated from the under-loaded ONUs can be fairly shared among the overloaded ONUs [15].

The DBA algorithm also determines how the multiple transmission slots are ordered within the DBA cycle. For example, transmission slots can be ordered in ascending order by their lengths (shortest transmission slot first). Another possibility is that the ONUs are arranged in descending order by their upstream queue sizes (largest upstream queue first).

III. USING THE DOZE MODE

The ITU-T has specified a protocol to support the doze mode [5], [6], but the policy to decide when to enter and exit this mode falls out of its scope. The design of the control policy is left for implementers and network operators. Probably, the most natural way to govern the doze mode is entering it when there is no traffic to transmit and restoring normal operation when new upstream traffic arrives. Clearly, for maximizing energy savings, ONU transmitters should only be active when there is some data ready for transmission, so ONUs should enter the doze mode every time the upstream queue gets empty. When an ONU enters this mode, its transmission circuitry is powered down and thus, its energy demands are minimal, though, obviously, no transmission can be carried out and any upstream traffic must be buffered.

Going back to the active mode from the doze mode can be immediately triggered by the presence of new upstream traffic. Unfortunately, this policy is not very efficient for high loads or non bursty traffic. While the time needed to power down the transmission circuitry is negligible, the time needed to power it up ($T_w$) cannot be neglected and ONUs consume energy while doing this.\footnote{We assume that ONUs consume about the same power during transitions as in the active mode.}

In addition, ONUs are not yet ready to transmit data immediately after the doze mode is left. Recall that, before being allowed to transmit its queued data, the ONU must first send a TR to the OLT to be allocated a transmission slot in a future DBA cycle. However, as shown in Fig. 1(a), the ONU may have to wait to send its TR in the next DBA cycle if its assigned transmission slot in current DBA cycle has already gone. On the other hand, as suggested in [18], if the duration of the DBA cycle is fixed (for example, with the fixed DBA algorithm), then some extra energy can be saved simply keeping the transmission circuitry off until the proper time to switch it on arrives, that is, until $T_w$ seconds before the start of the next DBA cycle, in which the ONU will be able to transmit its TR (Fig. 1(b)).

In short, mode transitions can be quite costly and waking up dozed ONUs as soon as a new frame is ready for transmission
may cause them to spend great amounts of energy on switching between modes rather than on useful data transmission. Hence, it seems wise to delay the exit from the doze mode until the upstream queue reaches a certain threshold \( Q_w \) thus reducing the amount of mode transitions. Obviously, this policy will provide greater energy savings at the expense of increasing frame delay, but, in any case, the maximum frame delay is upper bounded since the maximum time an ONU can remain continuously in the doze mode must be limited to 50 ms to avoid disconnection from the OLT.\(^2\)

### A. Dynamic Queue Threshold

Obviously, a good tuning of the queue threshold is key for the performance of this mechanism. If this threshold is too high, frames can be excessively delayed. On the contrary, setting a lower queue threshold results in less power savings. Additionally, as shown later, a single threshold value does not suit well for any possible upstream traffic load.

In [19] we presented an interesting solution to keep energy consumption upper bounded simply adapting the queue threshold to current traffic conditions. Here, we adjust this method to minimize power consumption in ONUs while trying to maintain the average frame delay around a given target value. We assume that, although traffic characteristics do change over time, they usually do so over periods of several DBA cycles. So, to adjust the \( Q_w \) parameter to the existent traffic conditions, we measure the average delay experienced by frames in the last DBA cycle \( D[i] \) and compare it with our target delay \( D^* \). Then, if \( D[i] > D^* \), \( Q_w \) should be reduced to diminish frame delay. Conversely, if \( D[i] \leq D^* \), current average frame delay is low enough and \( Q_w \) can be incremented to reduce power consumption. Assuming that upstream traffic characteristics in DBA cycle \( i+1 \) remain similar to those of DBA cycle \( i \), we propose to modify \( Q_w \) by just a small amount \( \gamma \) as shown in Table I.

### B. Doze Mode Parameters

Both the actual time required to exit the doze mode and restore normal operation \( T_w \) and the power consumption

\[ \text{TABLE I} \]

**Dynamic Tuning Algorithm of \( Q_w \).**

1. At the end of DBA cycle \( i \), just before entering the doze mode, measure \( D[i] \).
2. Update \( Q_w \) as shown:

\[
Q_w[i + 1] \leftarrow \begin{cases} 
Q_w[i] + \gamma & \text{if } D[i] \leq D^* \\
\max\{Q_w[i] - \gamma, 1\} & \text{if } D[i] > D^*
\end{cases}
\]

\[ \text{TABLE II} \]

**Simulation Parameters.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of ONUs</td>
<td>32</td>
</tr>
<tr>
<td>Guard time</td>
<td>1 ( \mu s )</td>
</tr>
<tr>
<td>TR size</td>
<td>64 bytes</td>
</tr>
<tr>
<td>Frame size</td>
<td>1500 bytes</td>
</tr>
<tr>
<td>Arrivals distribution</td>
<td>Pareto (( \alpha = 2.5 ))</td>
</tr>
<tr>
<td>Upstream queue capacity</td>
<td>( \infty )</td>
</tr>
<tr>
<td>( Q_w )</td>
<td>{1, 10, 100}</td>
</tr>
<tr>
<td>Dynamic ( Q_w ) tuning</td>
<td>( D^* = 5 ) ms, ( \gamma = 1 ) frame</td>
</tr>
<tr>
<td>&quot;Laggy&quot; doze mode</td>
<td>( T_w = 2 \ms, W = 30% )</td>
</tr>
<tr>
<td>&quot;Swift&quot; doze mode</td>
<td>( T_w = 1 \mu s, W = 60% )</td>
</tr>
</tbody>
</table>

\(^2\)Under MPCP, if the OLT does not receive a TR from an ONU for more than 50 ms, the OLT considers that the ONU has been disconnected.
Fig. 2. Energy consumption in ONUs using a “laggy” doze mode.

Each simulation was run for 10 seconds and repeated with different random seeds. For each simulation experiment we measured both the average energy consumed in the ONUs and the average frame delay.³

A. Results with a “Laggy” Doze Mode

In this scenario we consider that the doze mode just consumes 30% of the power demanded in the active mode.

³The 95% confidence intervals are not represented in the graphs since they are consistently lower than ±1% and just clutter the figures.
are ordered using the policy “shortest transmission slot first”. As shown in Fig. 2 and 3, even with a small queue threshold of just 1 frame, the energy savings obtained when the traffic load is low are quite significant although increasing the queue threshold provides greater energy savings at medium and high loads at the expense of increasing frame delay. Note, however, that the average frame delay is upper bounded since, as explained in Section III, the maximum time an ONU can be continuously in the doze mode is limited. Alternatively, our dynamic $Q_w$ tuning algorithm achieves its goal of maintaining the average frame delay around the target value for any load without increasing power consumption more than necessary.

Regarding the impact of the DBA algorithm, if the queue threshold is set to 1 frame, less energy is consumed with the fixed DBA algorithm than with the IPACT algorithms. On contrary, for higher queue thresholds, the situation is reversed and less energy is consumed when using the IPACT algorithms. The obtained results are a consequence of the relatively large DBA cycles introduced by the fixed DBA algorithm. In fact, this algorithm is carrying out an implicit frame coalescing because a large period of time is comprised since the upstream queue reaches the threshold until the actual beginning of frame transmission. On the other hand, few differences are observed between the gated and the limited IPACT algorithms. It is only worth noting that, at high loads with high queue thresholds, gated IPACT just demands a little less energy than limited IPACT since the former is able to assign transmission slots that can accommodate all the data stored in the upstream queues of the ONUs and, therefore, reduces the number of mode transitions.

**B. Results with a “Swift” Doze Mode**

In the following experiments we consider a doze mode that demands up to 60% of the power required in the active mode. This more energy demanding doze mode, however, is able to restore normal operation in just 1 µs. The energy consumption and the average frame delay measured when using such a “swift” doze mode are shown in Fig. 4 and 5 respectively. These results are similar to those obtained using the “laggy” doze mode with some minor differences.

Again, increasing the queue threshold implies greater energy savings at the expense of increasing delay but, since power consumption cannot be less than 60% with this “swift” mode and this limit is easily reached at low loads with a small queue threshold, the impact of this threshold on power consumption is now less important than in the previous scenario.

Also note that, even with a low queue threshold, IPACT algorithms consume less energy than the fixed DBA algorithm under low and medium load conditions. That is because the “swift” doze mode involves considerably shorter transition times than the “laggy” doze mode, so the impact of transitions on power consumption is greatly reduced and only when the traffic load is high and a huge number of transitions are induced the fixed DBA algorithm consumes less energy.

![Fixed DBA algorithm](attachment:fig4a.png)

![Gated IPACT algorithm](attachment:fig4b.png)

![Limited IPACT algorithm](attachment:fig4c.png)

**Fig. 4.** Energy consumption in ONUs using a “swift” doze mode.

Finally, we wish to highlight that our dynamic $Q_w$ tuning algorithm still keeps under control the average frame delay just causing a very slight increment in power consumption. Figure 6 shows how our algorithm dynamically accommodates the queue threshold to the incoming traffic characteristics. As traffic load increases (and hence frame interarrival times decrease), the algorithm chooses higher queue thresholds. These higher thresholds enable greater power savings without sacrificing frame delay, as the time required to reach them is lower when traffic load is high.

---

4 Results obtained using the policy “largest upstream queue first” are not included in the paper since they are quite similar to those shown using the policy “shortest transmission slot first”.

---

---
C. “Laggy” vs. “Swift” Doze Mode

In order to make easier the comparison between the “laggy” and the “swift” doze mode, we summarize in Fig. 7 some of the most representative results obtained in our study. This figure shows the power consumption and the average frame delay measured under low, medium and high traffic load conditions. For the low load scenario, it is assumed a whole traffic load in the EPON of just 5% of its upstream channel capacity, with the load equally distributed among all the ONUs. Percentages of 25% and 50% are assumed for the medium and high load scenarios respectively.

As can be seen in Fig. 7, if the doze mode is left as soon as a new frame is ready for transmission \((Q_w = 1)\), using the “swift” doze mode is undoubtedly the best option since this mode achieves greater energy savings with lower delays at the same time. However, the situation reverses as the queue threshold increases. With higher queue thresholds, ONUs spend most of the time in the doze mode, so, under these circumstances, achievable energy savings are basically limited by the power consumption of the doze mode. Consequently, the “laggy” doze mode consumes far less energy making it an excellent choice in spite of a slight increment in frame delay.

Finally, it is worth noting that the “laggy” doze mode is also preferred when our dynamic \(Q_w\) tuning algorithm is applied since this doze mode demands less energy than the “swift” doze mode without affecting the performance of our algorithm.

V. CONCLUSIONS

In this paper we have evaluated the potential energy savings that can be obtained in EPON systems when ONUs can make use of a doze mode to reduce their power consumption. We have analyzed the impact of the DBA algorithm and the control policy on both energy efficiency and average frame delay with two different implementations of the doze mode: a relatively high energy demanding doze mode that requires a
very short transition time to restore normal operation (“swift” doze mode), and a low-power doze mode with a quite larger transition time (“laggy” doze mode).

The main conclusions of our study are summarized below:

- If frame coalescing cannot be applied and the doze mode must be abandoned as soon as a new frame is ready for transmission, then we recommend to use in the ONUs a “swift” doze mode instead of a “laggy” doze mode to achieve greater energy savings with lower delays at the same time.
- Frame coalescing provides greater energy savings, especially at medium and high traffic loads, at the expense of increasing frame delay. If frame coalescing is applied, then a “laggy” doze mode is preferred.
- A good tuning of the queue threshold is key for the performance of the coalescing policy. Essentially, the higher the queue threshold, the greater energy savings, although frames can get excessively delayed. In addition, a single threshold value does not suit well for any possible upstream traffic load, so it is highly recommended to apply some algorithm, as the one proposed in this paper, to dynamically adjust this parameter in accordance with existing traffic conditions thus sparing the administrator from manually tuning it.

ACKNOWLEDGMENTS

This work was supported by the “Ministerio de Educació y Ciencia” through the project TEC2009-12135 within the “Plan Nacional de I+D+I” (partly financed with FEDER funds).

REFERENCES