

Energy Efficiency in TDMA-based Next-Generation Passive Optical Access Networks

Ahmad R. Dhaini, Pin-Han Ho, Gangxiang Shen, and Basem Shihada

Abstract—Next-generation passive optical network (PON) has been considered in the past few years as a cost-effective broadband access technology. With the ever-increasing power saving concern, energy efficiency has been an important issue in its operations. In this paper, we propose a novel sleep time sizing and scheduling framework for the implementation of green bandwidth allocation (GBA) in TDMA-PONs. The proposed framework leverages the batch-mode transmission feature of GBA to minimize the overhead due to frequent ONU on-off transitions. The optimal sleeping time sequence of each ONU is determined in every cycle without violating the maximum delay requirement. With multiple ONUs possibly accessing the shared media simultaneously, a collision may occur. To address this problem, we propose a new sleep time sizing mechanism, namely Sort-And-Shift (SAS), in which the ONUs are sorted according to their expected transmission start times, and their sleep times are shifted to resolve any possible collision while ensuring maximum energy saving. Results show the effectiveness of the proposed framework and highlight the merits of our solutions.

Index Terms—Passive Optical Network (PON), Green Communications, Energy Efficiency, quality-of-service (QoS), Dynamic Bandwidth Allocation, Analytical Model, Queuing Theory.

I. INTRODUCTION

Passive Optical Network (PON) has been considered as an attractive solution to the broadband wired access network bottleneck by which the final frontier of optical FTTH networks is formed [1]. It comprises an Optical Line Terminal (OLT) that resides in a Central Office (CO) and connects a set of associated Optical Network Units (ONUs) through a single fiber. Time division multiple access (TDMA)-based PONs, such as Ethernet PON (EPON) and Gigabit PON (GPON), have been standardized, and they employ one wavelength in the upstream (i.e., from ONUs to OLT) and another wavelength in the downstream (i.e., from OLT to ONUs).

Although PON consumes the least power among all the reported access technologies [2], it still gets a good attention in the global energy saving plan of modern telecommunications networks which requires reducing global Green House Gas (GHG) emissions at large quantities [3]–[6]. One of the most

promising and viable solutions to save energy in PON is to support sleep mode functionality at the ONUs. This is expected to decrease the ONU energy consumption by almost 80% [4]–[8]. *Fast/Cyclic sleep* has been the most attractive ONU sleep mode. Under this mode, the ONU state alternates between the sleep state (i.e., when the transceiver is completely turned off such that the downstream and upstream traffics are buffered at the OLT and ONUs, respectively) and the active state (i.e., when the transceiver is turned on). The network performance with this method is mainly dependent on the dynamic bandwidth allocation (DBA) employed in the active period.

The active period and the subsequent sleep period form the ONU *sleep cycle*, which also comprises the overhead time required to switch the ONU from the *sleep* state to the *active* state. The overhead time mainly consists of the ONU clock recovery time and the OLT clock synchronization time¹, and varies between 0.125 *ms* and 5.125 *ms* depending on the deployed ONU architecture [7]. A low 0.125 *ms* overhead is considered acceptable and can only be achieved by manufacturing new, yet costly, ONUs. These do also not save the maximum possible energy [7]. Conversely, legacy ONUs can go into the sleep mode with a simple and straightforward sleep functionality support. However, the latter causes a high overhead with a conservative estimate equal to 5.125 *ms* and an aggressive estimate equal to 2.125 *ms* [6], [7].

Determining the ONU sleep time is a major design challenge in realizing green PONs. The allocation of the ONU sleep time can either be static or dynamic. For static sleep time allocation, the authors of [9] proposed a new medium access control (MAC) protocol which allows an ONU to wake up periodically for pre-defined periods of time to send and receive data. The authors of [10] proposed two MAC protocols to save energy in EPONs using a fixed bandwidth allocation (FBA) mechanism. The first scheme, upstream centric scheme (UCS), maximizes the ONU sleep time by having the OLT to launch downstream traffic for the ONU during its allocated upstream time slot. In the second scheme, namely downstream centric scheme (DCS), an ONU goes into the sleep mode whenever both the upstream and downstream traffics are absent. Clearly, static sleep time allocation can neither guarantee the network performance nor achieve maximum energy saving.

For dynamic sleep time allocation, the authors of [11] proposed two sleep mode scenarios. In the first scenario, an ONU remains in the sleep mode for more than one DBA

Manuscript received xxxx, xxxx; revised xxxx, xxxx.

A. R. Dhaini is with the Department of Electrical Engineering, Stanford University, CA, 94305, USA. (e-mail: adhaini@stanford.edu).

P.-H. Ho is with the Department of Electrical and Computer Engineering, University of Waterloo, ON, N2L 3G1, Canada, and with the School of Systems Information Science, Sooshow University, China (e-mail: p4ho@uwaterloo.ca).

G. Shen is with the School of Systems Information Science, Sooshow University, China (e-mail: shengx@suda.edu.cn).

B. Shihada is with the Department of Computer Science, KAUST, Thuwal, 23955-6900, Saudi Arabia (e-mail: basem.shihada@kaust.edu.sa).

¹The ONU clock recovery time is the time spent by the ONU to recover the OLT clock; whereas the OLT synchronization time is the time spent by the ONU to retrieve network synchronization from the OLT.

cycle (with a typical length of 2 *ms*) based on a predefined threshold in which the ONU is allowed to stay idle. In the second scenario, an ONU goes into the sleep mode within the DBA cycle based on a threshold that is estimated through the downstream traffic profiles of all the ONUs. Without addressing any performance issue, the proposed scenarios leave room for further investigation in terms of quality-of-service (QoS) performance.

In [12], a new green bandwidth allocation (GBA) framework is proposed, which leverages batch-mode transmissions in the upstream and downstream channels to save more energy with the same amount of sleep time. Different from other schemes in the literature, GBA allows the ONU to go into the sleep mode even if upstream and/or downstream packets are present. The idea is to let the ONU be in the sleep mode for a period of time determined based on the maximum delay requirement rather than the availability of upstream and/or downstream traffics. Another unique feature of GBA is that it does not require the installation of new types of ONUs, as it is designed to exploit the sleep mechanism of legacy ONUs so as to ensure maximum energy conservation in PONs at the lowest cost. Nevertheless, GBA can also be operated over any ONU architecture envisioned to be deployed in the future. Conversely, GBA does not specify any ONU sleep time computation mechanism. Instead, it offers a bandwidth and sleep time allocation platform that can incorporate any sleep time sizing scheme. Therefore in this paper, we propose a novel sleep time sizing and scheduling framework to operate and exploit the proposed GBA. The framework comprises a new simple yet effective analytical model that provides a closed-form expression of the ONU sleep time based on the maximum delay requirements. Each ONU is modeled as an M/G/1 system with “vacations”, such that the sleep and overhead times are taken as vacation periods.

While M/G/1 with vacations has been thoroughly explored [13], [14], and lately applied to single-class TDMA-PON [15]–[17], to the best of our knowledge, this is the first closed-form sleep time model for multi-class PONs using M/G/1 queue with vacations.

With the assigned sleep times, a collision may occur if multiple ONUs are scheduled to start transmission directly upon waking up from the sleep mode without synchronizing their sleep times. To resolve this problem, the proposed framework includes a new sleep time sizing and scheduling scheme, so-called Sort-And-Shift (SAS), which sorts and shifts the sleep time of the collided ONUs until the collision is eliminated. SAS ensures that the ONU is fully granted with its allocated sleep time, thereby ensuring maximum energy saving.

Our results prove the effectiveness of our solutions and highlight their advantages. They also show that the saving is not much affected by the increased traffic loads as generally expected.

The rest of the paper is organized as follows. In Section II, we present the system model. In Section III, we present the new analytical model to compute the maximum ONU sleep time. Section IV discusses the sleep time sizing problem and presents SAS. Simulation results are given in Section V. In Section VI, we conclude the paper and discuss future work.

TABLE I: Most relevant notations used in the paper.

Notation	Description
\mathbf{N}	The set of all $ \mathbf{N} $ ONUs
$v_{s,i}$	The OLT serving/vacation time for ONU i
v_c	Control vacation time
$\lambda_{c,i}$	Data arrival rate for CoS c of ONU i
C_{pon}	PON speed
C_{light}	Speed of light
$\mathbb{D}_{c,i}$	One-way delay requirement for CoS c of ONU i
T_i^o	Total overhead time of ONU i
$T_i^{o,1}$	Full/Cyclic overhead time of ONU i
$T_i^{o,2}$	Doze overhead time of ONU i
T_i^s	Total sleep time of ONU i
$T_i^{s,1}$	OLT-based sleep time of ONU i
$T_i^{s,2}$	ONU-based sleep time of ONU i
T_{gba}	GBA computation time
T_i^a	Active time of ONU i
T_g^a	Guard time in active period between two ONUs
T_g^{gr}	Guard time between two grants
T_{gr}^{proc}	Grant processing time
T_{gr}^{tran}	Grant transmission time
T_{br}^{proc}	Bandwidth request processing time
T_{br}^{tran}	Bandwidth request transmission time
T_i^{prop}	Propagation time between OLT and ONU i
d_i	Distance between OLT and ONU i
h_i	Shifted sleep time of ONU i

II. SYSTEM MODEL

In this section we define the system model which consists of the power model, the ONU architecture, the proposed GBA [12], and the condition in which the ONU is allowed to go into the sleep mode. The most relevant notations used in the paper are summarized in Table I, and the rest are defined in the text.

A. Power Model

Let E_{olt} and E_i denote the energy consumed by the OLT and ONU i , respectively. The total energy consumed by a single PON system, E_{pon} can be formulated as follows:

$$E_{pon} = E_{olt} + \sum_{i \in \mathbf{N}} E_i. \quad (1)$$

In current PON systems, it is argued that the OLT cannot go into the sleep mode and its power consumption is fixed to 20 W [18]. This is due to the fact that the OLT plays the role of distributor, arbitrator, and aggregator of the network and is responsible for controlling and initiating the energy saving mechanism at the ONUs [10]. In addition, putting the OLT into sleep might cause a synchronization problem for the ONUs.

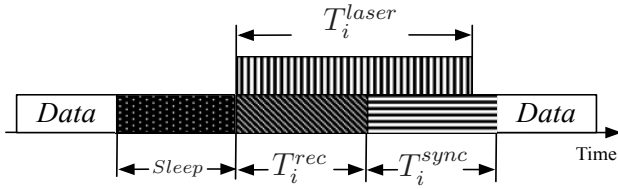


Fig. 1: Illustration of the overhead time composition.

It is also argued that the statistical multiplexing nature of the upstream channel and the bursty traffic from ONUs forbids us from turning off the OLT's receiver as it will threaten the QoS performance in the network [8]. Hence in our model, we only focus on saving energy in TDMA-PON by putting the ONU into the sleep mode.

The total energy consumed by a "green" ONU is given by:

$$E_i = E_i^a + E_i^d + E_i^s + E_i^o, \quad (2)$$

where E_i^a , E_i^d , E_i^s , and E_i^o denote the energy consumed by ONU i in the *active* state, the *doze* state, the *sleep* state, and the *overhead* period, respectively.

In the active state, the energy consumed by ONU i is estimated as follows:

$$\begin{aligned} E_i^a &= P_i^a \times (T_i^a + T_i^{idle}) \\ &= P_i^a \times [(T_i^{tran} + T_i^{recv} - T_i^{overlap}) + T_i^{idle}], \end{aligned} \quad (3)$$

where P_i^a , T_i^{tran} , T_i^{recv} , and T_i^{idle} are the power consumption in the active state, the data transmission period, the data receiving period, and the idle period, respectively. The time $T_i^{overlap}$ accounts for the case where the ONU's receiving and transmitting times are overlapped (e.g., the UCS protocol).

In the doze mode, the ONU's transmitter is turned off while the receiver is kept on [5]. Thus, the energy consumed by ONU i in this mode is computed as follows:

$$E_i^d = P_i^d \times T_i^{recv}, \quad (4)$$

where P_i^d is the power consumed in the doze state.

The total energy consumed by ONU i during the sleep state is given by:

$$E_i^s = P_i^s \times T_i^s, \quad (5)$$

where P_i^s and T_i^s are the power consumed in the sleep state, and the ONU sleep time, respectively.

Note that the ONU sleep time T_i^s and the ONU active time T_i^a are strictly evaluated based on the sleep time control mechanism and the DBA scheme employed in the active period. Conversely, the overhead time spent by the ONU to transition from the sleep state to the active state consists of a period T_i^{laser} denoting the time to have the laser of the transceiver ready, the OLT clock synchronization period T_i^{sync} , and the ONU clock recovery period T_i^{rec} . As shown in Fig. 1, upon waking up from a UCS-based sleep period, the ONU turns its laser on² and simultaneously starts recovering the OLT clock before synchronizing with it. Hence, the overhead time

is formulated as follows:

$$T_i^o = \max(T_i^{rec} + T_i^{sync}, T_i^{laser}). \quad (6)$$

T_i^o occurs every time the ONU wakes up to send or receive data. If the downstream and upstream transmissions do not overlap, the period T_i^{laser} will be consumed in the upstream, and the period $T_i^{rec} + T_i^{sync}$ will be consumed in the downstream. Nevertheless, overlapping the downstream and upstream transmissions can save maximum energy by enabling more ONU sleep time, at the expense of making the required QoS performance for downstream and upstream traffic harder to achieve.

Assuming that turning the laser diode of the transceiver on also consumes P_i^a , the total overhead energy consumed by ONU i is given by³:

$$E_i^o = P_i^a \times T_i^o. \quad (7)$$

TABLE II: Green ONUs [12].

Characteristic	Green ONU (GR-ONU) Architecture			
	1/C	1/A	2	3
Active power: P_i^a (W)	3.85	3.85	3.85	3.85
Sleep power: P_i^s (W)	0.75	0.75	1.08	1.28
Full overhead: $T_i^{o,1}$ (ms)	5.125	2.125	0.125	
Doze power: P_i^d (W)	1.7			
Doze overhead: $T_i^{o,2}$ (ms)	0.125			

B. ONU Architecture

The ONU architecture plays an important role in the design and selection of the DBA mechanism. The DBA schemes which require the ONUs to sleep within one DBA cycle are not suitable for legacy ONUs due to their high overhead times. Consequently, new ONU architectures with low overhead times have been proposed [7]. Table II exhibits the ONUs envisioned to be operated under green PONs (legacy and new). We briefly describe these architectures as follows⁴:

- **Green ONU-1 (GR-ONU-1):** It is basically the legacy ONU with enabled sleep mode. The advantages of GR-ONU-1 is that it requires no new ONU manufacturing, meanwhile it can save maximum energy by turning off most of its hardware when going into the sleep mode. The overhead time of GR-ONU-1 can either be estimated in a conservative manner (hence, GR-ONU-1/C) or in an aggressive manner (hence, GR-ONU-1/A) depending on the clock synchronization protocol that is employed between the OLT and ONUs [6].
- **Green ONU-2 (GR-ONU-2):** This new ONU keeps some of its hardware on so that a very short overhead time is caused at the expense of slightly higher energy consumption while in the sleep mode.
- **Green ONU-3 (GR-ONU-3):** This new architecture exhibits the same overhead time as GR-ONU-2 by shutting

³The energy consumed by a device upon turning it on is usually higher than when it is functioning at normal operation.

⁴For a detailed overview, we refer the reader to [7], [12].

²A study on the laser on time can be found in [18].

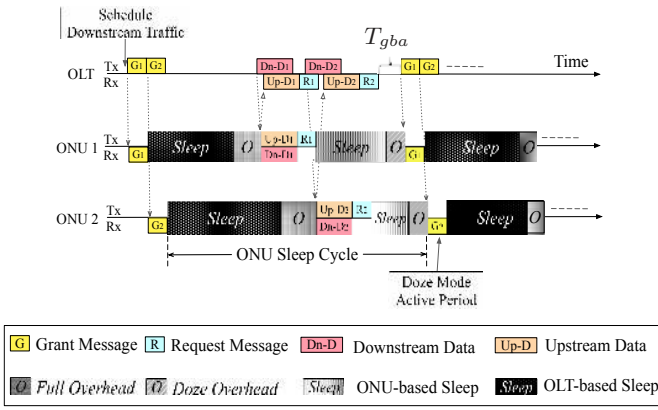


Fig. 2: Illustration of the proposed GBA [12].

down the minimum hardware required to put the ONU into sleep. However as noticed, the energy consumption in the sleep mode will be higher.

GR-ONU-2 can be seen as the most favourable ONU due its very short overhead time and slightly higher energy consumption. Nevertheless, design and installation costs might decelerate its deployment. On the other hand, although subject to longer overhead time and hence possibly impaired QoS level in the power saving mode, GR-ONU-1 consumes the least power in the sleep mode and has been widely deployed as the mainstream of the PON industry. Thus, with intelligent manipulation of GR-ONU-1's features, a sleep allocation mechanism can save maximum energy at the lowest cost.

The *full* (or *cyclic*) overhead time is the time required by the ONU to switch from the sleep mode to the active mode; whereas the *doze* overhead time represents the time needed by the ONU to switch from the sleep mode to the *doze* mode.

C. Green Bandwidth Allocation

The GBA framework [12] mainly enables batch-mode transmissions in the upstream and downstream directions so as to maximize the ONU sleep time and thereby minimize the wasted overhead due to frequent ONU transitions from the sleep state to the active state. It also employs the UCS mechanism in order to save maximum energy. However, unlike [10], GBA employs UCS under a dynamic bandwidth allocation scheme rather than the fixed/static one, which does not achieve good network performance [19]. Consequently, GR-ONU-1's features are exploited and the impact of its long overhead time is reduced via efficient sleep time scheduling. Finally, it defines a platform for the computation/allocation of ONU sleep time, in which the ONUs are polled by the OLT in a quasi-leaved round-robin fashion. As illustrated in Fig. 2, the OLT waits until all the request messages are received from all ONUs before running GBA. Based on the scheduling discipline (e.g., UCS), the OLT computes the sleep time for every ONU and then broadcasts the grant messages to all the ONUs, each specifying the ONU's sleep time and its assigned time slot.

GBA's quasi-leaved polling enables effective *offline* scheduling of the ONUs' sleep times so as to resolve any possible collision caused by multiple ONUs waking up and trying to

access the shared media simultaneously (e.g., using SAS). This can be very complex to attain using *online* schedulers (e.g., Interleaved Polling with Adaptive Cycle Time, IPACT [19]), which do not enable manipulation of already-scheduled ONU sleep times and grants. The *offline* time spent by the OLT to serve ONU i is given as follows⁵:

$$v_{s,i} = T_{br}^{proc} + 2T_i^{prop} + T_{gba} + i \times T_{gr}^{proc} + (i-1)T_g^{gr} + iT_{gr}^{tran}. \quad (8)$$

In other words, $v_{s,i}$ is the time spent by the OLT directly after the last polled ONU sends its request message to the OLT, and until ONU i receives its designated grant message. Specifically, the OLT requires T_{br}^{proc} to process the last ONU bandwidth request (or REPORT) message, since the other requests were processed while the last ONU was transmitting. This last bandwidth request message requires T_i^{prop} to reach the OLT, and the broadcasted grant messages require another T_i^{prop} . The time to compute/run GBA is T_{gba} , and to transmit the grant messages is iT_{gr}^{tran} . Every two grant messages are separated by one grant guard time T_g^{gr} , and hence $(i-1)T_g^{gr}$ in total are exhibited in every downstream cycle.

Under GBA, the ONU will go into the sleep mode at most twice in every GBA cycle. The first time is upon receiving its designated grant message and until it's time to start transmitting/receiving packets; and the second time is directly after sending the request message and until the end of its *sleep cycle* (i.e., until receiving the grant message). Therefore in each grant the OLT specifies two sleep time fields:

- 1) **ONU-based sleep** ($T_i^{s,2}$): which is the time that the ONU spends in the sleep mode directly after sending a request message and while other ONUs are active.
- 2) **OLT-based sleep** ($T_i^{s,1}$): which represents the time that the ONU spends in the sleep mode directly upon receiving the grant message, such that:

$$T_i^{s,1} = T_i^s - T_i^{s,2}. \quad (9)$$

In both the OLT-based sleep and ONU-based sleep periods, the ONU will consume the same power level P_i^s .

As noticed in Fig. 2, unlike in legacy DBAs, the ONU sleep cycle in GBA exhibits no wasted idle time usually caused by the exchange of control messages between the OLT and ONUs. This is achieved by putting the ONU into the ONU-based sleep mode while waiting for the next grant. On the other hand, as shown in Fig. 2, since the ONU is only required to process its designated grant message after the expired ONU-based sleep period, it can spend the grant processing period T_{gr}^{proc} in the *doze* mode. Here, the ONU consumes P_i^d instead of P_i^a (hence, overhead $T_i^{o,2}$ is displayed instead of $T_i^{o,1}$) so as to save maximum energy.

We note that with GBA, all the ONUs may be "sleeping" simultaneously. Hence, the OLT may then go into the sleep mode when all the ONUs are in the sleep mode, which is well known for the OLT at the beginning of every GBA cycle. This can significantly reduce the power consumption of a PON system and achieve high energy efficiency. However, new OLT architectures and sleep control mechanisms will be required

⁵Equation has been enhanced, different from the IEEEExplore available one.

to avoid the aforementioned synchronization problem (e.g., by keeping the OLT clock always functional, etc.), which is subject to further investigations and is outside the scope of this work. Interested readers are referred to [12] for a comprehensive overview of GBA.

D. ONU Sleep Mode Condition

Proposition II.1. *Under GBA, ONU i can go into the sleep mode if and only if the following condition is satisfied:*

$$T_i^s \geq \sum_{j \in \mathbf{N}, j \neq i} T_j^a + |\mathbf{N}| \times T_g^a + v_{s,i} - T_i^o, \quad \forall i \in \mathbf{N}. \quad (10)$$

In specific, the sleep time (plus the overhead time) of an ONU should be greater than or equal to the transmission time of all other ONUs (which are separated with a guard time), plus the OLT serving time.

Proof: In TDMA-PONs, a necessary condition for an ONU to go into the sleep mode is that it is idle [10]. With GBA, the ONU is expected to sleep at least when it is idle, which gives us the following:

$$T_i^s \geq T_i^{idle} - T_i^o, \quad (11)$$

where T_i^{idle} is the ONU idle time, which is equivalent to the time while other ONUs are transmitting and receiving data, plus the time spent by the OLT to serve ONU i . Thus, T_i^{idle} can be formulated as follows:

$$T_i^{idle} = \sum_{j \in \mathbf{N}, j \neq i} T_j^a + |\mathbf{N}| \times T_g^a + v_{s,i}. \quad (12)$$

Replacing (12) in (11) ends our proof. ■

Remark II.1. *Breaking the condition in Eq. (10) also means that the ONU cannot be idle and hence the TDMA-PON system is not stable. We therefore assume that this condition always holds and the system is stable.*

III. MAXIMUM ONU SLEEP TIME

Computing an “extended” ONU sleep time is the core theme of GBA. With this feature, the frequent control messaging overhead between the OLT and ONUs is reduced, and the ONU listening period in the *cyclic sleep* mode is saved [5]. Hence, the network throughput will be ameliorated and less energy will be consumed by the ONU. However, the maximum ONU sleep time should be taken such as the user’s QoS requirements are not violated.

In this section, we introduce an analytical model to measure the expected delay for each CoS c of ONU i based on the QoS information sent by the ONU such as the number of CoSs, buffers occupancies, and arrival rates. Using the measured expected delay and the maximum delay requirement for every CoS, we derive a closed-form expression of the ONU sleep time so as to optimize the energy saving without impairing the QoS delay for each type of traffic. To the best of our knowledge, our model is regarded as the first research effort to analytically determine the ONU sleep time based on a given packet delay constraint.

A. Preliminaries

To compute the expected packet delay, we first denote:

- $W_{c,i}$: The waiting time in CoS queue c of ONU i .
- R_i : The residual service time as seen by a packet arriving to ONU i in any CoS queue.
- N_i : The total number of the packets, in all priority queues, waiting to be transmitted by ONU i .
- Y_i : The reservation/gap interval in which ONU i is not active, as seen by a packet arriving to any CoS at the ONU before it is transmitted.

To facilitate the analysis, we illustrate in Fig. 3 the ONU sleep cycle denoted as C_i , which consists of the sleep and overhead vacation times, and the ONU active period denoted as T_i^a . In the active period, the ONU spends a busy period $Z_{c,i}$ ($c = 1, \dots, n$) for each served queue in a round robin fashion, such that the number of packets selected from each queue is based on the scheduling discipline in the active period (e.g., gated, limited, etc.). Finally, it spends a “vacation” period $T_{br}^{proc} + T_{br}^{tran}$ for control messages.

Our model focuses on extracting an expression of the maximum/optimal ONU sleep time T_i^s as whole, independent of how it is broken under GBA (see Fig. 3(a)) and then calibrated using SAS (as we will see in the next section). Therefore, with $T_i^s = T_i^{s,1} + T_i^{s,2}$ and $T_i^o = T_i^{o,1} + T_i^{o,2}$, the ONU sleep cycle can be simplified as shown in Fig. 3(b)). Consequently, we denote the total vacation period as v_c , such that:

$$v_c = T_{br}^{proc} + T_{br}^{tran} + T_{gr}^{proc} \quad (13)$$

To make our model tractable, we assume the data arrivals of all classes to be Poisson-distributed with arrival rate $\lambda_{c,i}$. Hence, we model each ONU as an M/G/1 with Vacations [13].

Remark III.1. *Although Poisson can be seen as the simplified assumption, the batch-mode transmission in GBA makes the system performance indifferent to the arrival traffic pattern that gets smoothed due to the long buffering time at each ONU and the OLT. Furthermore, there are many applications to PON in which the type of traffic that is present in the network can be modeled using Poisson. Specifically, we have shown in our previous work in [20] that when PON is provisioned to support service envelopes (i.e., flows shaped using a traffic policer such as Leaky Bucket, etc.), Poisson distribution is able to accurately capture their behavior.*

Note that in our GBA framework, unlike traditional TDMA-PONs, the sleep time of one ONU is at least the time when other ONUs are transmitting or receiving traffic. Therefore, the ONU sleep cycle C_i and consequently the sleep time T_i^s are much larger than the traditional TDMA-PON cycle time (typically 2 ms). As a result, the whole system can be represented by only one M/G/1 multi-class ONU server with multiple vacation periods.

The first and second moments of the packet transmission times belonging to class c and ONU i can be expressed as $E\{X_{c,i}\} = \bar{X}_{c,i} = 1/\mu_{c,i}$, and $E\{X_{c,i}^2\} = \bar{X}_{c,i}^2$, respectively, with $\mu_{c,i}$ denoting the service rate of a CoS c packet belonging to ONU i , and μ_i as the mean overall service rate. The utilization factor per class c of ONU i is $\rho_{c,i} = \lambda_{c,i} \bar{X}_{c,i}$.

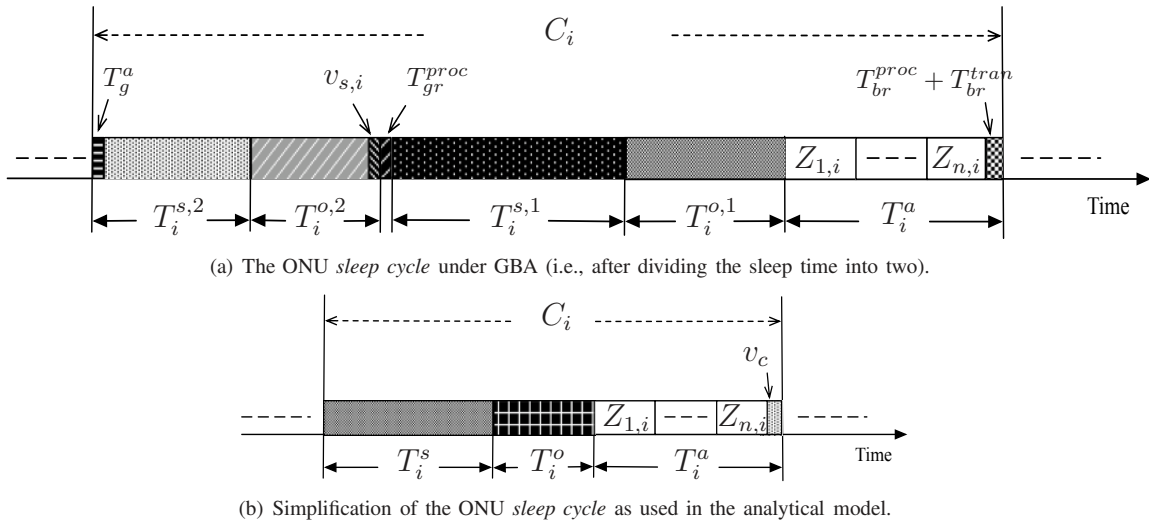


Fig. 3: Illustration of ONU sleep cycle.

We also denote \overline{V}_i and \overline{V}_i^2 as the first and second moment of ONU i 's vacation time, respectively. We assume that a steady state always exists. That is, $\rho_i = \rho_{1,i} + \rho_{2,i} + \dots + \rho_{n,i} < 1$.

The vacation time for ONU i , denoted as v_i , is given by:

$$v_i = v_c + T_i^s + T_i^o. \quad (14)$$

B. Expected Packet Delay

Each incoming packet ξ may arrive to any ONU i during C_i and can belong to any class $c = 1, \dots, n$. Packet ξ will be subject to a total delay $D_{c,i}$, given as:

$$D_{c,i}(\xi) = W_{c,i}(\xi) + T_i^{prop} + X_{c,i}(\xi), \quad (15)$$

where $X_{c,i}(\xi)$ are ξ 's transmission time. In the steady state, as $\xi \rightarrow \infty$, the expected value of the total delay is given as:

$$\begin{aligned} E\{D_{c,i}\} &= E\{W_{c,i}\} + T_i^{prop} + E\{X_{c,i}\} \\ &= E\{W_{c,i}\} + \frac{d_i}{C_{light}} + \frac{1}{\mu_{c,i}}. \end{aligned} \quad (16)$$

Thus to get $E\{D_{c,i}\}$, we need to get the value of $E\{W_{c,i}\}$.

C. Queuing Delay Analysis

The queuing delay of ξ arriving at ONU i is expressed as:

$$W_i(\xi) = R_i(\xi) + N_i(\xi) \times X_i(\xi) + Y_i(\xi), \quad (17)$$

where $X_i(\xi)$ is the service time of packet ξ . Since Little's theorem holds regardless of the packet service order, it follows that $E\{N_i\} = \lim_{\xi \rightarrow \infty} N_i(\xi)$. Similarly, $E\{R_i\} = \lim_{\xi \rightarrow \infty} R_i(\xi)$, $E\{Y_i\} = \lim_{\xi \rightarrow \infty} Y_i(\xi)$, and $E\{W_i\} = \lim_{\xi \rightarrow \infty} W_i(\xi)$.

A packet ξ belonging to class $1 \leq c \leq n$ will exhibit an expected queuing delay equal to the following:

$$E\{W_{c,i}\} = E\{R_i\} + \frac{E\{N_i\}}{\mu_i} + E\{Y_i\}. \quad (18)$$

Hence in order to get the value of $E\{W_{c,i}\}$, we need to find $E\{N_i\}$, $E\{R_i\}$, and $E\{Y_i\}$. While $E\{R_i\}$ is determined

based on the type of system employed (here, M/G/1 with vacations), the values of $E\{N_i\}$ and $E\{Y_i\}$ depend on the type of scheduling discipline employed by the DBA [14].

Lemma III.1. *The mean residual time seen by a packet arriving at ONU i in any CoS c is given as follows:*

$$E\{R_i\} = \frac{1}{2} \left[\left(\sum_{j=1}^n \lambda_{j,i} \right) \overline{X_i^2} + \frac{\overline{V_i^2} (1 - \rho_i)}{\overline{V_i}} \right]. \quad (19)$$

Proof: To find the value of $E\{R_i\}$, we use the concept of the mean residual service time with a graphical argument. In our system, there are multiple queues and multiple vacation periods; and both periods do not appear continuously (they alternate in every C_i). Hence, the value of the residual time as function of packet ξ 's arrival time τ , $R_i(\tau)$, is displayed as illustrated in Fig. 4.

Considering that triangles in Fig. 4 are all right-angled and isosceles, the time average of $R_i(\tau)$ in $[0, t]$ is computed as follows:

$$\begin{aligned} E\{R_i\} &= \frac{1}{t} \int_0^t R_i(\tau) d\tau \\ &= \frac{1}{t} \left[\sum_{j=1}^n \left(\sum_{x_{j,i} \in S_{j,i}(t)} \frac{1}{2} x_{j,i}^2 \right) + \sum_{y_i \in L_i(t)} \frac{1}{2} y_i^2 \right], \end{aligned} \quad (20)$$

where $L_i(t)$ is the number of vacations seen in C_i , and $S_{j,i}(t)$ denotes the number of CoS j serviced packets. $x_{j,i}$ denotes a class j serviced packet, and y_i denotes the single vacation time seen by ONU i , all during $[0, t]$.

By arrangement of parameters, we can rewrite Eq. (20) as follows:

$$E\{R_i\} = \frac{1}{2} \left[\left(\sum_{j=1}^n \frac{S_{j,i}(t)}{t} x_{j,i}^2 \right) + \left(\frac{L_i(t)}{t} \times y_i^2 \right) \right]. \quad (21)$$

We note that [14]:

$$\lim_{t \rightarrow \infty} \frac{S_{j,i}(t)}{t} \rightarrow \lambda_{j,i}.$$

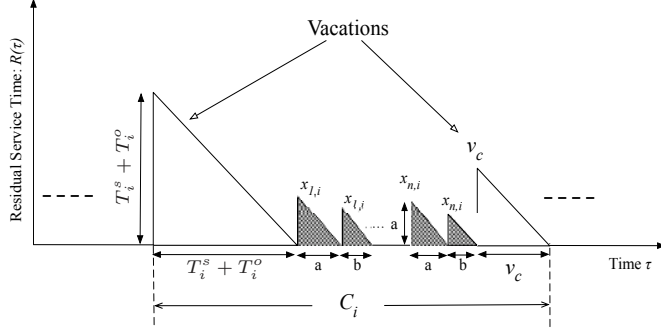


Fig. 4: Residual time derivation.

Furthermore, as $t \rightarrow \infty$, the fraction of time spent serving packets from all queues approaches ρ_i , and thus the fraction of time occupied with vacations is $(1 - \rho_i)$. Assuming time averages can be replaced by ensemble averages, we get:

$$\lim_{t \rightarrow \infty} \frac{t(1 - \rho_i)}{\bar{V}_i} \rightarrow L_i(t).$$

Replacing these approximations and converting to ensemble averages in Eq (21) (e.g., $y_i^2 \rightarrow \bar{V}_i^2$) gives us the expression of $E\{R_i\}$ in Eq. (19), such that \bar{X}_i^2 denotes the second moment of ONU i 's service time averaged over all classes. More specifically,

$$\bar{X}_i^2 = \frac{\lambda_{1,i}}{\sum_{j=1}^n \lambda_{j,i}} \bar{X}_{1,i}^2 + \dots + \frac{\lambda_{n,i}}{\sum_{j=1}^n \lambda_{j,i}} \bar{X}_{n,i}^2, \quad (22)$$

which ends our proof. ■

Many scheduling disciplines have been considered for upstream DBA design in PONs. Most famously, IPACT with gated service has been taken as the mainstream scheduling discipline for most DBAs in TDMA-PONs [15], [17], [19], [21]. Therefore to find $E\{N_i\}$ and $E\{Y_i\}$, we proceed with the analysis given that a gated service is employed. Our focus on gated service is motivated by its smallest packet delays among various scheduling disciplines, particularly at high network loads [16], which is a desired feature for mission-critical services and virtual private networks (VPNs) [22]. However, we note that the gated service may not achieve fairness among ONUs; thus if the latter is set as a salient network requirement, other scheduling disciplines should instead be analyzed (e.g., limited service [17]).

D. Analysis with Gated Service

With gated service, the ONU sends a request message in each GBA cycle to the OLT containing the buffer occupancy for each of its n priority queues. The OLT then assigns to each priority queue upstream bandwidth equal to the requested bandwidth. When the ONU switches to the active mode, using its Intra-ONU scheduler, it selects traffic from each CoS queue based on the granted transmission slot. Consequently, our system can be seen as a cyclic gated server with vacations [23].

In the steady state, packet ξ of CoS c will see the same average number of packets queued, at both its enqueueing and dequeueing times. Using Little's theorem, the expected number

of arriving packets to all CoSs while ξ is waiting, can be expressed by $\sum_{j=1}^n \lambda_{j,i} \times E\{W_{c,i}\}$. By breaking N_i and μ_i into its priority class components, we get the following:

$$\begin{aligned} \frac{E\{N_i\}}{\mu_i} &= \sum_{j=1}^n \lambda_{j,i} \bar{X}_{j,i} \times E\{W_{c,i}\} \\ &= \rho_i \times E\{W_{c,i}\}. \end{aligned} \quad (23)$$

Hence, the expected queuing delay of a CoS c packet is:

$$E\{W_{c,i}\} = \frac{E\{R_i\} + E\{Y_i\}}{1 - \rho_i}. \quad (24)$$

Remark III.2. In Eq. (23), N_i is taken because using GBA with gated service, any packet ξ belonging to any priority queue will wait until all the packets that were buffered before its arrival, in higher or lower priority queues, to be transmitted in the current or next GBA cycle. All these packets form N_i . Consequently, the delay expression $E\{W_{c,i}\}$ in Eq. (24) will be similar for every CoS c of each ONU i . Of course, different scheduling disciplines (e.g., strict priority) will entitle different number of packets, and hence different ONU sleep time expression.

Lemma III.2. The expected interval time $E\{Y_i\}$ that the packet waits for in every ONU sleep cycle is given as follows:

$$E\{Y_i\} = T_i^o + T_i^s + v_c. \quad (25)$$

Proof: In a gated system, $E\{Y_i\}$ is the total vacation time appearing in one server cycle [14]. In our system, the total vacation time appearing in C_i is v_i , which is $T_i^o + T_i^s + v_c$. ■

Lemma III.3. The expected queuing delay in a PON system for a CoS c packet of ONU i , under a gated service, is:

$$E\{W_{c,i}\} = \frac{\left(\sum_{j=1}^n \lambda_{j,i}\right) \bar{X}_i^2 + (3 - \rho_i)v_i}{2(1 - \rho_i)}. \quad (26)$$

Proof: By combining $E\{R_i\}$ from Eq. (19) and $E\{Y_i\}$ from Eq. (25), we obtain the expected queuing delay $E\{W_{c,i}\}$ for CoS c : $1 \leq c \leq n$ as follows:

$$E\{W_{c,i}\} = \frac{\left(\sum_{j=1}^n \lambda_{j,i}\right) \bar{X}_i^2}{2(1 - \rho_i)} + \frac{v_i}{1 - \rho_i} + \frac{\bar{V}_i^2}{2\bar{V}_i}. \quad (27)$$

We have $\bar{V}_i^2 = (\bar{V}_i)^2 = v_i^2$. Then, by substituting these values in Eq. (27), we get $E\{W_{c,i}\}$ in Eq. (26). ■

E. Expression of ONU Sleep Time

Theorem III.1. Given the one-way delay requirement $\mathbb{D}_{c,i}$, ONU i can sleep for a period $T_{c,i}^s$, expressed as follows:

$$\frac{2(1 - \rho_i) \left(\mathbb{D}_{c,i} - \frac{d_i}{C_{light}} - \frac{1}{\mu_{c,i}} \right) - \left(\sum_{j=1}^n \lambda_{j,i} \right) \bar{X}_i^2}{(3 - \rho_i)} - T_i^o - v_c. \quad (28)$$

Proof: By substituting the value of $E\{W_{c,i}\}$ from Eq. (16) in Eq. (26), and by extracting T_i^s from v_i as well as inverting the equation, we get $T_{c,i}^s$. ■

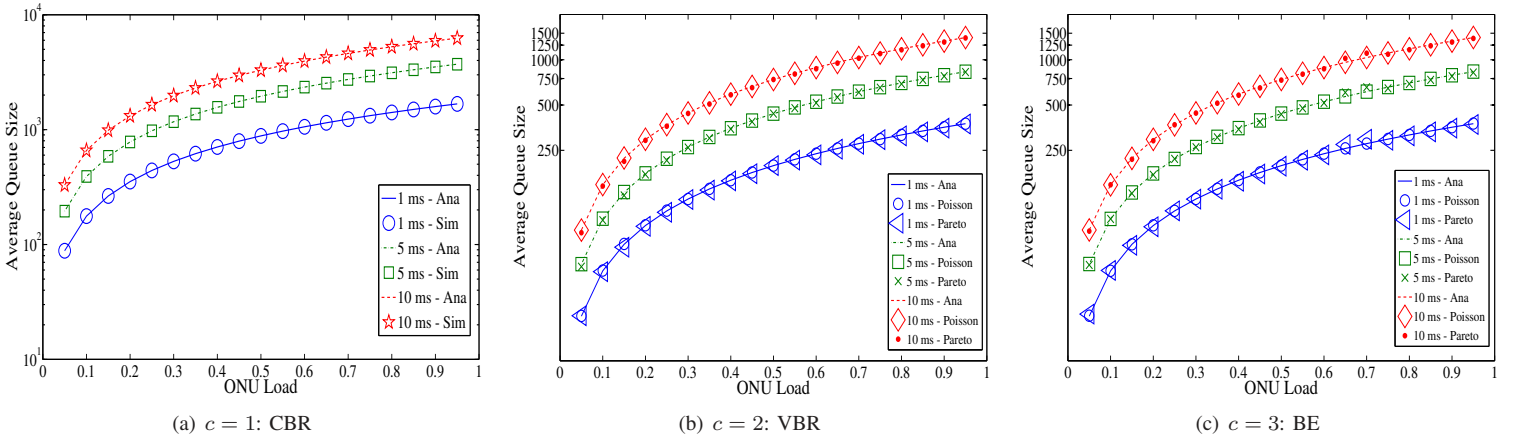


Fig. 5: Average queue size for GR-ONU-1/A (i.e., $T_i^o = 2.125$ ms) and $T_i^s = 1, 5$ and 10 ms.

Finally T_i^s is set as the minimum value as follows:

$$T_i^s = \min(T_{1,i}^s, \dots, T_{n,i}^s). \quad (29)$$

F. ONU “Wake Up Call”

By replacing the value of T_i^s in Eq. (26), we can get the maximum $E\{W_{c,i}\}$. Then using Little’s theorem, we compute the expected average queue size of class c and ONU i , $E\{Q_{c,i}\}$, as follows:

$$E\{Q_{c,i}\} = \lambda_{c,i} E\{W_{c,i}\}. \quad (30)$$

Consequently, as an alternative to setting a timer which yields additional overhead due to clock synchronization, each ONU may hold a “counter” for each CoS queue, denoted $count_{c,i}$, to measure the number of buffered packets in each queue. When $count_{c,i} = E\{Q_{c,i}\}$, the ONU switches to the active state.

Remark III.3. The computed sleep time T_i^s may be bounded by 50 ms or 100 ms in case it was chosen to comply with EPON or GPON’s signaling protocols’ timeouts [11], [24]. However, this can possibly limit the maximum saved energy depending on the delay requirement. Alternatively, the protocols can be amended to enable dynamic timeouts so that T_i^s can exceed 50 ms or 100 ms.

On the other hand, while our analysis is based on the upstream packet delay requirement, measuring the ONU sleep time based on the expected downstream packet latency and its respective requirement (if any) remains to be investigated. When available, the OLT can set the ONU sleep time to be the minimum between the upstream-based and downstream-based values. Although this might not achieve maximum energy saving, it can ensure guaranteed delay performance in the downstream and upstream directions.

G. Model Validation & Observations

To validate the effectiveness of the proposed model, we have simulated an EPON network using OMNET++. The simulations parameters are shown in Table III. However as mentioned, since the system is represented by a single ONU and because the existence of multiple ONUs might create a

TABLE III: Simulations parameters.

Number of ONUs ($ N $)	16
Channel speed (C_{pon})	1 Gbps
Speed of light (C_{light})	$\approx 300,000$ km/s
Distance from OLT to ONU (d_i)	25 km
Guard time (T_g^a and T_g^{gr})	1 μ s
Processing time (T_{gr}^{proc} and T_{br}^{proc})	10^{-3} μ s
GBA computation time (T_{gba})	neglected

collision (which we resolve in Section IV), for model verification purposes, only one ONU is activated and monitored at a given time. The traffic profile for each ONU is divided as follows: 20% of the total generated traffic is considered for CBR traffic, and 80% is divided equally between VBR and best effort (BE) traffics [19]. CBR flows have fixed size packets equal to 70 bytes; whereas VBR and BE flows have packet payload sizes that vary from 64 to 1518 bytes with the distribution based on [19] as follows: 64 bytes (47%), 300 bytes (5%), 594 bytes (15%), 1300 byte (5%), and 1518 bytes (28%). VBR and BE traffics are simulated as both Poisson and Pareto-distributed (i.e., self-similar traffic with Hurst $H = 0.8$), whereas CBR traffic is simulated as Poisson [19]. The 95% confidence interval of the simulation results gives a very small variation, and is thus not shown in the figures.

We first plot in Fig. 5 the average queue size of each CoS (mainly used to switch the ONU back to the active mode) under an ONU architecture of type 1 with an aggressive wake up overhead time. The figure shows that while Poisson-simulated traffics perfectly match with the analytical model, only a small deviation of ≈ 10 -15% at higher loads (e.g., $\rho_i = 0.6, 0.7$) is observed for Pareto-simulated traffics from the analytical results, especially with longer sleep times. Therefore, we can claim that our model can be effectively used to compute the ONU sleep time without major adjustments.

We then show in Fig. 6 the sleep time T_i^s (computed as per Eq. (29) with a conservative wake up overhead time (i.e., $T_i^o = 5.125$ ms) versus the maximum delay requirement

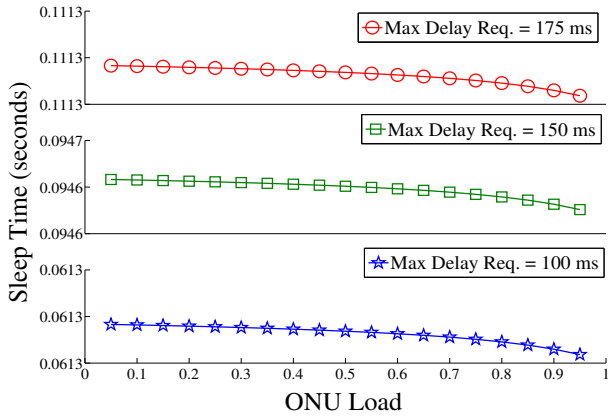


Fig. 6: T_i^s for GR-ONU-1/C (i.e., $T_i^o = 5.125$ ms).

$\mathbb{D}_{c,i} = 100, 150,$ and 175 ms. Numerical results show that for a 100 ms delay requirement, the ONU can sleep for 61.3 ms and for a 150 ms delay requirement (which is the typical maximum voice delay requirement [25]), it can sleep for 94.7 ms; and finally for a 175 ms delay requirement, the ONU can sleep for 111.3 ms. These numbers highlight the advantages of the proposed framework which leverages the GBA's batch-mode transmission to minimize the wasted overhead so as to save maximum possible energy while maintaining the users' requirements.

Remark III.4. It is observed in Fig. 6 that the sleep time barely varies as the load increases (in fact the variation is in fractions of milliseconds, therefore is not noticed). This is due to the fact that the value of T_i^s is DBA-dependant (in our case we employ the gated service). For more details on the delay variation under the gated service in PONs, we refer the reader to [17]. There, unlike in our model, multiple ONUs contribute to the vacation time, which causes higher packet delay as the load increases, yet in the order of few milliseconds.

IV. SLEEP TIME SIZING

With GBA, every ONU is required to transmit/receive traffic immediately after $T_i^{s,1}$ is expired; hence, a collision might occur. Thus, the OLT is required to resize and schedule the OLT-based sleep times of all ONUs such that the collision is avoided. Furthermore, it should ensure that the total assigned sleep time T_i^s is fully granted in order to achieve maximum energy saving.

A. Problem Formulation

To formulate the problem, we first define the transmission start time and end time of every ONU as follows (see Fig. 7):

$$t_i^{st} = t_i^0 + T_i^{s,1} + T_i^{o,1}, \quad (31)$$

$$t_i^{end} = t_i^0 + T_i^{s,1} + T_i^{o,1} + T_i^a, \quad (32)$$

where t_i^0 denotes the time instant when the grant message is received and processed at the ONU, given as:

$$t_i^0 = t_0 + T_{g,ba} + T_i^{prop} + T_{gr}^{tran} + (i-1)T_g^{gr} + (i+1)T_{gr}^{proc}, \quad (33)$$

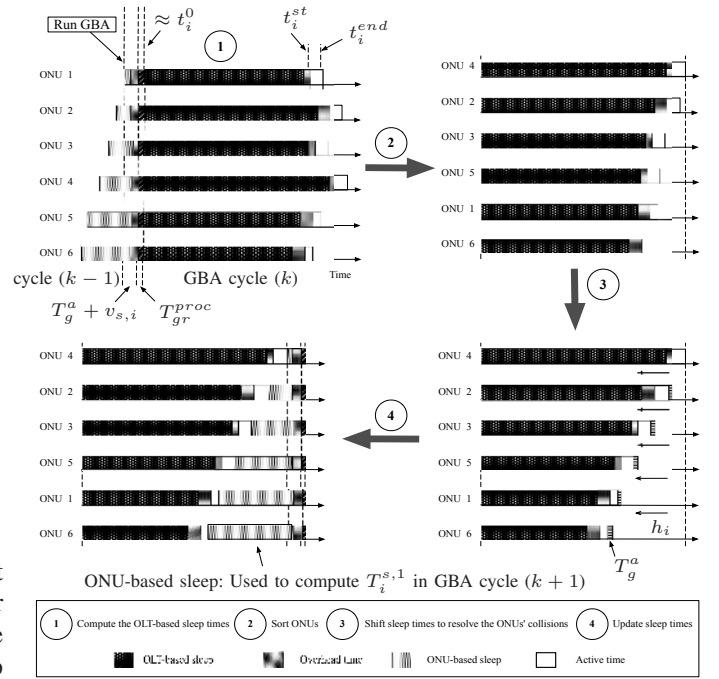


Fig. 7: Illustration of the proposed SAS mechanism.

with t_0 denoting the time instant when the OLT processes all the received ONUs' requests.

For equally distant ONUs, and given the negligible grant guard time ($T_g^{gr} = 1$ μ s) and grant processing time ($T_{gr}^{proc} = 10^{-3}$ μ s), the instant time t_i^0 is almost the same for all ONUs (i.e., $t_1^0 \approx t_2^0 \approx \dots \approx t_{|N|}^0$).

Assuming that a collision takes place with the current start and end times of all ONUs, each ONU i will be required to shift its sleep time with a portion h_i until the collision is eliminated. However, this should be done while ensuring that the computed sleep time $T_i^{s,1}$ gets its maximum value. As a result, we formulate the sleep time sizing problem as follows:

$$\text{Find: } \max \left(T_i^{s,1} - h_i \right), \quad \forall i \in \mathbb{N}, \quad (34)$$

which is equivalent to:

$$\text{Find: } \min \left(\sum_{i \in \mathbb{N}} h_i \right), \quad (35)$$

subject to:

$$t_j^{st} - h_j \geq t_i^{end} - h_i + T_g^a \quad \forall i, j \in \mathbb{N}, \quad j > i. \quad (36)$$

$$h_i \leq T_i^{s,1} + T_i^{o,1} \quad \forall i \in \mathbb{N}. \quad (37)$$

Constraint (36) requires the transmission start time of an ONU to be larger than or equal to the transmission end time of any other ONU of lower index plus a guard time. Constraint (37) means that the shifted time cannot be greater than the sleep time (plus the overhead time) of the ONU.

B. Sort-And-Shift (SAS) Scheme

To solve the active time collision problem, we propose a new sleep time sizing scheme, so-called Sort-And-Shift (SAS),

which is initiated at the OLT to determine the start time t_i^{st} of each ONU in every GBA cycle. With SAS, the ONUs are sorted according to their expected transmission start times; and t_i^{st} , $\forall i \in N$, is then determined and updated sequentially such that no collision between any ONU's transmission occurs (i.e., to enforce Constraint 36). Consequently, Constraint (36) will be rewritten as follows:

$$h_i \geq h_j + T_g^a + t_i^{st} - t_j^{st} + T_i^a. \quad (38)$$

The solution to the problem is optimal by setting the shift time of the ONU with the nearest transmission start time (or the "last" ONU after the sorting), $h_{|N|} = 0$. The steps of the SAS mechanism are shown in Fig. 7 and the detailed pseudo-code is given in Algorithm 1. The shifted time is included in the ONU-based sleep time in order to ensure that the ONU is fully granted its assigned sleep time. The ONU-based sleep time is computed as follows: Let $\mathcal{A}_i = t_{|N|}^{end} + v_{s,i} + T_g^a - t_i^{end}$,

$$T_i^{s,2} = \begin{cases} \mathcal{A}_i - T_i^{o,2} & \text{if } \mathcal{A}_i > T_i^{o,2}. \\ 0 & \text{otherwise.} \end{cases} \quad (39)$$

As observed, with SAS, the ONU sleep cycle C_i spans two

Algorithm 1 Sort-And-Shift (SAS) mechanism

```

1: for  $i = 1 \rightarrow |N|$  do
2:   Given  $T_i^{s,2}$  from GBA cycle ( $k-1$ ),
3:   compute  $T_i^{s,1}$  for GBA cycle ( $k$ ) according to Eq. (9).
4: end for
5: /* Initialize and sort the set of ONU start-times  $\mathbf{X}$  */
6:  $\mathbf{X} \leftarrow \{t_1^{st}, t_2^{st}, \dots, t_{|N|}^{st}\}$ ;
7: Sort  $\mathbf{X}$  s.t.,  $t_i^{st} < t_j^{st}, \forall i, j \in N$ 
8: /* Set the shift time at the last index to zero */
9:  $h_{|N|} \leftarrow 0$ ;
10: /* Compute the ONU shift time according to Eq. (38) */
11: for  $i = |N| - 1 \rightarrow 1$  do
12:    $h_i \leftarrow h_{i+1} + t_i^{st} - t_{i+1}^{st} + T_i^a + T_g^a$ ;
13: end for
14: /* Update all values if overlapping occurs */
15: for  $i = |N| - 1 \rightarrow 1$  do
16:   if  $h_i > 0$  then
17:      $T_i^{s,2} \leftarrow T_i^{s,2} - h_i$ ;
18:      $t_i^{st} \leftarrow t_i^{st} - h_i$ ;
19:      $t_i^{end} \leftarrow t_i^{end} - h_i$ ;
20:   end if
21: end for
22: /* Compute the ONU-based sleep time */
23: for  $i = 1 \rightarrow |N|$  do
24:   Compute  $T_i^{s,2}$  according to Eq. (39).
25: end for

```

consecutive GBA cycles, and the OLT-based sleep time in the current GBA cycle is computed based on how much the ONU has slept (using the ONU-based sleep time) in the previous GBA cycle. Hence, the OLT-based sleep time in the current GBA cycle can be taken as "remaining credit" after subtracting the ONU-based sleep time of the previous GBA cycle from the total assigned sleep time T_i^s , which is computed using Eq. (29) at the end of each GBA cycle.

Note that if the traffic requirements of all ONUs remain unchanged, the OLT may apply the same polling order in every GBA cycle so as to accurately maintain $v_{s,i}$ that was used to verify Eq. (10) in the previous GBA cycle; otherwise before

running SAS, in addition to T_i^s , the OLT will have to re-compute $v_{s,i}$ and verify Eq. (10) for each ONU in every GBA cycle.

Theorem IV.1. *The obtained solution by SAS is optimal.*

Proof: The obtained solution is optimal if it satisfies constraints (36) and (37), and ensures $T_i^{s,1} + T_i^{s,2} = T_i^s$.

Constraint (36) is always satisfied by computing h_i according to Eq. (38). For constraint (37), the ONU is assumed to be capable of going into the sleep mode. Thus, Eq. (10) is always true, and we re-write it as follows:

$$T_i^{s,1} + T_i^{s,2} + T_i^{o,1} + T_i^{o,2} - h_i \geq \sum_{j \in N, j \neq i} T_j^a + |N| \times T_g^a - h_i + v_{s,i}. \quad (40)$$

We note that:

$$h_i \leq T_g^a \left(\sum_{j \in N, j > i} T_j^a \right). \quad (41)$$

The worst-case scenario occurs when the shifted time of ONU i is equivalent to the sum of all other ONUs' data intervals (e.g., the worst case scenario for ONU 6 in Fig. 7). Assuming such a scenario for ONU i , we then get:

$$h_i = \sum_{j \in N, j \neq i} T_j^a + (|N| - 1) \times T_g^a,$$

which reduces Eq. (40) to:

$$T_i^{s,1} + T_i^{s,2} + T_i^{o,1} + T_i^{o,2} - h_i \geq v_{s,i} + T_g^a.$$

From Eq. (9), we can have two cases:

- 1) $T_i^{s,2} + T_i^{o,2} \geq v_{s,i} + T_g^a$.
- 2) $T_i^{s,2} = T_i^{o,2} = 0$.

For the first case, we can write $T_i^{s,2} + T_i^{o,2} = v_{s,i} + T_g^a$, as the minimum value (e.g., the case of ONU 4 in Fig. 7), and hence we get:

$$T_i^{s,1} + T_i^{o,1} \geq h_i.$$

For the second case, we get:

$$T_i^{s,1} + T_i^{o,1} \geq h_i + v_{s,i} + T_g^a > h_i.$$

Finally, SAS ensures $T_i^{s,1} + T_i^{s,2} = T_i^s$ by allocating the shifted time in the ONU-based sleep time as per Eq. (39). Hence, the solution is optimal. ■

C. Time Complexity of SAS

In this section, we analyze the time complexity of SAS, and we show that it is very suitable for real PON deployments.

Lines 1 \rightarrow 4 in Algorithm 1 compute $T_i^{s,1}$ for every ONU, which is of order $O(|N|)$. The same order applies for lines 11 \rightarrow 13, 15 \rightarrow 19, and 23 \rightarrow 25. For the sorting part of the algorithm (i.e., line 7), the runtime depends on the employed sorting algorithm. Some algorithms can be faster than others (e.g., merge sort, bucket sort, quick sort, etc.). The fastest one that we use is quick sort, which, on average, has a runtime of order $O(|N| \log |N|)$. Nevertheless, the sorting space (i.e., N)

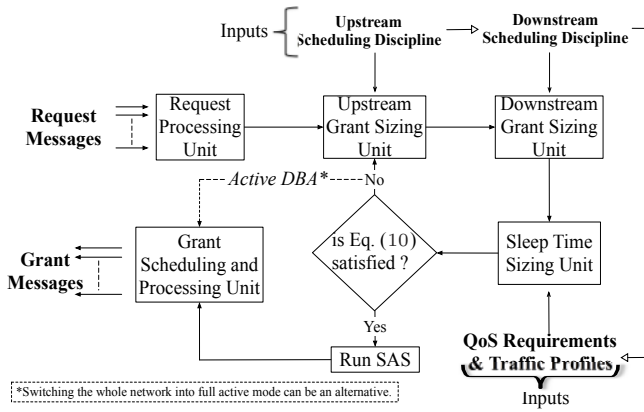


Fig. 8: Green resource management flowchart at the OLT.

is very small, since a typical TDMA-PON network connects a maximum number of 128 ONUs [24]. Therefore, SAS can be extremely fast, and hence can be employed in practical PON deployments.

D. Updated ONU Server Vacation Time

With GBA, T_i^s is divided into two, and the ONU sleep cycle is as shown in Fig. 3(a). Hence, we can re-write the vacation time of ONU i in Eq. (14) as follows:

$$v_i = T_i^{s,2} + T_i^{o,2} + T_{gr}^{proc} + T_i^{s,1} + T_i^{o,1} + T_{br}^{proc} + T_{br}^{tran}. \quad (42)$$

By always taking the total overhead, the computation of the ONU sleep time (in Section III) assumes that $T_i^{s,2} + T_i^{o,2} \geq v_{s,i} + T_g^a$, $\forall i \in \mathbf{N}$. This will not be true if, after running SAS, ONU i ends up having its ONU-based sleep time equal to zero as per Eq. (39). Consequently, $T_i^{o,2}$ will need to be excluded from the vacation time v_i , which should alternately include the time gap $v_{s,i} + T_g^a$ (see Fig. 3(a)). Hence at the end of every GBA cycle, v_i in Eq. (14) should be evaluated as follows:

With $v_c = T_{br}^{proc} + T_{br}^{tran} + T_{gr}^{proc}$,

$$v_i = \begin{cases} T_i^s + T_i^{o,1} + T_i^{o,2} + v_c & \text{if } T_i^{s,2} > 0. \\ T_i^s + T_i^{o,1} + v_{s,i} + T_g^a + v_c & \text{otherwise.} \end{cases} \quad (43)$$

$T_{c,i}^s$ in Eq. (28) and T_i^s in Eq. (29) are then computed accordingly.

E. Green Resource Management Framework for PONs

As illustrated in Fig. 8, by embedding SAS and the proposed ONU sleep time model of Section III in the proposed GBA framework [12], we can achieve a complete *green* resource management framework for PONs. Here, in every GBA cycle, upon receiving the request messages from all ONUs, the OLT performs the upstream and downstream bandwidth allocation based on the selected scheduling discipline(s) (e.g., UCS with gated service, etc.). Consequently, given a set of QoS requirements and traffic characteristics, the OLT computes the ONU sleep times and then checks if the ONU sleep mode condition in Eq. (10) is satisfied for every ONU. If the latter is true, this means that the network is not overloaded and thus

all the ONUs can go into the sleep mode. Consequently, SAS is executed to resolve any collision, and sleep and bandwidth grants are issued for all the ONUs. If Eq. (10) is not satisfied for an ONU, the OLT can select different downstream and/or upstream scheduling disciplines for this (or all) ONU(s), and the process is repeated. Alternatively, the OLT may decide to switch the system into “full active” mode and abolish the energy saving plan by employing a legacy DBA.

V. SIMULATION RESULTS

To validate the effectiveness of the proposed solution, we conduct extensive simulations. To stress test our framework, we consider four simulation scenarios as detailed in Table IV. The simulations parameters and traffic settings are the same as in Section III-G. However here, all ONUs are activated so as to test SAS. Note that where applicable, only Scenario 4 is employed when comparing our scheme with the UCS scheme of [10] (which is here implemented as a dynamic time slot allocation rather than the fixed one used in [10]). This is due to the fact that the latter cannot be employed if the overhead time is not shorter than the cycle time. Hence with the existence of legacy ONUs, the UCS protocol cannot be utilized, which highlights the advantage of our proposed framework over UCS, in being operational on top of any deployed ONU architecture.

We first compare in Fig. 9 the total sleep time allocated using SAS under Scenario 1, for three tagged ONUs of different types, versus the assigned sleep time values (i.e., the optimal values). As shown, despite the fact that the sleep time is divided into two parts, SAS still allocates the maximum sleep time for an ONU and hence ensures that the maximum possible energy is saved in the network. Note that a tagged ONU from the final set (e.g., ONU 12) exhibits the same behaviour as ONU 9, due to the same displayed overhead time and a thus closely equivalent computed sleep time. Therefore we do not show it in the figure.

Fig. 10 shows the sleep time behavior of different types of ONUs under SAS. As observed, both the OLT-based and ONU-based sleep times of ONUs with longer sleep times

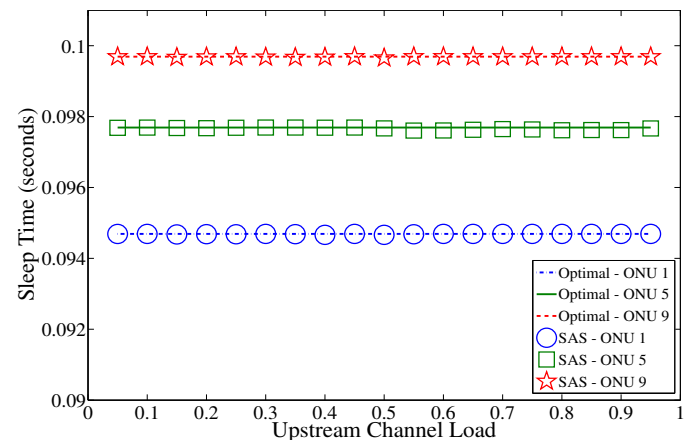
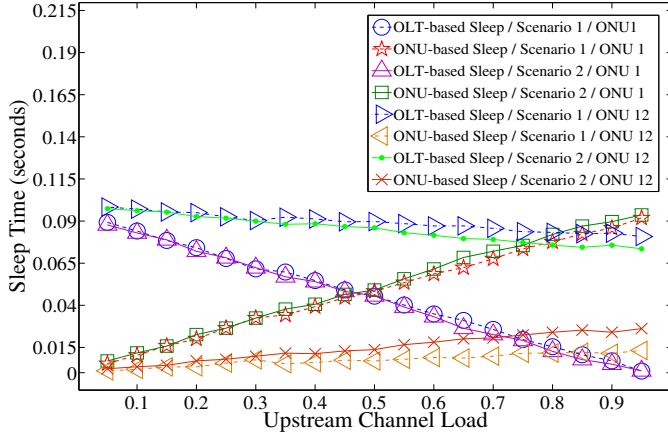


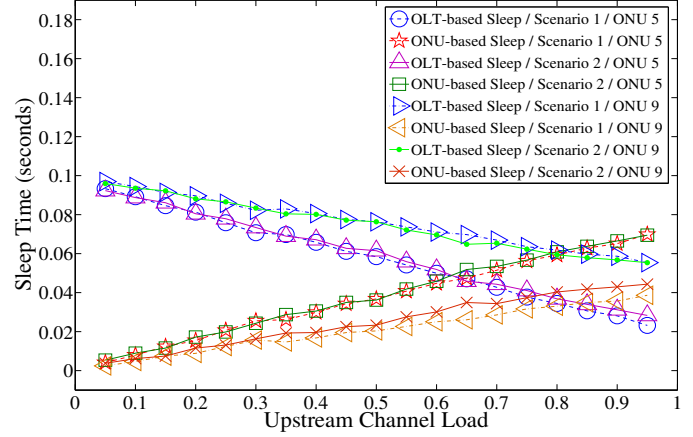
Fig. 9: Measured sleep time vs. optimal sleep time under Scenario 1, with $\mathbb{D}_{c=1,i} = 150 \text{ ms}$, $\forall i \in \mathbf{N}$.

TABLE IV: Simulation scenarios.

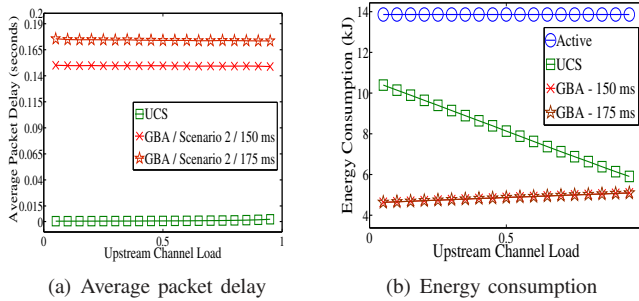
ONU Set	Scenario 1		Scenario 2		Scenario 3		Scenario 4	
	Load	ONU Arch.	Load	ONU Arch.	Load	ONU Arch.	Load	ONU Arch.
{1 → 4}	20%	GR-ONU-1/C	25%	GR-ONU-1/C	20%	GR-ONU-1/C	25%	GR-ONU-3
{5 → 8}	32%	GR-ONU-1/A	25%	GR-ONU-1/A	32%	GR-ONU-1/C	25%	GR-ONU-3
{9 → 11}	36%	GR-ONU-2	25%	GR-ONU-2	36%	GR-ONU-1/C	25%	GR-ONU-3
{12 → 16}	12%	GR-ONU-3	25%	GR-ONU-3	12%	GR-ONU-1/C	25%	GR-ONU-3



(a) ONUs 1 and 12



(b) ONUs 5 and 9

 Fig. 10: Sleep time behaviour under Scenario 1, with $\mathbb{D}_{c=1,i} = 150 \text{ ms}$, $\forall i \in \mathbb{N}$.


(a) Average packet delay

(b) Energy consumption

Fig. 11: Tradeoff between energy efficiency and packet delay.

(e.g., ONU 9 and 12) show a similar behavior under both scenarios, whereas ONUs 1 and 5 which have shorter sleep times exhibit different sleep times behaviors depending on the network load. Notably at moderately high network loads (e.g., $0.5 \rightarrow 0.65$), the sleep times of ONUs 1 and 5 are almost evenly distributed between their ONU-based and their OLT-based sleep times, respectively; whereas the OLT-based sleep time value outweighs the ONU-based one at lower loads, and vice versa at higher loads. This behavior is due to the way SAS adaptively sorts the sleep times of all ONUs so as the remaining residual time between the end time of the last ONU and the end time of each ONU is taken as credit for the OLT-based sleep time in the next GBA cycle.

Fig. 11 highlights the tradeoff between “better network performance” and “higher energy saving” by depicting the average packet delay and the energy consumption of a tagged ONU. The energy consumption of an ONU under GBA and UCS is computed based on the power model in Section II-A

in a one-hour period, such that the sampling unit is the sleep cycle C_i . Hence, the energy consumption (in kJ) of GR-ONU-3 i is computed as follows:

$$E_i = \frac{P_i^s \times T_i^s + P_i^d \times T_i^d + P_i^a (T_i^a + T_i^o)}{C_i} \times \frac{3600}{1000},$$

where T_i^d is the time spent by the ONU in the doze mode. On the other hand, the energy consumption of an active ONU in a one-hour period is simply given by: $3.85 \times 3600/1000 = 13.86 \text{ kJ}$.

As observed with GBA, in contrast to both the UCS scheme and a fully active PON system, although the delay is being sacrificed, the energy consumption for every ONU is significantly decreased. However as shown in Fig. 11(a), the delay performance under GBA depends on the maximum delay requirement. On the other hand, it is noticed in Fig. 11(b) that the energy consumption under UCS is high at low loads, and it decreases at high loads. This counterintuitive behaviour is due to the way we implement UCS in our simulations. Specifically, unlike in [10] where a fixed time slot is allocated, we implement UCS such that a dynamic time slot (based on the gated service discipline) is allocated to the ONUs. Consequently at low loads, when the amount of traffic that needs to be transmitted by all the ONUs is little, the DBA cycle is very small. As a result, the ONU idle time (in which the ONU is supposed to switch to the sleep mode) is smaller than the ONU overhead time, thereby having the ONU almost always unable to go into the sleep mode. In contrast as the load increases, the amount of traffic that needs to be sent by the ONUs grows, which makes the ONU idle time be larger than

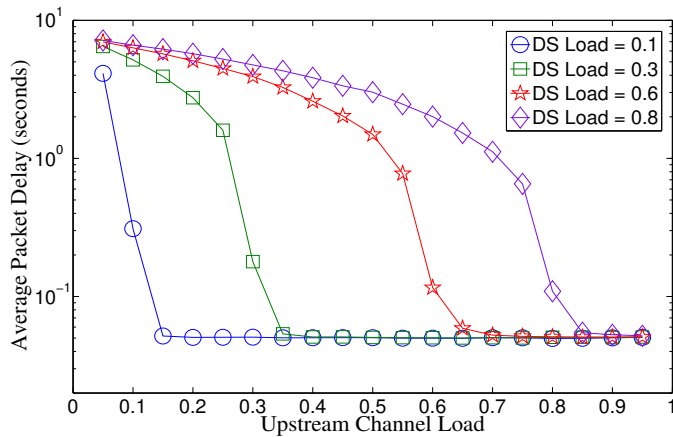


Fig. 12: Average downstream packet delay.

the overhead time, thereby allowing the ONU to go into the sleep mode and reduce its power consumption. On the other hand with GBA, the ONU sleep time is mainly based on the maximum delay requirement, and hence only a small increase in the energy consumption is witnessed as the load increases. Finally we observe in Fig. 11(a) how the delay barely varies as the load increases (under UCS and GBA) which is, as discussed in Remark III.4, due to the gated service discipline.

In Fig. 12, we show how the UCS mechanism may impact the downstream traffic performance. Specifically, when the downstream traffic load is lower than the upstream traffic load, the downstream packet delay is low; whereas it can no longer be controlled if the downstream traffic load is higher than the upstream one. This motivates computing the ONU sleep time based on both the upstream and downstream delay constraints, and then selecting the minimum value between them (as discussed in Remark III.3). Although this may enable lower energy saving than with the current scheme, it can ensure guaranteed network performance in the upstream and downstream directions.

Finally to test the resilience of the proposed model in achieving high energy saving under different scenarios, we show in Table V the network loads in which the employed GBA model (i.e., GBA with gated service) can no longer operate due to the fact that some ONUs cannot sleep for as long as all other ONUs are active. As observed, with high delay requirements (e.g., at 150 ms which is the typical voice delay requirement, and higher), the proposed model can achieve maximum energy saving under all network loads, and at the same time maintain the desired QoS performance. Conversely for lower delay requirements and under different scenarios, GBA's operation is bounded by certain network load thresholds. For example, for $\rho > 0.8$, this means that all the loads from 0.8 \rightarrow 1.0 (or higher) are able to break Eq. (10). At these thresholds, as mentioned earlier, the OLT can either put the PON system into fully active mode, or it may employ a different scheduling discipline to compute the ONU sleep time so as to achieve energy saving without compromising the operation of the network.

TABLE V: Network loads that break Eq. (10).

Scenario	Maximum Delay Requirement (\mathbb{D}_c)			
	50 ms	100 ms	150 ms	175 ms
1	$\rho > 0.8$	$\rho > 0.9$	N/A	N/A
2	$\rho > 0.6$	$\rho > 0.8$	N/A	N/A
3	$\rho > 0.8$	$\rho > 0.9$	N/A	N/A
4	$\rho > 0.6$	$\rho > 0.8$	N/A	N/A

VI. CONCLUSION

In this paper, we propose a new sleep time sizing and scheduling framework for the operation of green bandwidth allocation (GBA) in TDMA-PONs. The ONU sleep time is computed based on the maximum delay requirement of each CoS, via a new analytical model. We then address the collision problem that emerges from assigning different sleep time values to the ONUs which are expected to access the shared media directly upon waking up. A new sleep time sizing and scheduling mechanism, namely Sort-And-Shift, is proposed to effectively synchronize the sleep times of all ONUs without compromising their total assigned sleep time values so as to save maximum energy. Our results validate the presented model, and show how the proposed framework can leverage the sleep mode functionality at the ONUs to achieve maximum energy saving without impairing the users' requirements.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their constructive comments, which considerably improved the quality of the paper. This work was partially supported by the Natural Sciences and Engineering Research Council of Canada under the award NSERC-PDF-419675-2012.

REFERENCES

- [1] J.-I. Kani, F. Bourgart, A. Cui, A. Rafel, and S. Rodrigues, "Next-Generation PON—Part 1: Technology Roadmap and General Requirements," *IEEE Communications Magazine*, vol. 47, no. 11, pp. 43–49, Nov. 2009.
- [2] J. Baliga, R. Ayre, K. Hinton, W. V. Sorin, and R. S. Tucker, "Energy Consumption in Optical IP Networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 27, no. 13, pp. 2391–2403, July 2009.
- [3] R. Kubo, J.-I. Kani, H. Ujikawa, T. Sakamoto, Y. Fujimoto, N. Yoshimoto, and H. Hadama, "Study and Demonstration of Sleep and Adaptive Link Rate Control Mechanisms for Energy Efficient 10G-EPON," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 2, no. 9, pp. 716–729, Sep. 2010.
- [4] "IEEE 802.3az Energy Efficient Ethernet Task Force." [Online]. Available: <http://www.ieee802.org/3/az/index.html>
- [5] ITU-T, "G.sup45: GPON power conservation." [Online]. Available: <http://www.itu.int/rec/T-REC-G.Sup45-200905-1/en>
- [6] J. Mandin, "EPON Power Saving via Sleep Mode." [Online]. Available: http://www.ieee802.org/3/av/public/2008_09/3av_0809_mandin_4.pdf
- [7] S. Wong, L. Valcarenghi, S. Yen, D. R. Campelo, S. Yamashita, and L. Kazovsky, "Sleep Mode for Energy Saving PONs: Advantages and Drawbacks," in *Proceedings of IEEE GLOBECOM'09*, Hawaii, USA, Dec. 2009.
- [8] B. Skubic and D. Hood, "Evaluation of ONU Power Saving Modes for Gigabit-Capable Passive Optical Networks," *IEEE Network*, vol. 25, no. 2, pp. 20–24, Mar./Apr. 2011.
- [9] S. S. W. Lee and A. Chen, "Design and Analysis of a Novel Energy Efficient Ethernet Passive Optical Network," in *Proceedings of ICN'10*, French Alps, France, April 2010.

- [10] Y. Yan, S.-W. Wong, L. Valcarenghi, S.-H. Yen, D. R. Campelo, S. Yamashita, L. Kazovsky, and L. Dittmann, "Energy Management Mechanism for Ethernet Passive Optical Networks (EPONs)," in *Proceedings of IEEE International Conference on Communications (ICC'10)*, Cape Town, South Africa, May 2010.
- [11] J. Zhang and N. Ansari, "Toward Energy-Efficient 1G-EPON and 10G-EPON with Sleep-Aware MAC Control and Scheduling," *IEEE Communications Magazine*, vol. 49, no. 2, pp. S33–S38, Feb. 2011.
- [12] A. R. Dhaini, P.-H. Ho, and G. Shen, "Toward Green Next-Generation Passive Optical Networks," *IEEE Communications Magazine*, vol. 29, no. 11, pp. 94–101, Nov. 2011.
- [13] H. Takagi, *Analysis of Polling Systems*. MIT Press, April 1986.
- [14] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Prentice-Hall, 1992.
- [15] X. Bai, A. Shami, and Y. Ye, "Delay Analysis of Ethernet Passive Optical Networks with Quasi-leaved Polling and Gated Service Scheme," in *Proceedings of AccessNets'07*, Ottawa, ON, Canada, Aug. 2007.
- [16] F. Aurzada, M. Scheutzw, M. Herzog, M. Maier, and M. Reisslein, "Delay Analysis of Ethernet Passive Optical Networks With Gated Service," *Journal of Optical Networking*, vol. 7, no. 1, pp. 25–41, Dec. 2007.
- [17] S. Bharati and P. Saengudomlert, "Analysis of Mean Packet Delay for Dynamic Bandwidth Allocation Algorithms in EPONs," *IEEE/OSA Journal of Lightwave Technology*, vol. 28, no. 23, pp. 3454–3462, Dec. 2010.
- [18] Mindspeed, "Using Mindspeed Laser Drivers in PON burst-Mode Applications." [Online]. Available: <http://www.mindspeed.com/web/download/download.jsp?docId=24461>
- [19] G. Kramer, B. Mukherjee, S. Dixit, Y. Ye, and R. Hirth, "Supporting Differentiated Classes of Service in Ethernet Passive Optical Networks," *OSA Journal of Optical Networking*, vol. 1, no. 8/9, pp. 280–298, Aug. 2002.
- [20] A. R. Dhaini, P.-H. Ho, and X. Jiang, "QoS Control for Guaranteed Service Bundles over Fiber-Wireless Broadband Access Networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 29, no. 10, pp. 1500–1513, May 2011.
- [21] S. Bhatia, D. Garbuzov, and R. Bartos, "Analysis of the Gated IPACT Scheme for EPONs," in *Proceedings of IEEE International Conference on Communications (ICC'06)*, Istanbul, Turkey, 2006, pp. 2693 – 2698.
- [22] A. R. Dhaini and P.-H. Ho, "MC-FiWiBAN: An Emergency-Aware Mission-Critical Fiber-Wireless Broadband Access Network," *IEEE Communications Magazine*, vol. 49, no. 1, pp. 134–142, Jan. 2011.
- [23] O. J. Boxma and W. P. Groenendijk, "Pseudo-Conservation Laws in Cyclic-Service Systems," *Journal of Applied Probability*, vol. 24, no. 4, pp. 949–964, Dec. 1987.
- [24] IEEE 802.3ah Task Force Home Page. [Online]. Available: <http://www.ieee802.org/3/efm>
- [25] "ITU-T Recommendation G.114: One-Way Transmission Time in Series G: Transmission Systems and Media, Digital Systems and Networks." [Online]. Available: <http://www.itu.int/rec/T-REC-G.114-200305-I/en>



Ahmad R. Dhaini received his B.Sc. in Computer Science from the American University of Beirut (AUB), Lebanon in 2004; his M.App.Sc. in Electrical and Computer Engineering from Concordia University, Montreal, Canada, with a best thesis award nomination in 2006; and his Ph.D. degree in Electrical and Computer Engineering from University of Waterloo, Canada in 2011.

Dr. Dhaini is the winner of the prestigious Natural Sciences and Engineering Research Council of Canada Postdoctoral Fellowship (NSERC PDF), the

Ontario Graduate Scholarship in Science and Technology (OGSST), and various teaching and research awards at University of Waterloo.

He is currently a Postdoctoral Research Fellow at Stanford University, working in the Photonics and Networking Research Laboratory (PNRL). In 2011–2012, he was a Research Associate at University of Waterloo. He worked as a Software Consultant at TEKSystems, Montreal, in 2006–2007; and as a Software Designer at Ericsson, Montreal, in 2007–2008.

Dr. Dhaini has authored a book, and authored/co-authored more than 15 papers in major journals and conferences related to broadband access networks. His research interests focus on access networks; more specifically on fiber-wireless (FiWi) broadband access networks and green communications.



Pin-Han Ho received the B.Sc. and M.Sc. degrees at the Electrical Engineering Department, National Taiwan University, in 1993 and 1995, respectively, and the Ph.D. degree from Queens University, Kingston, Canada, in 2002.

He is currently an Associate Professor in the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He is the author/coauthor of more than 150 refereed technical papers, several book chapters, and the coauthor of a book on optical networking and survivability.

His current research interests cover a wide range of topics in broadband wired and wireless communication networks, including survivable network design, fiber-wireless network integration, and network security.

Dr. Ho is the recipient of the Distinguished Research Excellent Award in the Electrical and Computer Engineering Department of University of Waterloo, Early Researcher Award (Premier Research Excellence Award) in 2005, the Best Paper Award in International Symposium on Performance Evaluation of Computer and Telecommunication Systems in 2002, International Conference on Communication Optical Networking Symposium in 2005, and International Conference on Communication Security and Wireless Communications symposium in 2007, and the Outstanding Paper Award in International Conference on High Performance Switching and Routing in 2002.



Gangxiang Shen is a Distinguished Professor with Optical Network Technology Research Center, School of Electronic and Information Engineering, Soochow University. From October 2008 to June 2010, he was with Ciena Corporation, Baltimore, MD as a Lead Engineer.

Before he joined Soochow University, he was with ARC Special Research Centre for Ultra-Broadband Information Networks, Department of Electrical Engineering, University of Melbourne, Australia, as an Australian Postdoctoral Fellow (APD) and ARC Research Fellow. He was also a Research Scientist (NSERC Industrial R&D Fellow) with Optiwave Systems Inc. in Canada in 2006. He received a PhD from Department of Electrical and Computer Engineering, University of Alberta, Canada in January 2006. He received an MSc from Nanyang Technological University of Singapore in 1999 and BEng from Zhejiang University in P. R. China in 1997.

His major research interests are in Optical Communication Networks and Green Internet. He has authored and co-authored more than 45 technical papers and/or book chapters. He is also in the editorial board of Elsevier Journal of Optical Switching and Networking and TPC members of many international conferences such as Infocom, Globecom, ICC, and so on.



Basem Shihada Dr. Basem Shihada has been appointed Assistant Professor of Computer Science in the Mathematical and Computer Science and Engineering Division at KAUST. He joined Stanford University as a visiting faculty in Stanford Computer Science in 2008. Prior to his appointment at KAUST, Dr. Shihada was a research associate in the Department of Electrical and Computer Engineering at the University of Waterloo in Ontario, Canada.

Dr. Shihada has served on the review process of several international journals and conferences. In 2003, he was responsible for designing the user-access layer and the grid-service-provisioning layer in the joint Canarie, Canada Research Center and Cisco Canada, research and development project entitled, "User controlled lightPaths". Dr. Shihada's current research covers a wide range of topics in broadband wired and wireless communication networks, including wireless Metropolitan Area Networks such as IEEE 802.16 networks, Fiber-Wireless (FIWI) network integration, and optical networks, and green communications.

In 1997, he obtained a Bachelor's degree in Computer Science from the UAE University. In 2001, he obtained a Master's degree in Computer Science from Dalhousie University, Halifax, Canada. In 2007, he obtained a PhD degree in Computer Science, University of Waterloo, David R. Cheriton School of Computer Science, Waterloo, Canada.