Building an Energy-efficient Uplink and Downlink Delay Aware TDM-PON System

S.H. Shah Newaz[a,*], Min Seok Jang[b], Alaelddin Fuad Yousif Mohammed[a], Gyu Myoung Lee[c], Jun Kyun Choi[a]

[a]*Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea*
[b]*Software division, Humax, Seong-nam, South Korea*
[c] *Liverpool John Moores University (LJMU), Liverpool, United Kingdom*

Abstract

With the increasing concern over the energy expenditure due to rapid ICT expansion and growth of Internet traffic volume, there is a growing trend towards developing energy-efficient ICT solutions. Passive Optical Network (PON), which is regarded as a key enabler to facilitate high speed broadband connection to individual subscribers, is considered as one of the energy-efficient access network technologies. However, an immense amount of research effort can be noticed in academia and industries to make PON more energy-efficient. In this paper, we aim at improving energy saving performance of Time Division Multiplexing (TDM)-PON, which is the most widely deployed PON technology throughout the world. A commonly used approach to make TDM-PON energy-efficient is to use sleep mode in Optical Network Units (ONUs), which are the customer premises equipment of a TDM-PON system. However, there is a strong trade-off relationship between traffic delay performance of an ONU and its energy saving (the longer the sleep interval length of an ONU, the lower its energy consumption, but the higher the traffic delay, and vice versa). In this paper, we propose an Energy-efficient Uplink and Downlink Delay Aware (EUDDA) scheme for TDM-PON system. Prime object of EUDDA is to meet both downlink and uplink traffic delay requirement while maximizing energy saving performance of ONUs as much as possible. In EUDDA, traffic delay requirement is given more priority over energy saving. Even so, it still can improve energy saving of ONUs noticeably. We evaluate performance of EUDDA in front of two existing solutions in terms of traffic delay, jitter, and ONU energy consumption. The performance results show that EUDDA significantly outperforms the other existing solutions.

Keywords:
TDM-PON, Sleep mode, Tx sleep, Rx sleep, Traffic delay, Jitter, Energy consumption.

---

1. Introduction

Recent studies (e.g. [1, 2]) reveal that there is a stupendous growth of CO_2 footprint due to rapid expansion of ICT. It is worth noticing that currently ICT is responsible for consuming 8% of total electricity

*Corresponding author: Dept. of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. Tel.: +82-42-350-8059; Fax: +82-42-350-7140.
E-mail address: newaz@kaist.ac.kr (S.H.S. Newaz).

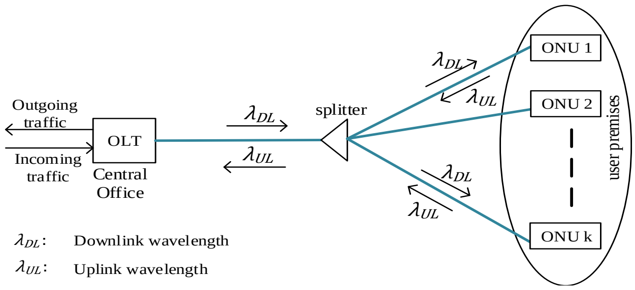


Fig. 1: Generic TDM-PON Architecture.

consumption in the world [3]. This has triggered many research initiatives which have led to develop energy-efficient protocols and hardware development in order to minimize energy consumption of network equipment. Researchers from both industries and academia have centered their efforts on maximizing energy-efficiency of core and access network equipment. Recently, one of the most interesting findings associated with network utilization is presented in [2]. Authors in [2] impart that around 70% of overall Internet energy consumption is consumed by access network equipment. However, utilization of these equipment is only around 15% [4, 5]. Consequently, in recent years, there has been an increasing interest in developing energy-efficient protocol and hardware design for access network equipment.

To date, there are several access network technologies, such as Passive Optical Networks (PON), Worldwide Interoperability for Microwave Access (WiMAX), Wireless Fidelity (WiFi) and Digital Subscriber Line (DSL). Among these access network technologies, PON is considered as a very promising technology. This is because PON provides not only huge data rate (up to the order of Gbps) but also it consumes significantly less energy compared to other technologies like WiMAX [6, 7]. Among different PON architectures, such as Wavelength-Division Multiplexing-PON (WDM-PON), Time Division Multiplexing-PON (TDM-PON) and Hybrid WDM/TDM-PON, TDM-PON has been widely deployed in many countries (e.g. China, Korea, and Taiwan).

A generic TDM-PON (e.g. Ethernet PON (EPON) and Gigabit-capable PON (GPON)) is composed of an Optical Line Terminal (OLT) (centralized intelligence of a TDM-PON), a passive splitter and several Optical Network Units (ONUs), which are installed at user premises, as depicted in Fig. 1. The splitting ratio of a passive splitter in a TDM-PON is termed as *1:n*, where *n* can be 16, 32, 64 or 128 ONUs. A TDM-PON architecture employs a single wavelength in the downlink direction (OLT to the ONUs), whereas a separate wavelength is deployed for uplink communication (ONUs to the OLT). In a TDM-PON, downlink is broadcast-and-select mechanism. The OLT marks the downlink traffic with a unique identifier, so that each of the ONUs can identify its corresponding traffic (e.g. in EPON, an OLT uses a unique Logical Link Identifier (LLID) [8]). On the other hand, the OLT controls the uplink transmission of all connected ONUs by assigning grant time in time domain for each ONU to send its uplink traffic. In this manner, a TDM-PON can secure that an uplink traffic conflict will not occur. Thereby, an ONU sends bandwidth request message to inform the OLT its uplink bandwidth requirement. The OLT collects all bandwidth requests from its connected ONUs and uses Dynamic Bandwidth Allocation (DBA) algorithm to measure upstream transmission slots for each of the ONUs. After calculating transmission slots, the OLT notifies the ONUs.

Although PON is more energy-efficient compared to other access technologies, there is a big room to improve its energy saving performance while meeting Quality of Service (QoS) of its traffic [6, 7]. Findings in [9] reveal that ONUs are responsible for 65% power consumption of a PON system. The earlier standards (e.g. IEEE 802.3ah [8]) consider that ONUs should be kept always on. It is because in the downlink direction of a TDM-PON is true broadcast. An ONU never knows when the OLT will have traffic to send. Consequently, an ONU needs to be always on to receive downlink traffic, thus wasting energy unnecessarily. Researchers have pointed out this limitation of the earlier TDM-PON standards and they have come up with sleep mode mechanism and low-power-consuming optical transceivers for ONUs in order to reduce energy consumption.

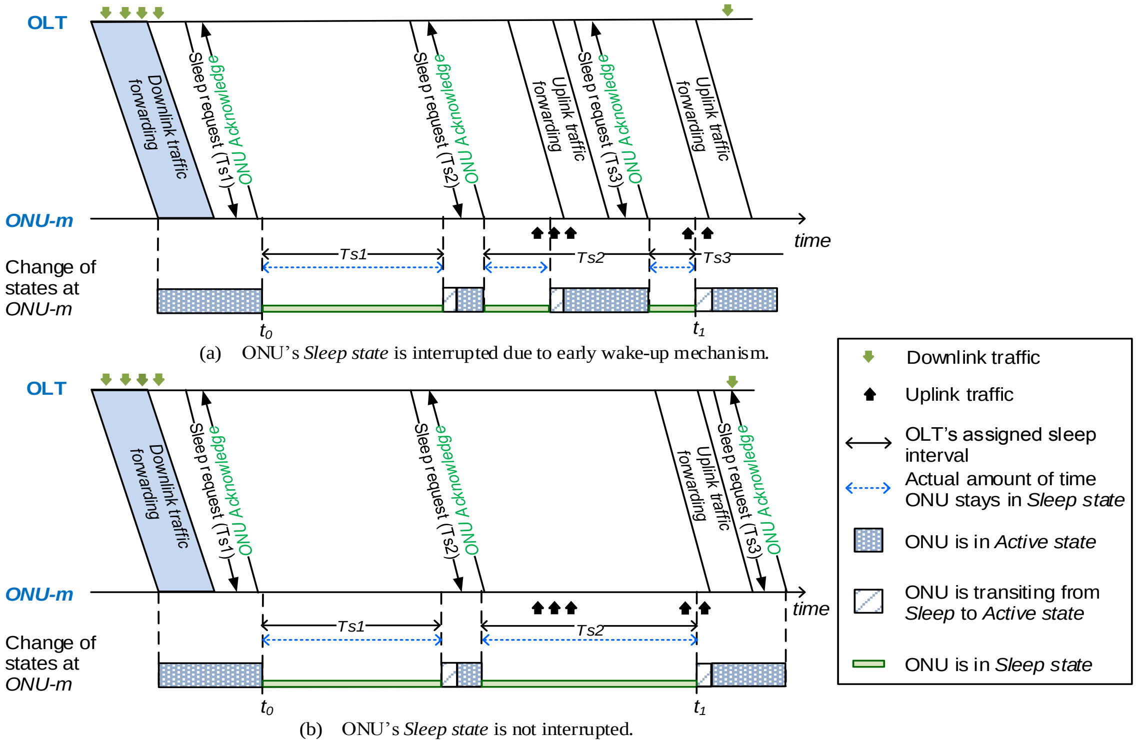
One of the widely applied approaches used for ONUs of TDM-PONs to improve energy saving performance is sleep mode in which an ONU switches off its power hungry components during a defined amount of time [7, 10]. Although sleep mode is an efficient approach to design an energy-efficient TDM-PON, it could affect traffic delay performance significantly if sleep interval lengths of ONUs are not carefully decided. In fact, an ONU's sleep interval length brings a trade off relationship: the longer the sleep interval length of an ONU, the less energy it consumes, but the higher the traffic delay, and vice versa [7]. Therefore, to date, many researchers have centered their efforts in developing sleep mode deciding algorithms (e.g. [6, 7, 11–15]). Energy saving in TDM-PONs has gained attention of different standardization bodies. For example, ITU-T G.sup 45 [16] has introduced four different power saving techniques for an ONU: power shedding, doze mode, deep sleep mode and fast sleep (these four power saving techniques are briefly explained in Section 2).

When an ONU in a TDM-PON uses sleep mode to improve its energy saving performance, generally, the OLT is in charge of deciding sleep interval lengths of the ONU. An ONU supporting sleep mode can have several states (e.g. *Sleep state*, *Active state*). The OLT calculates a sleep interval length of an ONU using a sleep interval length deciding algorithm and notifies the ONU in absence of traffic [7, 11, 13, 14]. The ONU leaves *Sleep state* after the assigned sleep interval period expiration and waits for the OLT's further instruction. The OLT invokes the ONU to stay active if there is any frame to receive and/or transmit. Otherwise, the OLT instructs the ONU to move into *Sleep state* mentioning next sleep interval length. It needs to mention here that whenever an ONU moves into *Sleep state* it loses OLT's clock and synchronization [10]. Therefore, after completion of *Sleep state*, to establish communication with the OLT, the ONU needs to spend around 2 *ms* to gain OLT's clock and synchronization [11, 13, 14]. In this paper, the time required for gaining OLT's clock and synchronization is referred as transition time. The power consumption of an ONU during the transition time is almost the same as in the *Active state* [17].

Most of the sleep mode deciding algorithms (e.g. [5, 7, 12–14]) take into consideration only the presence or absence of downlink traffic while deciding sleep interval lengths for ONUs. In those solutions, authors consider that a sleeping ONU should leave *Sleep state* whenever it is interrupted due to uplink traffic arrival. Similarly, energy-efficiency aware TDM-PON standards (e.g. Service Interoperability in Ethernet Passive Optical Networks (SIEPON) IEEE 1904.1 [18] and ITU-T G.988 [19]) recommend that ONU's *Sleep state* should be interrupted on arrival of uplink traffic (sleeping ONU should leave *Sleep state* upon arrival of uplink traffic). Leaving *Sleep state* before the allocated sleep interval period due to uplink traffic arrival is referred to as early wake-up in [16, 18, 19]. The most important limitation of the solutions that consider ONU early wake-up lies in the fact that, during high uplink traffic arrival scenario, an ONU will not be able to complete the OLT's assigned sleep interval length, and thus an ONU will end up spending significant amount of energy for *Sleep state* to *Active state* transition. To solve this problem, some researchers suggest in their solutions that uplink and downlink traffic forwarding should take place at the same time (e.g. [20, 21]). However, these solutions suffer from one or more major drawbacks that we explain in Section 2.

Figure 2 provides an explanation to understand the downside of early wake-up. Based on aforementioned discussion, this figure compares two cases: (i) an ONU uses early wake-up and leaves *Sleep state* on uplink traffic arrival (Fig. 2(a)) and (ii) an ONU leaves *Sleep state* after completing OLT's assigned sleep interval length, regardless the presence or absence of uplink traffic (Fig. 2(b)). It is interesting to observe from Fig. 2(a) that between the time t_0 and time t_1 the number of times a particular *ONU-m* transits from *Sleep state* to *Active state* and the number of control messages exchanged between the OLT and *ONU-m* are less than that of the second case depicted in Fig. 2(b).

In this paper, we propose Energy-efficient Uplink and Downlink Delay Aware (EUDDA) scheme for TDM-PON system. Similar to [3, 7], we believe that the main goal of a TDM-PON operator should be meeting the PON traffic delay requirement first, and then reduce PON energy consumption as much as possible. Keeping this into consideration, in this paper, we come up with a novel algorithm that favors meeting delay requirement of both uplink and downlink traffic, while saving energy of a TDM-PON as much as possible. In particular, here both uplink and downlink traffic delay requirements are taken into account while deciding on ONUs' sleep mode associated parameters. Therefore, unlike an ONU in [5, 7, 12–14], early wake-up is not required for an ONU in our EUDDA scheme. Consequently, this leads to reduce the number of *Sleep state* to *Active state* transitions, and control messages exchanged between the OLT and an



(a) ONU's *Sleep state* is interrupted due to early wake-up mechanism.

(b) ONU's *Sleep state* is not interrupted.

Fig. 2: Understanding influence of early wake-up mechanism.

ONU in our solution. Furthermore, in this paper, we propose a novel sleep mode associated control message exchange approach between the OLT and ONUs to reduce further energy consumption in a TDM-PON.

We use 24 h real network traffic traces to evaluate the performance of EUDDA. Performance results are compared in front of two existing solutions. Results show that EUDDA outperforms two existing solutions in terms of frame delay, jitter and ONU energy saving performance.

The remainder of this paper is organized as follows. In Section 2, we provide a brief background of TDM-PON and related work associated with TDM-PON energy-efficiency improvement. Section 3 presents our EUDDA scheme. In Section 4, we compare the performance of EUDDA in front of other existing solutions. Finally, we conclude this work in Section 5.

2. Related Work

In this section, we review existing proposals of sleep mode management in TDM-PONs.

Earlier standards of TDM-PONs (e.g. IEEE 802.3ah [8]) do not take into consideration energy-efficiency issue. However, over the last several years, energy saving in ICT has become a significantly important research issue due to stupendously large contribution of ICT in increasing CO_2 footprint in the globe. In PON research area, there has been a considerable research effort (e.g. [7, 12–14]) devoted to improving energy-efficiency of the PON equipment (e.g. OLT, ONU). It is worth noting that, to date, most of the research objectives is to minimize energy consumption in ONUs. It is because ONUs contribute in consuming 65% of total energy consumption of a TDM-PON [9]. One of the most common approaches for maximizing energy saving in ONUs is sleep mode.

Four different power saving techniques for an ONU have been introduced by ITU-T G.sup 45 [16]. These are: power shedding, doze, deep sleep mode and fast sleep mode. Power saving techniques of ITU-T G.sup 45 are summarized below:

- Power shedding: an ONU powers off non-essential activities or reduces supplied power in order to reduce its energy consumption. In this case, optical link remains fully operational.
- Doze mode: an ONU switches off its transmitter (Tx), while keeping its receiver (Rx) on. This allows the ONU to receive traffic from the OLT whenever the OLT has traffic destined to that ONU. This power saving technique can be used when an ONU does not have any uplink traffic to forward.
- Deep sleep mode: an ONU turns off both of its Tx and Rx. When an ONU is in deep sleep mode, the OLT does not have any chance to communicate with the ONU.
- Fast sleep: when an ONU uses fast sleep, the Tx and Rx of an ONU are turned off periodically during a certain amount time, which is defined by the OLT. This technique is also termed as cyclic sleep.

SIEPON (Std. IEEE P1904.1) defines power saving mechanism for ONUs [18]. This standard also specifies control messages that should be exchanged between the OLT and an ONU to facilitate sleep mode management. In particular, two sleep mode management techniques have been defined in this standard: Tx sleep mode, in which Tx of an ONU is turned off, and TRx sleep mode, in which an ONU switches off both of its Tx and Rx for a defined amount of time. Tx sleep and TRx sleep mode in SIEPON are the same as the Doze and Fast sleep mode defined by ITU-T in [16, 19], respectively.

Most of the solutions (e.g. [5–7, 13, 14]) devoted to improving energy saving in ONUs using sleep mode take into consideration downlink traffic only while deciding on a sleep interval length of an ONU. These solutions consider that a sleeping ONU should execute early wake-up mechanism on arrival of uplink traffic from user premises. Authors in [5–7, 13, 22] consider that the OLT reserves a small amount of bandwidth for the sleeping ONUs, so that all sleeping ONUs can have opportunity to make a bandwidth request to the OLT on arrival of uplink traffic. The major limitation of early wake-up mechanism is that, during high traffic arrival (e.g. peak hours of a day) a sleeping ONU would be forced to leave *Sleep state* frequently, thereby spending significant amount of energy for the transition from *Sleep state* to *Active state* [22]. It is worth noting that an ONU takes around 2 *ms* to transit from *Sleep state* to *Active state* [11, 13, 14] and power consumption of the ONU during this period is almost the same as in the *Active state* [17].

To avoid interruption while an ONU in *Sleep state*, a number of research efforts have aimed at developing novel sleep mode management mechanism for ONUs taking into consideration both uplink and downlink traffic (e.g. [20, 21]). However, these studies fail to explain how sleep interval length can be dynamically defined for different types of traffic delay requirement (strict delay requirement traffic (e.g. voice traffic) and relaxed delay requirement traffic (e.g. HTTP traffic)).

Authors in [3] propose a potentially important algorithm to decide sleep interval length for an ONU. However, their proposed algorithm decides sleep intervals for an ONU considering the QoS requirement of high priority traffic. According to the solution, if an ONU queue does not have high priority traffic, the OLT allows the ONU to sleep for maximum 10 *ms* (maximum sleep interval length is 10 *ms*). It is important to note here that traffic arrival of an ONU is very low during off-peak hours of a day [7]. Therefore, an ONU with 10 *ms* sleep interval length might have no traffic to send and/or receive after waking up from *Sleep state*, consequently the ONU will end up uselessly expending energy during off-peak hours of a day for idle listening[1].

Authors in [23] propose batch-mode based transmission mechanism to reduce number of *Sleep state* to *Active state* transition period. Authors in [24] introduce Upstream Centric Scheduling (UCS) scheme. According to their proposal, the OLT in a TDM-PON forwards traffic during uplink transmission of an ONU. The major limitation of this solution is that if the downlink traffic arrival for an ONU is more than its uplink traffic arrival, downlink traffic experience delay noticeably. One major limitation of the solutions proposed in [23, 24] is high control message overhead associated with sleep mode management and bandwidth allocation for both uplink and downlink transmission for an ONU. Authors in [25] refer to the shortcoming

[1]Idle listening refers to the cases in which an ONU leaves *Sleep state* to listen OLT's instruction after turning on its power hungry components; however, the OLT does not have any traffic to forward to the ONU. Therefore, an ONU ends up spending energy unnecessarily.

of control message overhead in energy-efficiency performance of a TDM-PON. Their findings reveal that control message overhead associated with sleep mode management and bandwidth allocation can noticeably reduce energy-efficiency performance of an ONU in a TDM-PON.

According to the solutions presented in [13, 14], in absence of downlink traffic of an ONU, the OLT computes sleep interval length for an ONU and notifies it. After receiving sleep request from the OLT, the ONU sends an acknowledgment message and moves into sleep mode turning off its power hungry components. In these solutions, whenever an ONU wakes up from *Sleep state*, the OLT and ONU exchange control message associated with sleep mode management. Authors in [13, 14], mention that throughput of both downlink and uplink can reduce due to sleep mode associated control message exchange if the OLT has large number of ONUs to serve. Additionally, due to both way control message exchange after an ONU wakes up from *Sleep state*, the ONU needs to remain both of its Tx and Rx powered on, thus expending significant amount of energy for sleep mode associated control message exchange. As we will see later that there are other solutions (e.g. [6, 7]) that do not force an ONU to exchange control messages with the OLT every time the ONU wakes up from *Sleep state*.

Authors in [6] propose that when an ONU does not receive traffic from the OLT in a TDM-PON during a predefined period of time, it should move into sleep mode assuming that the OLT does not have any traffic for it. After a certain amount of time, the sleeping ONU leaves *Sleep state* to check presence or absence of downlink traffic arrival. The ONU uses an algorithm to calculate its sleep interval lengths. The same algorithm is also used at the OLT. Therefore, the OLT is always aware of the status (*Sleep state*, *Active state*) of the ONU. According to the solution proposed in [7], in absence to downlink traffic of an ONU, the OLT quantifies lower bound (T_{min}) and upper bound (T_{max}) of sleep interval of an ONU, and notifies to the ONU. Every time the ONU wakes up from *Sleep state* and finds that the OLT does not have traffic to forward, it uses the T_{min} and T_{max} to calculate its sleep interval lengths using an algorithm (see Eq. (1)). Similar to [6], in [7], the OLT uses the same algorithm that an ONU uses to determine the ONU's status. Therefore, the OLT always knows when the ONU is available to check presence of downlink traffic. By doing so, these solutions can reduce number of control messages exchange between the OLT and ONUs, thus reducing energy consumption. However, in these solutions, only downlink traffic arrival has been taken into consideration while deciding sleep interval lengths for an ONU (rely on early wake-up mechanism when a sleeping ONU receives an uplink traffic).

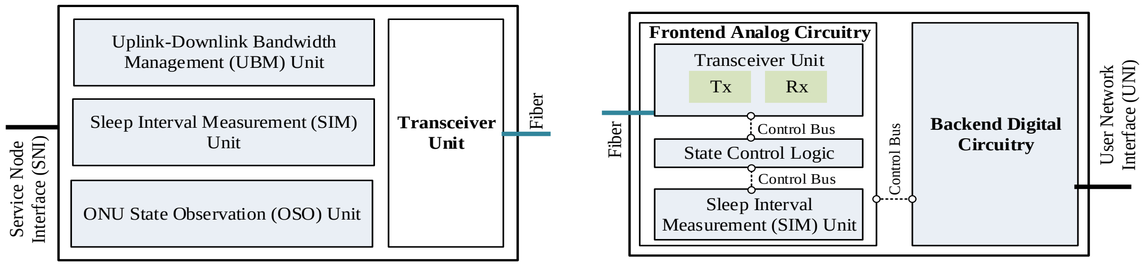
3. Proposed Energy-efficient Uplink and Downlink Delay Aware (EUDDA) Scheme

EUDDA is devised to meet downlink and uplink traffic delay requirement and minimize ONU's energy consumption as much as possible. As mentioned earlier, there is a strong trade-off relationship between ONUs' energy saving and traffic delay performance. Therefore, finding a suitable sleep interval length for an ONU is a challenging task indeed. In this section, we present our novel EUDDA scheme, which has mainly two objectives: (i) meeting traffic QoS in terms of TDM-PON traffic delay requirement, and (ii) improving energy saving performance of ONUs as much as possible.

It is important to note here that EUDDA treats a strict and relaxed delay requirement traffic differently. For strict delay requirement traffic scenario, EUDDA aims at satisfying delay requirement of 100% traffic of an ONU. To meet this objective, in our proposal, the OLT and an ONU takes part in deciding a fixed sleep interval length. In case of a relaxed delay requirement traffic scenario, there is a big room to maximize ONU's energy saving performance. In this particular case, our scheme favors saving energy as much as possible while satisfying delay requirement of traffic. In EUDDA, an ONU having relaxed delay requirement traffic uses Exponentially Increment Sleep Interval Management Policy (EI-SIMP), in which an ONU restricts its sleep interval lengths within a lower bound (T_{min}) and upper bound (T_{max}) while it is in sleep mode (we will explain in the subsequent part of this section how both the OLT and ONU take part in finding value of T_{min} and T_{max} in EUDDA).

3.1. EUDDA System Model

In this part, we explain our operational assumption associated with OLT and ONU in EUDDA. Similar to [7], we suppose that a TDM-PON operator assigns traffic delay requirement for each of the ONUs in



(a) OLT block diagram.

(b) ONU block diagram.

Fig. 3: OLT and ONU block diagram in EUDDA.

our solution. Figure 3 represents an OLT and an ONU functional block diagram. First, we briefly explain functional block diagram of an OLT and ONU in EUDDA. Then, we describe the different states that an ONU can have in EUDDA.

3.1.1. Functional Block Diagram of OLT in EUDDA

Here, we assume that the OLT performs all the functionality of a traditional OLT; for example, uplink bandwidth allocation and downlink traffic forwarding. For EUDDA scheme, similar to [6, 7, 12], we assume the following main functional units for the OLT:

- Uplink-downlink Bandwidth Management (UBM) Unit: The OLT receives traffic from core network through its Service Network Interface (SNI) and forwards them if the destination ONU is available for communication. If the destination ONU is in *Sleep state*, in which an ONU powers off its Tx and Rx, the OLT should buffer those traffic. In addition, we assume that the OLT allocates bandwidth for different ONUs taking into consideration their bandwidth requirement and traffic priority.

- Sleep Interval Management (SIM) Unit: SIM unit is in charge of maintaining three important activities associated with ONUs' sleep mode: (i) calculating a fixed sleep interval length for an ONU having strict delay requirement traffic, (ii) measuring a value of T_{min} and T_{max} for an ONU having relaxed delay requirement traffic, and (iii) taking part with an ONU to select appropriate sleep mode associated parameters (e.g. T_{max}, T_{min}) (we explain this in details in the subsequent part of this section).

- ONU State Observation (OSO) Unit: OSO unit is responsible for keeping record of the states of different ONUs and exchanging sleep mode associated control messages with the ONUs.

3.1.2. Functional Block Diagram of ONU in EUDDA

We assume that an ONU in EUDDA can maintain sleep mode to maximize its energy saving, besides doing all the functionality that a traditional ONU does (e.g. uplink bandwidth request, traffic forwarding). An ONU's functional diagram can be separated into two parts (see Fig. 3(b)): frontend analog circuitry part and backend digital circuitry part [10]. To reduce power consumption in ONUs, we assume that Tx and Rx in frontend analog circuitry part of an ONU can be turned off selectively when they do not have any use, as considered in [6, 7, 12, 25]. To manage sleep mode in ONUs, we assume that an ONU includes a State Control Logic (SCL) and SIM unit. The role of the SIM unit in an ONU is the same as in the OLT. The role of the SCL unit is to select an appropriate state (e.g. *Sleep state*, *Active state*) for the ONU at a given time. We consider that when an ONU is in *Sleep state* turning off its Tx and Rx, all the incoming traffic received through UNI should be buffered inside the ONU.

3.1.3. ONU States

In our solution, similar to [6, 12, 25], we assume that an ONU has four states as listed below:

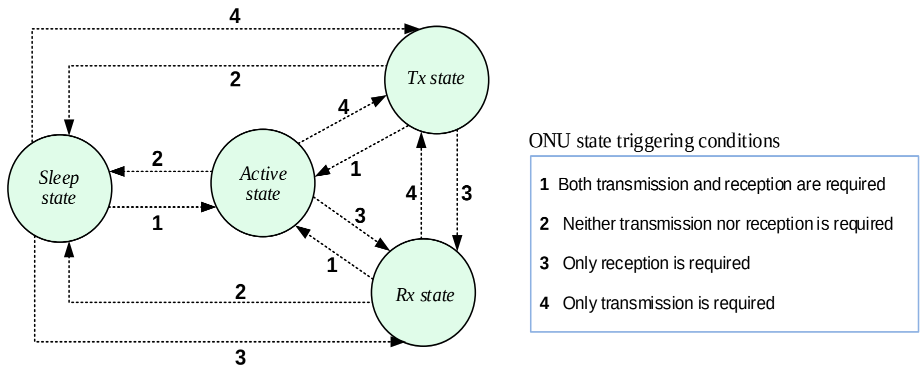


Fig. 4: ONU state transition diagram in EUDDA.

- ***Active state***: In this state, both Tx and Rx of an ONU is turned off, thus allowing the ONU to be fully functional. This state allows the ONU to perform both uplink and downlink communication. In this state, an ONU consumes 4.69 *W* [10, 25].
- **Transmission state (*Tx state*)**: This state refers to the period in which an ONU switches off its Rx, while keeping the Tx on. An ONU can transmit uplink traffic in this state and its power consumption is 2.99 *W* [25].
- **Reception state (*Rx state*)**: An ONU in this state switches off the Tx and keeps all other components on. In *Rx state*, an ONU can receive traffic from the OLT. Power consumption of the ONU in this state is 1.7 *W* [25].
- ***Sleep state***: This is the most energy saving state (power consumption is 0.7 *W* [10, 25]) since an ONU turns off its both Tx and Rx in this state. While an ONU is in *Sleep state*, it can neither transmit nor receive traffic.

Figure 4 presents ONU state transition diagram along with the conditions for different state transitions in EUDDA.

3.1.4. Algorithm for ONU's Sleep Interval Measurement

As mentioned earlier that our proposal has two different Sleep Interval Management Policies (SIMPs) for strict and relaxed delay requirement traffic. Here, we explain these two different policies.

- **Exponentially Incrementing SIMP (EI-SIMP)**: In this policy, the OLT and a particular ONU (e.g. *ONU-m*) find T_{min} and T_{max} using Delay Requirement Aware Parameters Selection (DRAPS) algorithm, which we will present in the subsequent part of this section. After deciding on the value of sleep mode associated parameters, which are T_{min} and T_{max} in EI-SIMP, the *ONU-m* uses Eq. (1) to calculate its *j*-th sleep interval length T_j. The OLT also uses Eq. (1) to know when the *ONU-m* is available for future communication.

$$T_j = \begin{cases} 2^{j-1}T_{\min} & , 2^{j-1}T_{\min} < T_{\max} \\ T_{\max} & , otherwise \end{cases} \tag{1}$$

  where, $j = \{1, 2, ..., n\}$.

- **Fixed Length SIMP (FL-SIMP)**: This is a policy in which a particular *ONU-m* and the OLT agree to use a fixed length sleep interval (T_{fix}). Therefore, T_{fix} is the only one sleep mode management related parameter in FL-SIMP case. Both the OLT and *ONU-m* use DRAPS algorithm to find T_{fix} for the *ONU-m*. The prime objective of FL-SIMP is to satisfy delay requirement of 100% of traffic having strict delay requirement of an ONU.

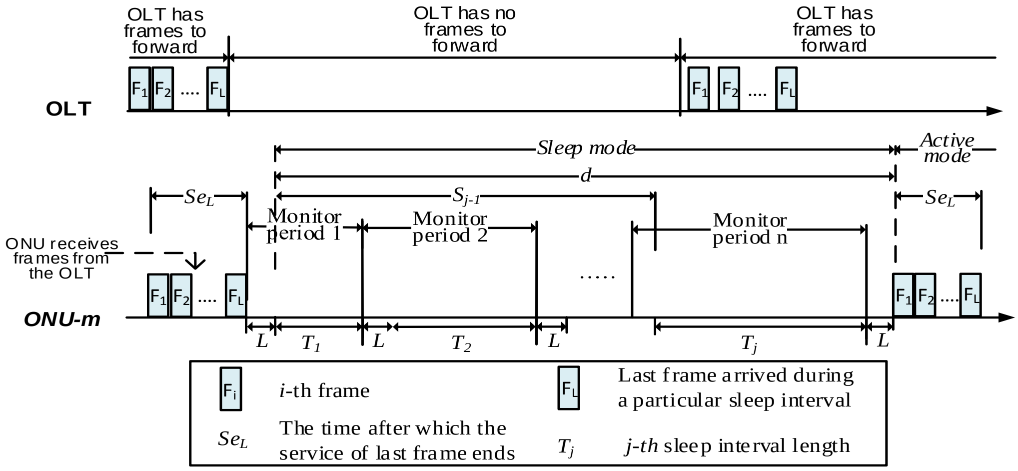


Fig. 5: Frame delay analysis in EI-SIMP.

3.2. *EUDDA's Algorithm for Strict and Relaxed Delay Requirement Traffic*

Here, we present EUDDA's DRAPS algorithm which takes into consideration traffic arrival rate (λ) and operator assigned delay requirement (D_R) to measure sleep mode associated parameters of an ONU. It is mentioned earlier that EUDDA treats a strict and relaxed delay requirement traffic differently. This is where EUDDA's DRAPS algorithm comes into play. Here, we first formulate to explain how value of T_{min} and T_{max} influence traffic delay. Then, we present DRAPS algorithm.

3.2.1. *Traffic Delay Estimation and Discussion for EI-SIMP*

In this subsection, traffic delay is quantified when T_{min}, T_{max}, and λ are given. Let L be the listening interval during which the OLT and an ONU wait for their mutual instruction (e.g. sleep management associated control message) and T_j is the length of j-th sleep interval. We assume that λ follows Poisson process similar to [7, 13, 25]. Then, sum of j-th sleep interval and listening interval is $T_j + L$. An ONU's sleep mode duration (d) can be composed of one or more sleep and listening intervals (see Fig. 5). Thus, the expression to compute total sleep and listening interval length up to T_{j-1} is thus expressed as follows:

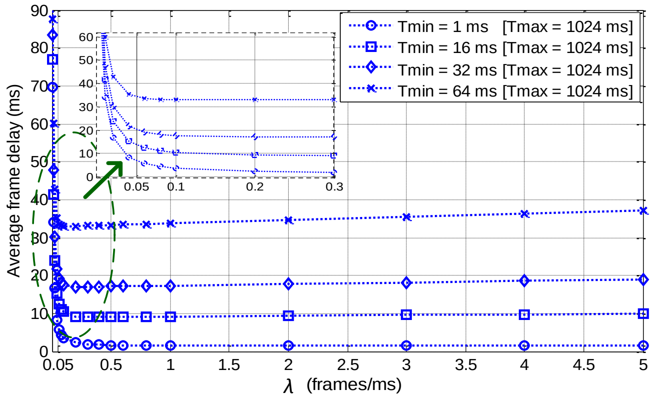
$$S_{j-1} = \sum_{i=1}^{j-1} (T_i + L). \tag{2}$$

Assume that e_j indicates an event in which the OLT receives one of more downlink frames for an *ONU-m* during the j-th monitor period.

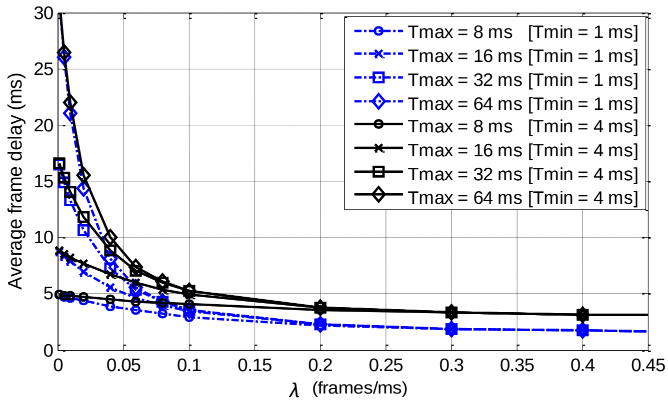
$$Pr(e_j = true) = 1 - e^{-\lambda(T_j+L)}. \tag{3}$$

Then the probability that the *ONU-m* will have *j-th* sleep interval is defined as [26]

$$\begin{aligned} &Pr(n = j) = Pr(\text{no frame is available during } S_{j-1}) \\ &Pr(\text{at least 1 available frame during the monitor period } j) \\ &= \prod_{i=1}^{j-1} Pr(e_i = false)Pr(e_j = true) \\ &= e^{-\lambda S_{j-1}} \left\{1 - e^{-\lambda(T_j+L)}\right\}. \end{aligned} \tag{4}$$



(a) Influences of T_{min} on frame delay.



(b) Influences of T_{max} on frame delay.

Fig. 6: Influence of the length of T_{min} and T_{max} on frame delay.

Let E[·] denotes average value. Then, frame delay for the *ONU-m* is expressed as follows [26]:

$$E[F] = \frac{\sum_{j=1}^{\infty} Pr(n = j)(T_j + L)}{2} = \frac{\sum_{j=1}^{\infty} e^{-\lambda S_{j-1}} \left\{1 - e^{-\lambda(T_j+L)}\right\}(T_j + L)}{2}. \tag{5}$$

The average amount of time the *ONU-m* spends in sleep mode is composed of one or more sleep and listening intervals (see Fig. 5) before moving into active mode is quantified as follows [26]:

$$E[d] = \sum_{j=1}^{\infty} Pr(n = j) \sum_{k=1}^{j} (T_k + L). \tag{6}$$

Average amount of time required to serve all the traffic arriving during *E[d]* for the *ONU-m* is calculated as follows:

$$Se_{L,\,EI} = \frac{E[d]\lambda F_s}{D_r}, \tag{7}$$

where, F_s is the average frame size and D_r is the TDM-PON link rate.

At this point, it is important to mention once again that one of the most important goals of EUDDA is to meet traffic delay requirement. Therefore, to calculate traffic delay for an ONU that uses sleep mode, we need to not only take into consideration TDM-PON frame delay due to sleep interval length but also the required service time of the frames that arrive during ONU's sleep mode. Then, using Eq. (5) and (7), for a given set of values T_{min}, T_{max} and λ, the total average amount of delay experienced by the last frame that arrives during $E[d]$ is thus expressed as follows:

$$E[LF] = \frac{\sum_{j=1}^{\infty} e^{-\lambda S_{j-1}} \left\{1 - e^{-\lambda(T_j + L)}\right\} (T_j + L)}{2} + \frac{E[d]\lambda F_s}{D_r} + T_{p,m}, \tag{8}$$

where, $T_{p,\,m}$ is the propagation delay between the OLT and *ONU-m* ($m = \{1, 2, ..., N\}$).

Average frame delay results are presented in Fig. 6 based on Eq. (8) for different T_{min} and T_{max} values. Figure 6(a) depicts the influence of T_{min} on TDM-PON frame delay. It shows that at low arrival rate, T_{min} does not significantly affect traffic delay (this is also noticed in [7]). That is, regardless the T_{min} values, frame delay is similar in low arrival region (e.g. $\lambda < 0.05$ $frames/ms$). Nevertheless, as we can observe from this figure that with the increment of T_{min} value, frame delay increases in the high λ region. Note that when there is high frame arrival, the ONU has always frame to receive from the OLT after completing its *1st* sleep interval, which is equal to T_{min} (see Eq. (1)) [7]. Therefore, the larger the T_{min} value, the more frame delay. It is worth noticing from Fig. 6(a) that for a particular T_{min} value, traffic delay at low λ is very high and then, rapidly decreases as λ increases. However, as λ gradually increases, frame delay slowly grows up again. This phenomenon is noticeably observed for higher values of T_{min}. The reason to explain this is that for larger values of T_{min} (e.g. above 16 ms), at high λ, service time of the frames that arrive during an ONU's sleep interval period at the OLT is comparatively higher than that of the smaller values of T_{min}[2].

Here, we explain the influence of T_{max} on TDM-PON frame delay behavior. We can notice from Fig. 6(b) that the higher the T_{max} values, the more traffic delay over the low λ region (e.g. $\lambda < 0.05$ $frames/ms$). However, in high λ region, different T_{max} values provide almost the similar frame delay performance. It is also worth noticing from Fig. 6(b) that, regardless the T_{min} values, the delay performance results with the same T_{max} value are almost similar in low λ region.

3.2.2. *Traffic Delay and Sleep Interval Estimation for FL-SIMP*

Taking into account strict traffic delay requirement applications (e.g. teleprotection and synchrophasor applications in Smart Grid [27], and voice conversation (it has been defined by ITU-T that in a local exchange voice traffic delay should be less 5 ms [3])), we introduce FL-SIMP, in which an ONU uses T_{fix} when it moves into sleep mode. We consider that T_{fix} should be less than or equal to D_R. Additionally, to be on the safe side, we need to consider the required serving time for arrived frames during a sleep interval period while deciding on T_{fix}. Then, the total service time of the frames that arrive during a sleep interval ($T_{fix,\,i}$) of a particular *ONU-m* can be expressed as

$$Se_{L,\,FL} = \frac{T_{fix,\,i}\,\lambda F_s}{D_r}. \tag{9}$$

Then, to ensure that all the TDM-PON traffic of the *ONU-m* meet delay requirement when FL-SIMP is imposed, we come up with the following expression:

$$T_{fix,\,i} + Se_{L,\,FL} + T_{p,\,m} \leq D_R. \tag{10}$$

3.2.3. *Novel DRAPS Algorithm*

EUDDA seeks to develop a sleep management policy in which meeting traffic delay requirement is given more priority over energy saving. Even though energy saving is considered as the second goal of our

[2]Authors in [7] also explain the influence of T_{min} and T_{max} on frame delay performance. However, authors do not show the influence of frame service time on frame delay performance of an ONU.

proposal, DRAPS[3] algorithm puts effort to maximize energy saving in ONUs as much as possible. DRAPS applies FL-SIMP and EI-SIMP for a strict and relaxed delay requirement traffic, respectively.

In case of EI-SIMP, DRAPS finds a suitable value of T_{min} and T_{max} that can meet traffic delay requirement, while saving energy. Based on the previous discussion associated with influence of T_{min} and T_{max} on traffic delay, it can be concluded that DRAPS algorithm should find suitable T_{min} when traffic arrival rate is high. We have noticed earlier that over the low traffic arrival region T_{max} noticeably influences traffic delay performance (see Fig. 6(b)), whereas T_{min} negligibly affects (see Fig. 6(a)). Taking this into consideration, we conclude that our algorithm should play with both T_{min} and T_{max} in low traffic arrival situation. At this point it is important to mention that the algorithm proposed in [7] also finds suitable T_{min} and T_{max} for deciding ONUs' sleep interval lengths. However, unlike DRAPS algorithm, the algorithm in [7] finds either T_{min} or T_{max} (it does not deal with both of them at the same time). Similar to [7], we define parameters in Table 1 for DRAPS algorithm. Next, we present our DRAPS algorithm for EUDDA solution.

Table 1: Parameters used in DRAPS algorithm.

Notation	Description
$D_{R,\,DL}$	Delay requirement of downlink traffic defined by TDM-PON operator.
$D_{R,\,UL}$	Delay requirement of uplink traffic defined by TDM-PON operator.
λ_{DL}	Downlink traffic arrival rate.
λ_{UL}	Uplink traffic arrival rate.
$T_{\max\_Threshold}$	The maximum amount of time that can be assigned to an ONU to stay in *Sleep state* considering physical layer and TDM-PON protocol constrains.
$T_{\min\_Threshold}$	The minimum amount of time that can be assigned to an ONU to stay in *Sleep state* considering physical layer and TDM-PON protocol constrains.
S_{Th}	This represents the time difference between the last packet arrival time and the time when the OLT decides on putting the ONU into sleep mode.
$Q\{T_{\max 1}, T_{\max 2}, ..., T_{\max n}\}$	This set contains all the possible values of T_{max} ($T_{\max i} < T_{\max i+1}$ and $T_{\max n} \leq T_{\max\_Threshold}$).
$R\{T_{\min 1}, T_{\min 2}, ..., T_{\min n}\}$	This set contains all the possible values of T_{min} ($T_{\min i} < T_{\min i+1}$ and $T_{\min 1} \geq T_{\min\_Threshold}$).
$S\{T_{fix,\,1}, T_{fix,\,2}, ..., T_{fix,\,n}\}$	This set contains all the possible values of T_{fix} ($T_{fix,\,i} < T_{fix,\,i+1}$, $T_{fix,\,1} \geq T_{\min\_Threshold}$ and $T_{fix,\,n} \leq T_{\max\_Threshold}$).
λ_{Th}	Decision threshold related to traffic arrival rate based policies.
$T_{present}$	Current time.
$T_{Last\ arrival}$ (m)	The last packet arrival time for a particular *ONU-m*.
$D_{Req,\ th}$	Above which a delay requirement value is assumed to be relaxed (below or equal of this is considered as a strict delay requirement).

[3] DRAPS algorithm is a follow-up of our earlier work in [7].

DRAPS Algorithm: Finding T_{min} and T_{max} in EI-SIMP, and T_{fix} in FL-SIMP

```
Data: λ_Th, λ, D_R, T_max_Threshold, T_min_Threshold, Q, R, S;
/* λ = λ_DL if the OLT runs this algorithm. λ = λ_UL if an ONU runs this algorithm.                    */
/* D_R = D_R,DL if the OLT runs this algorithm. D_R = D_R,UL if an ONU runs this algorithm.            */
Result: T_min, T_max, T_fix;
begin
    while t_present - T_Last arrival (m) ≥ S_Th do
        if D_R ≤ D_Req, th then
            Choose sleep interval management policy as FL-SIMP;
            /* Calculate value of T_fix for FL-SIMP case;                                              */
            while T_Delay ≤ D_R do
                T_Delay = T_fix, i + (T_fix, i λF_s)/D_r + T_p, m;
                increment i by 1;
            T_fix = T_fix, i-1;
        if D_R > D_Req, th then
            Choose sleep interval management policy as EI-SIMP;
            /* Calcualte value of T_min and T_max for EI-SIMP case.                                     */
            if λ > λ_Th then
                T_max = T_max_Threshold;
                T_min = findT_min(D_R, λ, T_max, T_min_Threshold, T_p, m);
            if λ ≤ λ_Th then
                T_max = findT_max(D_R, λ, T_min_Threshold, T_p, m);
                T_min = findT_min(D_R, λ, T_max, T_min_Threshold, T_p, m);
            /* value of T_min and T_max are the final outputs in EI-SIMP case, whereas value of T_fix is
               the final output in FL-SIMP case.                                                        */
```

$findT_{min}$: Sub-algorithm to find T_{min}

```
Data: R, T_max_Threshold;
Result: T_min;
begin
    while D_measured ≠ D_R do
        D_measured = Algorithm 1(T_max_Threshold, T_min i, λ, T_p, m);
        increment i by 1;
    return T_min i/2; /* return this value to the DRAPS algorithm.                                      */
```

$findT_{max}$: Sub-algorithm to find T_{max}

```
Data: Q, T_max_Threshold;
Result: T_max;
begin
    while D_measured ≠ D_R do
        D_measured = Algorithm 1(T_max i, T_min_Threshold, λ, T_p, m);
        increment i by 1;
    return T_max i; /* return this value to the DRAPS algorithm.                                        */
```

Algorithm 1: Sub-algorithm to measure delay

/* The calling function passes four parameters. This function uses first two parameters to assign values of T_{min}^{temp} and T_{max}^{temp}, while other two parameters are used for $E[LF]$ calculation using Eq. (8). */

Data: $L, \lambda, D_T, T_{min\,threshold}, T_{max\,threshold}$;

Result: $E[LF]$; /* $E[LF]$: total delay experienced by the last frame. */

begin

/* Calculate $E[LF]$ using Eq. (8) */

$$E[LF] = \frac{\sum_{j=1}^{\infty} e^{-\lambda S_{j-1}} \left\{1 - e^{-\lambda(T_j + L)}\right\}(T_j + L)}{2} + \frac{E[d]\lambda F_s}{D_r} + T_{p,m}$$

/* in Eq. (8), T_j is measured using following expression (i.e. Eq. (1)) */

/* $T_j = min(2^{j-1} T_{min}^{temp}, T_{max}^{temp})$; */

return $E[LF]$; /* return this value to the calling function. */

In EUDDA, DRAPS algorithm is used by both the OLT and ONU. One of the important roles of DRAPS algorithm is to decide on whether a particular *ONU-m* can be put into sleep mode or not based on a threshold value (S_{Th}). If no downlink or uplink frames arrive during S_{Th}, the DRAPS algorithm of the OLT and *ONU-m* follows the following procedures.

If no frames arrive during S_{Th}, DRAPS algorithm checks whether the assigned delay requirement for the traffic of the *ONU-m* is strict or relaxed. If it finds that operator's assigned delay requirement is a strict delay requirement (i.e. $D_R \leq D_{Req,\,th}$), it chooses FL-SIMP and finds a suitable fixed sleep interval length (T_{fix}), that *ONU-m* uses when it moves into sleep mode.

Conversely, in a relaxed delay requirement case (i.e. $D_R > D_{Req,\,th}$), DRAPS algorithm selects EI-SIMP and follows a separate procedures to find a value of T_{min} and T_{max}. At this point, it is worth to mention that T_{min} can significantly influence traffic delay performance in high traffic arrival region, whereas T_{max} affects traffic delay behavior negligibly in that region, as we have noticed from Fig. 6(a). Therefore, if traffic arrival rate is above a threshold (λ_{Th}), DRAPS algorithm seeks for a suitable T_{min} from the set R while T_{max} is set to $T_{\max\_Threshold}$ similar to the algorithm in [7]. The DRAPS algorithm invokes $findT_{min}$ function to find a suitable T_{min}.

If the operator assigned delay requirement is relaxed and $\lambda < \lambda_{Th}$ (traffic arrival rate is considerably low), proposed DRAPS algorithm looks for obtaining suitable value of both T_{min} and T_{max}. Note that in this particular case, the algorithm in [7] only seeks to find a suitable value of T_{max} and sets $T_{min} = T_{\min\_Threshold}$ ($T_{\min\_Threshold}$ is actually the smallest possible value of T_{min}). However, in this work, first, to find a suitable T_{max} that can maximize energy saving and meet operator assigned delay requirement, the DRAPS algorithm invokes $findT_{max}$ function. Then, the $findT_{max}$ function returns with a suitable T_{max} value. Note that to find a suitable value of T_{max}, the $findT_{max}$ function uses $T_{\min\_Threshold}$. However, we have noticed from Fig. 6(b) that different T_{min} and T_{max} pairs with the same T_{max} value (e.g. ($T_{min} = 1\ ms$, $T_{max} = 8\ ms$) and ($T_{min} = 4\ ms$, $T_{max} = 8\ ms$)) yield practically the same frame delay results in a low traffic arrival region. Therefore, in our solution, DRAPS algorithm uses the T_{max} value that $findT_{max}$ function returns to find the maximum possible T_{min} value that can still satisfy delay requirement (D_R) using the $findT_{min}$ function. Note that using a larger value T_{min} than $T_{\min\_Threshold}$ will lead to make our solution more energy-efficient in a low traffic arrival region compared to the solution in [7], in which in a low traffic arrival region, only a suitable T_{max} is found while setting $T_{min} = T_{\min\_Threshold}$. The motivation behind using larger values T_{min} in this particular scenario in our solution is that, the longer the value of T_{min}, the less number of transitions from *Sleep state* and idle listening periods an ONU can have, and thus the higher the energy saving in ONUs.

3.3. Proposed Operational Procedures of EUDDA

In this subsection, putting all our different contributions together, we present our novel operational procedure. Figure 7 presents overall procedures of EUDDA. Similar to [7, 13, 14], here, we assume that the

OLT in a TDM-PON remains always active. In EUDDA, an ONU moves into sleep mode only when it does not have any traffic to transmit and receive. However, if an ONU does not have any uplink traffic but the OLT has downlink traffic to transmit (or the other way around), an ONU should not move into sleep mode.

In our solution, both the OLT and an ONU use DRAPS algorithm to measure an ONU's sleep mode associated parameters (i.e. T_{min} and T_{max} in EI-SIMP or T_{fix} in FL-SIMP). Next, these measured sleep mode associated parameters are passed to Parameters Sharing and Selection (PSS) algorithm, which runs also on both the ONU and OLT (see Fig. 8). Eventually, the PSS algorithm is in charge of selecting final sleep mode associated parameters which the ONU uses when it enters into sleep mode (we explain these procedures in details in the subsequent part of this subsection). For a better explanation of EUDDA's working procedures, we assume that at a given point of time the OLT and a particular ONU (*ONU-m*) do not have any traffic to exchange.

(A) If the SIM unit of the OLT finds that there is no downlink and uplink traffic arrival for the *ONU-m*, it uses DRAPS algorithm to select appropriate SIMP. If FL-SIMP is selected (strict delay requirement case), DRAPS algorithm calculates a suitable value of T_{fix}, which is notified to the *ONU-m*. Similarly, in absence of downlink and uplink traffic arrival, the *ONU-m*'s SIM unit uses DRAPS algorithm to measure a suitable value of T_{fix}, which in turn is forwarded to the OLT. At this point, both the OLT and *ONU-m* have two suitable values of T_{fix}. Then, the PSS algorithm of both the OLT and *ONU-m* uses the following expression to derive final value of T_{fix}, which the *ONU-m* uses as a sleep interval length when it moves into sleep mode.

$$T'_{fix} = \begin{cases} T_i, & \text{if } T_i \leq T_i^r \\ T_i^r, & otherwise \end{cases}, \tag{11}$$

where, T_i is the value of T_{fix} calculated by DRAPS algorithm in a local node, whereas T_i^r is T_{fix} value measured in a remote node. For example, if the PSS algorithm of the *ONU-m* uses Eq. (11) to obtain final T_{fix} value, T_i and T_i^r should be interpreted by the PSS algorithm as calculated T_{fix} value measured by the DRAPS algorithm of the *ONU-m* (local node) and OLT (remote node), respectively. The same interpretation is valid when the PSS algorithm chooses final value of T_{min} and T_{max} in EI-SIMP case.

If EI-SIMP is selected (relaxed delay requirement case), the DRAPS algorithm in the OLT calculates a suitable value of T_{min} and T_{max} based on different important parameters (i.e. $D_{R,DL}$ and λ_{DL} of the *ONU-m*). After deciding on the value of T_{min} and T_{max}, the OLT notifies the *ONU-m* and waits for the reply from the ONU. In reply, the *ONU-m* notifies the value of T_{min} and T_{max}, which are calculated by its DRAPS algorithm after taking into consideration $D_{R,UL}$ and λ_{UL}. Once the value of T_{min} and T_{max} are exchanged between the OLT and *ONU-m*, the PSS algorithm of the OLT and *ONU-m* uses Eq. (12) and Eq. (13) to obtain the final length of T_{min} and T_{max}, which will be used by the *ONU-m* to manage sleep mode as long as it does not have any uplink and/or downlink traffic arrival.

$$T'_{min} = \begin{cases} T_{\min\,i}, & \text{if } T_{\min\,i} \leq T^r_{\min\,i} \\ T^r_{\min}, & otherwise \end{cases}, \tag{12}$$

where, $T_{\min\,i}$ and $T^r_{\min\,i}$ are interpreted by the PSS algorithm as the calculated values of T_{min} by a local node and remote node, respectively.

$$T'_{max} = \begin{cases} T_{\max\,i}, & \text{if } T_{\max\,i} \leq T^r_{\max\,i} \\ T^r_{\max}, & otherwise \end{cases}, \tag{13}$$

where, $T_{\max\,i}$ and $T^r_{\max\,i}$ are interpreted by the PSS algorithm as the calculated values of T_{max} by a local node and remote node, respectively.

(B) At this point, both the *ONU-m* and OLT know sleep mode associated parameters (i.e. T_{min} and T_{max} in EI-SIMP, and T_{fix} in FL-SIMP). Next, the *ONU-m* manages sleep mode using the parameters.

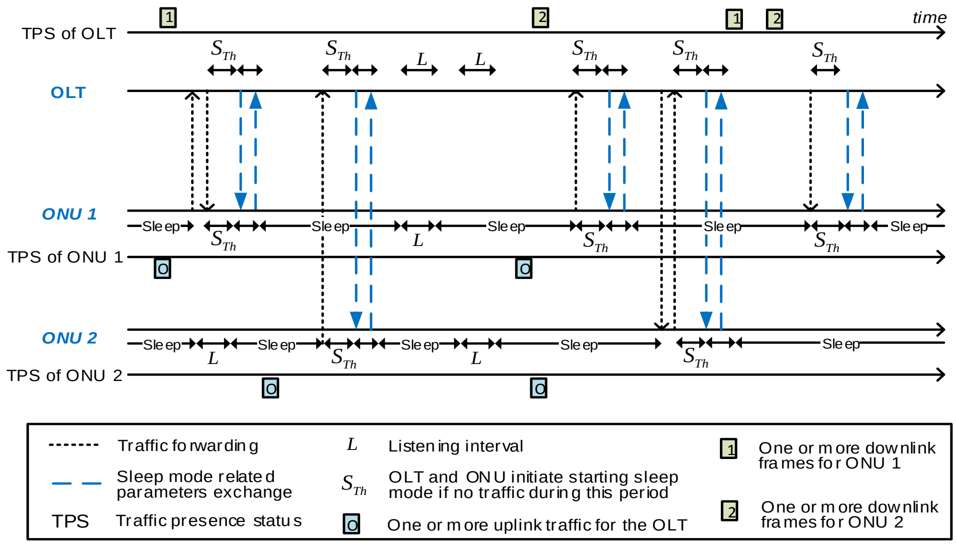
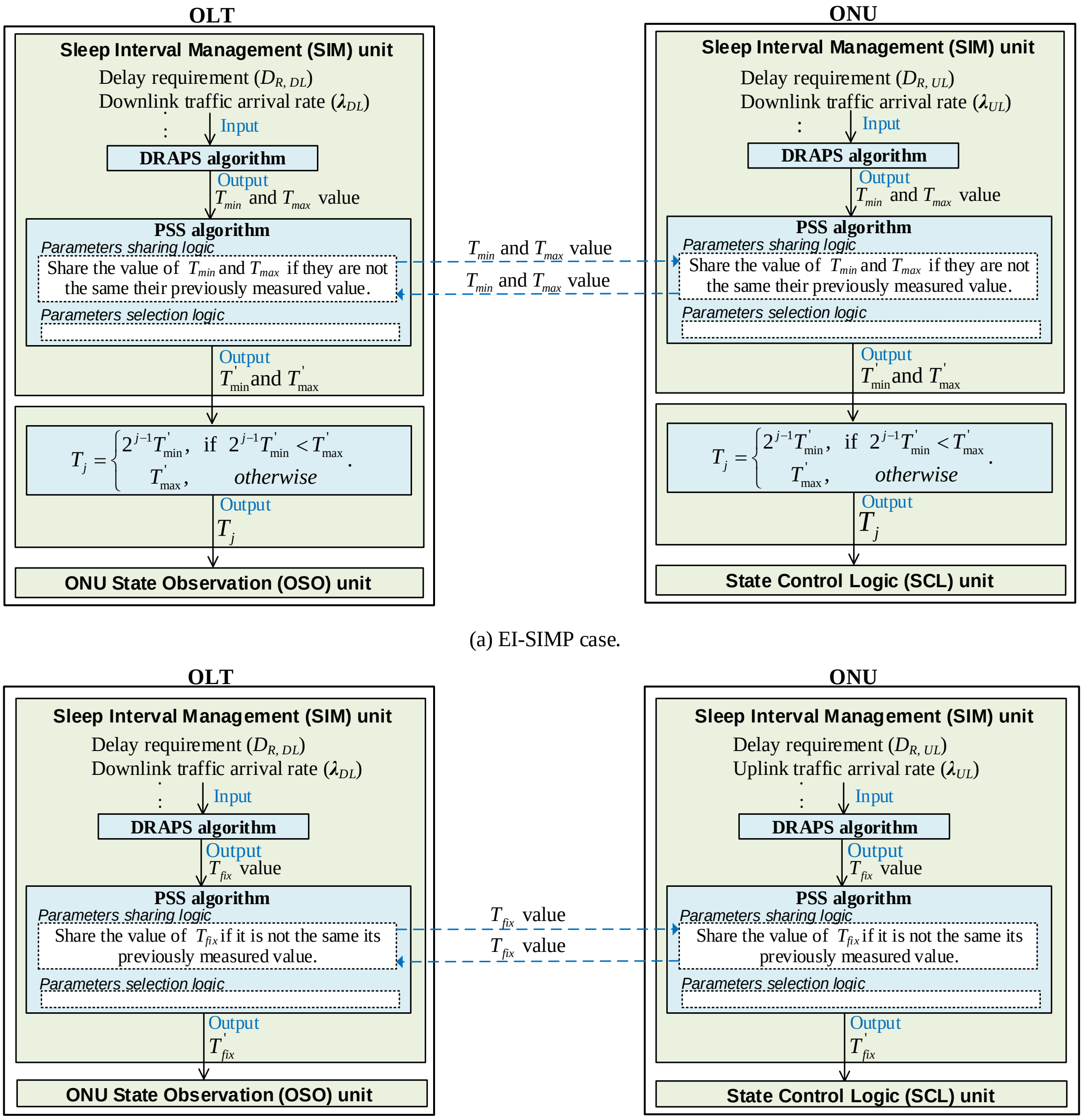


Fig. 7: Overall working procedures in EUDDA.

In case of FL-SIMP, the *ONU-m* moves from *Sleep state* to *Active* or *Rx state* after T'_{fix} to listen OLT's instruction. In our solution, after an ONU moves from *Sleep state* to listen OLT's instruction (after *Sleep state* an ONU waits for L amount of time for the OLT's instruction), the OLT directly sends traffic to the ONU if it has something to forward. Note that it is likely that the OLT might need to forward downlink traffic to one or more ONUs which are in *Rx state* or *Active state* waiting for OLT's instruction after completion of their *Sleep state*. In that case, the OLT needs to share downlink bandwidth among those ONUs to forward downlink traffic. On the other hand, if the ONU has uplink traffic to send, it sends a bandwidth request message mentioning its required amount of bandwidth after leaving *Sleep state*. The OLT collects the bandwidth request and assigns a slot to the *ONU-m* after taking into consideration all the bandwidth request of ONUs having uplink traffic to send. If the ONU has neither downlink traffic to receive nor any uplink traffic to transmit, the ONU moves into *Sleep state* again for T'_{fix} amount of time.

In case of EI-SIMP, other than the sleep interval length calculation procedures, all other procedures the OLT and *ONU-m* follow are exactly the same as explained above. In this particular case, the *ONU-m* uses sleep mode associated parameters (T_{min} and T_{max}) to calculate a sleep interval length based on the Eq. (1). As the OLT also knows those sleep mode associated parameters, it uses Eq. (1) to know when the *ONU-m* is available for future communication.

(C) As long as there is no traffic arrival from both directions, the sleep mode associated parameters are not changed in our solution. However, after new uplink or/and downlink traffic arrival (completion of ONU's sleep mode), the sleep mode associated parameters used for previous sleep mode are considered invalid for the next sleep mode. Hence, the *ONU-m* needs to take part in deciding the next sleep mode associated parameters again when there is no traffic arrival during S_{Th}, following the procedures explained in (A). However, in our solution, if the OLT and ONU find that the newly calculated parameters are the same as the previous ones, then they abstain from exchanging sleep mode associated parameters, considering previous sleep mode associated parameters as the current one. Note that whenever an ONU needs to transmit something it needs to switch on its Tx unit, which consumes 2.99 *W* [12, 25]. Therefore, minimizing control message exchange associated with sleep mode management could lead to save significant amount of energy in ONUs in our solution. Therefore, considering this fact, the PSS algorithm, which is presented at the end of this section, decides whether newly calculated sleep mode related parameters by DRAPS algorithm should be shared or not (see Fig. 8



(a) EI-SIMP case.

(b) FL-SIMP case.

Fig. 8: Sleep mode associated parameters sharing and selection procedures of EUDDA under EI-SIMP and FL-SIMP.

for better visualization of the aforementioned procedures). Table 2 defines some of the important parameters used in PSS algorithm.

(D) The SCL unit of *ONU-m* is in charge of selecting appropriate state at a given time. It selectively turns on or off the Tx and Rx unit based on uplink and downlink communication requirement, as explained in Fig. 4.

Table 2: Parameters used in PSS algorithm.

Notation	Description
$T_{min,\,i}$	Currently calculated T_{min} value.
$T_{min,\,i-1}$	Previously calculated T_{min} value.
$T_{max,\,i}$	Currently calculated T_{max} value.
$T_{max,\,i-1}$	Previously calculated T_{max} value.
T_i	Currently calculated T_{fix} value.
T_{i-1}	Previously calculated T_{fix} value.

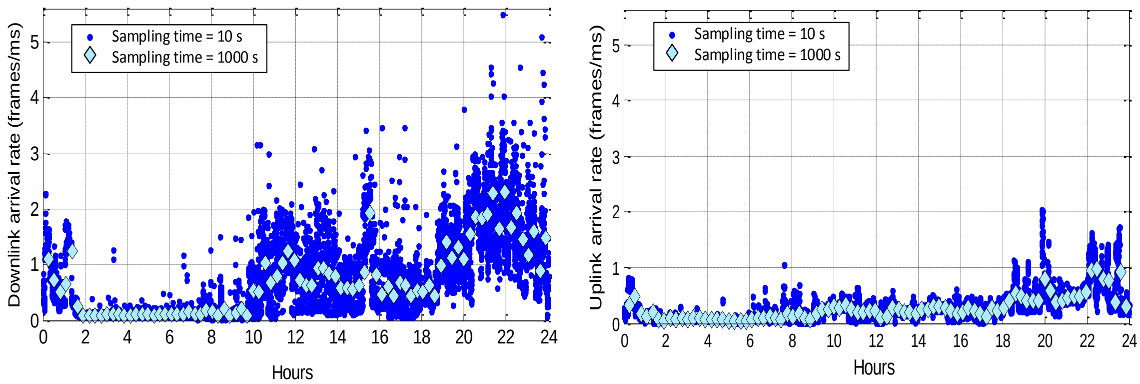
PSS Algorithm: Parameter selection and sharing for EI-SIMP and FL-SIMP

```
Data: T_{min,i}, T_{min,i-1}, T_{max,i}, T_{max,i-1}, T_i, T_{i-1}, T^r_{min,i}, T^r_{min,i-1}, T^r_{max,i}, T^r_{max,i-1}, T^r_i, T^r_{i-1};
/* a superscript "r" indicates that a particular value is measured in a remote node. For example, if this
   PSS algorithm runs in a particular ONU-m, then received values from the OLT at ONU-m should
   be identified with the superscript "r" (e.g. T^r_{min,i}) (here OLT is interpreted as a remote node).   */
Result: T'_{min}, T'_{max}, T'_{fix};
begin
    if D_R ≤ D_{Req,th} then
        /* Relaxed delay requirement case (EI-SIMP)                                                          */
        if T_{min,i} = T_{min,i-1} and T_{max,i} = T_{max,i-1} then
            T_{min,i} ← T_{min,i-1};
            T_{max,i} ← T_{max,i-1};
        else
            Notify T_{min,i} and T_{max,i} to remote node;   /* if this algorithm runs at an ONU, the
                                                  'remote node' refers an OLT, and vice versa.       */
        Wait for receiving newly calculated T_min and T_max values during L from remote node;
        if receiving T_min and T_max value within L is true then
            T^r_{min,i} ← Received T_min value;
            T^r_{max,i} ← Received T_max value;
        else
            T^r_{min,i} ← T^r_{min,i-1};           /* update with previously received T_min value.   */
            T^r_{max,i} ← T^r_{max,i-1};           /* update with previously received T_max value.   */
            Use Eq. (12) and Eq. (13) to quantify final value of T_min and T_max with parameters T_{min,i},
            T^r_{min,i}, T_{max,i} and T^r_{max,i};
    else
        /* Strict delay requirement case (FL-SIMP)                                                           */
        if T_i = T_{i-1} then
            T_i ← T_{i-1};
        else
            Notify T_i to remote node;
        Wait for receiving newly calculated T_fix value during L from remote node;
        if receiving T_fix value within L is true then
            T^r_i ← Received T_fix value;           /* update with recently received T_fix value.     */
        else
            T^r_i ← T^r_{i-1};                      /* update with previously received T_fix value.   */
            Use Eq. (11) to quantify final value of T_fix with parameters T_i and T^r_i;
```

4. Performance Evaluation

In this section, we compare our proposed EUDDA in front of two other solutions, which are proposed in [7] and [11], based on real network traffic traces as depicted in Fig. 9. At this point it is worth noting that an ONU's sleep interval length is measured based on sleep mode associated parameters under EUDDA' EI-SIMP and the solutions presented in [7, 11] rely on Eq. (1). This has motivated us to compare the performance of EUDDA with the solutions introduced in [7, 11]. However, in contrast to EUDDA, the solution in [11] uses fixed length of T_{min} and T_{max} always. Hence, in this section, we refer to this solution as Fixed T_{min} and T_{max} Selection (FTS) solution. Unlike our solution, proposed solution in [7] finds either a suitable value of T_{min} or T_{max} always for a given amount of delay requirement (Section 2 briefly explains this work). We refer to this solution as Adaptively T_{min} or T_{max} Selection (ATS) solution. We compare the performances of proposed solution with ATS and FTS by means of our C++ discrete event simulator[4]. To understand performances of these solutions under real network traffic traces, traffic of eight home and eight office users have been captured, assuming that there are 16 users served by an ONU, similar to [28].

In this section, based on real network traces, performance of EUDDA, FTS and ATS is measured in terms of frame delay, energy consumption and jitter performance. Table 3 represents parameters used in simulation. For convenience of result interpretation, we assume that at a given time a TDM-PON operator assigns the same delay requirement (D_R) for the traffic of both uplink and downlink direction of an ONU. To make a fair comparison of energy consumption performances of these three solutions, we assume that all these solutions use the same type of ONU architecture in which both the Tx and Rx can be truned on/off selectively and there exists four states (*Sleep state, Active state, Rx state, Tx state*).



(a) Downlink traffic arrival rate.

(b) Uplink traffic arrival rate.

Fig. 9: Downlink and uplink frame arrival rate of 16 users under an ONU during 24 h [7].

4.1. Delay Performance

Downlink traffic delay performance results are presented in Fig. 10 for those three different solutions (EUDDA, FTS, and ATS). To asses frame delay results, we present frame delay Cumulative Distributed Function (CDF) for each of the solutions. We evaluate their performance under two strict and one very relaxed delay requirement scenarios.

4.1.1. Delay Performance in Strict Delay Requirement Scenarios

Note that according to the ITU-T, in the local exchange, voice traffic delay should be less than 5 *ms* [3]. To asses how our proposed solution performs under such strict delay requirement scenario, we have conducted simulation considering $D_R = 5\,ms$ and plotted the results in Fig. 10(a). Results show that EUDDA

[4]The C++ discrete event simulator is developed in our work at KAIST for TDM-PONs' performance evaluation.

Table 3: Simulation parameters.

Description	Values
Power consumption in *Active state*	4.69 *W* [10, 29]
Power consumption in *Sleep state*	0.7 *W* [10, 25]
Power consumption in *Tx state*	2.99 *W* [25]
Power consumption in *Rx state*	1.7 *W* [25, 29]
Sleep state to *Active state* transition period	2 *ms* [13, 17]
Downlink and uplink data rate	1 *Gbps*
$T_{\min\_Threshold}$	3 *ms* [7]
$T_{\max\_Threshold}$	50 *ms* [7]
λ_{Th}	0.05 *frames/ms* [7]
$T_{p,\,m}$	0.2 *ms* [7, 13]
Traffic sampling time window	10 *s*
Number of ONUs in a TDM-PON	16
Uplink DBA grant cycle length (T_{cycle})	3 *ms* [25]
$D_{Req,\,th}$	10 *ms* [7]

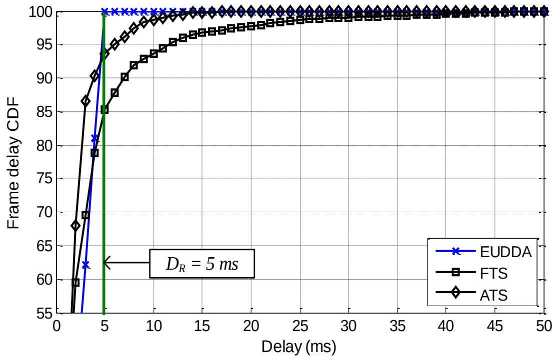
outperforms the other two solutions. In particular, it meets 100% of downlink traffic delay requirement, whereas, ATS and FTS satisfy 94% and 85% delay requirement, respectively. This happens because the proposed EUDDA uses the novel DRAPS algorithm which treats a strict and relaxed delay requirement traffic differently. As this is a strict delay requirement scenario, DRAPS uses FL-SIMP (Fixed length sleep interval management policy), the prime objective of which is to meet delay requirement of all incoming and outgoing traffic of an ONU in a TDM-PON. On the other hand, downlink frame delay CDF performance in ATS is superior than that of the performance result of FTS, as we can notice from Fig. 10(a). Note that both ATS and FTS use Eq. (1), in which lower bound (T_{min}) and upper bound (T_{max}) are given as inputs, to decide a sleep interval length. In both of these solutions, an ONU's sleep interval length increases exponentially (see Eq. (1)). ATS solution limits the size of T_{max} in strict delay requirement scenarios, so that traffic delay requirement is not violated. However, FTS does not have control over the length of T_{max}, resulting in unrestrained sleep interval increment. Consequently, FTS fails to maintain downlink traffic delay requirement significantly in a strict delay requirement scenario, as depicted in Fig. 10(a).

Figure 10(b) presents performance evaluation results of those three solutions for another strict delay requirement scenario. In this case, we set $D_R = 10\,ms$. This is actually a slightly relaxed delay requirement case than the previous one (i.e. $D_R = 5\,ms$). However, it is still considered as a strict delay requirement scenario. Therefore, EUDDA's DRAPS algorithm applies FL-SIMP, so that 100% of traffic can meet delay requirement. As can be observed from this figure, EUDDA is the one meeting 100% delay requirement of downlink traffic. Whereas, 96% and 93% of downlink traffic satisfy delay requirement in ATS and FTS, respectively.

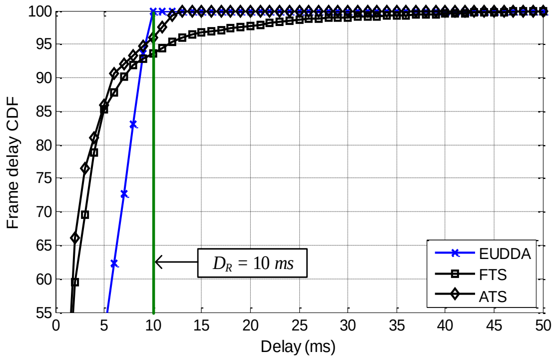
Uplink traffic delay results for EUDDA, ATS and FTS are presented in Fig. 11. We can observe EUDDA's performance for $D_R = 5\,ms$ and $D_R = 10\,ms$ in Fig. 11(a) and (b), respectively. Similar to the downlink performance results, EUDDA meets 100% of uplink traffic delay requirement, as can be noticed from these figures. This happens because EUDDA's DRAPS algorithm applies FL-SIMP to meet delay requirement of all uplink traffic.

As it is shown in Fig. 11(a) that in ATS and FTS all uplink frames do not meet delay requirement ($D_R = 5\,ms$). However, in case of $D_R = 10\,ms$ (Fig. 11(b)), both of these solutions satisfy delay requirement of 100% of uplink traffic. Note that both FTS and ATS apply early wake-up mechanism (an ONU leaves *Sleep state* instantly on uplink traffic arrival) for the uplink traffic of ONUs. Even an ONU in those solutions uses early wake-up, we can observe noticeable amount of delay experienced by uplink traffic from Fig. 11(a) and (b). The possible reasons behind are explained in the following paragraph.

According to early wake-up mechanism (defined in [18, 19]), when a sleeping ONU is interrupted due to uplink traffic arrival, it should leave *Sleep state* and complete synchronization and OLT's clock recovery,

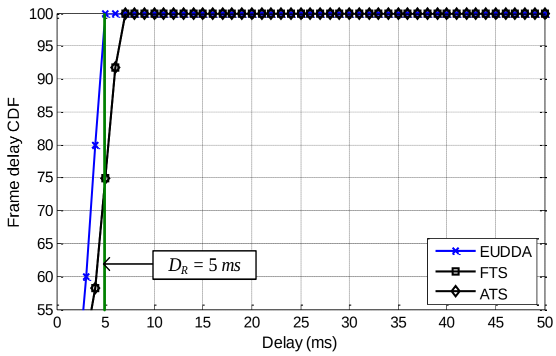


(a) D_R = 5 *ms.*

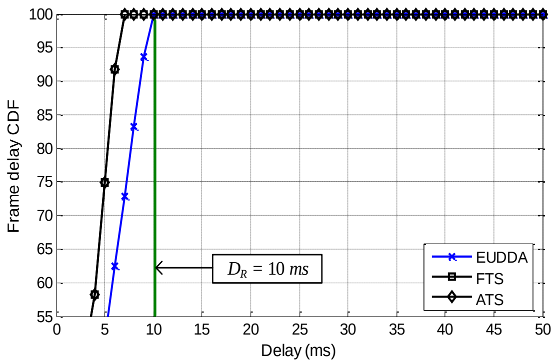


(b) D_R = 10 *ms.*

Fig. 10: Downlink frame delay performance comparison under strict delay requirement scenarios.



(a) D_R = 5 *ms.*

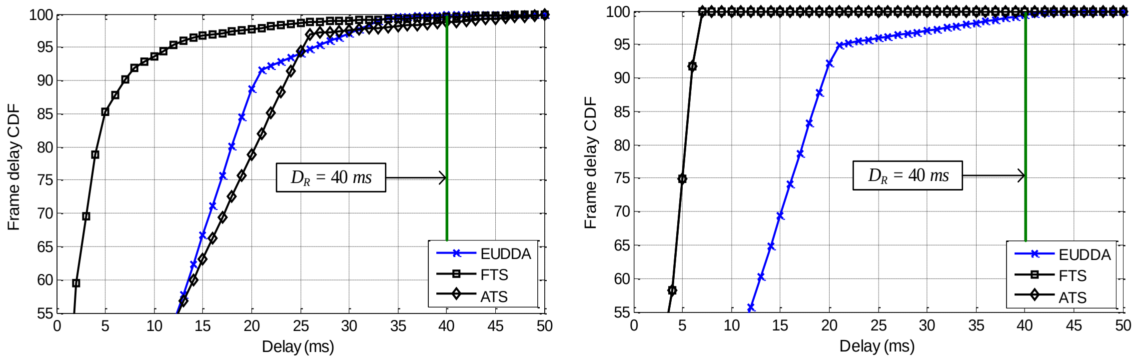


(b) D_R = 10 *ms.*

Fig. 11: Uplink frame delay performance comparison under strict delay requirement scenarios.

and then make bandwidth request to the OLT using a dedicated slot [13, 18, 22]. Upon receiving ONU's bandwidth request, the OLT assigns an uplink transmission slot within the next DBA grant cycles. It is worth to note that, after making bandwidth request to the OLT, an ONU might need to wait at most two DBA grant cycles for receiving its uplink transmission slot related information from the OLT [13]. Therefore, this is one of the important factors that could affect uplink traffic delay performance significantly in those solutions where early wake-up mechanism is employed (e.g. FTS [11] and ATS [7]). Another important reason that could contribute in increasing uplink traffic delay of a particulate ONU is the amount of uplink bandwidth demand of other ONUs in the same TDM-PON system. If at a given time the uplink traffic arrival of ONUs are high, an ONU might end up receiving uplink transmission bandwidth less than its requested amount. In fact, this phenomenon could lead to increase uplink frame delay and frame drop due to ONU's buffer overflow. Therefore, even an ONU in FTS and ATS leaves *Sleep state* instantly on uplink traffic arrival, the traffic could experience delay due to these reasons. Since FTS and ATS use the same principle (i.e. early wake-up mechanism) for uplink traffic, their delay performance are almost identical as depicted in Fig. 11.

FTS and ATS can provide satisfactory uplink traffic delay performance; however, they cannot meet 100% of downlink traffic delay requirement in a strict delay requirement scenario. Therefore, this draws us to the conclusion that FTS and ATS cannot be dispensed in a TDM-PON in which an operator seeks to meet 100% of traffic delay requirement in a strict delay requirement scenario and manage sleep mode in ONUs in order to maximize energy-efficiency of the TDM-PON system. At this point, it is worth to note



(a) Downlink traffic delay performance. (c) Uplink traffic delay performance.

Fig. 12: Downlink and uplink frame delay performance comparison under a relaxed delay requirement scenario.

that proposed EUDDA not only meets delay requirement of 100% of downlink and uplink traffic of an ONU but also shows noticeably energy saving performance improvement compared to two other solutions (we present an ONU's energy consumption results when EUDDA is in place in subsection 4.3).

4.1.2. Delay Performance in a Relaxed Delay Requirement Scenario

Next we present performance results of EUDDA, ATS and FTS under a very relaxed delay requirement scenario (i.e. $D_R = 40\,ms$). Figure 12(a) and (b) illustrate downlink and uplink traffic delay performance for those three solutions, respectively.

As this is a relaxed delay requirement scenario, EUDDA's DRAPS algorithm imposes EI-SIMP (exponentially incrementing sleep interval management policy). Similar to ATS and FTS, our proposed EUDDA uses Eq. (1) to calculate ONUs' sleep interval lengths in EI-SIMP. However, unlike FTS (uses fixed length of T_{min} and T_{max}) and ATS (finds either a suitable value of T_{min} or T_{max}), EUDDA seeks to find a suitable value T_{min} and T_{max} at the same time. The prime objective in EUDDA under a relaxed delay requirement scenario (e.g. $D_R = 40\,ms$) is to reduce an ONU's energy consumption as much as possible and meet traffic delay requirement. Results presented in Fig.12(a) and (b) show that all these solutions meet both downlink and uplink traffic delay requirement successfully. However, we can observe from these figures that EUDDA is showing the worst delay performance (but none of the traffic violates delay requirement boundary in EUDDA) compared to FTS and ATS. This is so because EUDDA deliberately uses larger values T_{min} and T_{max}, so that an ONU's energy saving performance can be maximized remarkably (see energy saving performance of EUDDA under a relaxed delay requirement scenario in Fig. 14).

4.2. Jitter Performance

Jitter performance of these three solutions are quantified based on the following equation [30, 31].

$$j = \sqrt{\frac{1}{N_F}\sum_{k=1}^{N_F}(x_k - \bar{x})^2}, \tag{14}$$

where, N_F is the total number of frames in a particular direction (e.g. downlink), x_k is the delay experienced by *k-th* frame and $\bar{x}$ is the mean delay of N_F frames.

Jitter or delay variance can have significant influence on voice and video quality. It is desired to have smaller jitter value while delivering high-quality voice transmission [3]. Table 4 compares uplink and downlink traffic jitter performance of EUDDA in front of two other solutions under a strict ($D_R = 10\,ms$) and relaxed delay requirement ($D_R = 40\,ms$) scenario. Note that ATS [7] is also a delay requirement aware solution. Therefore, here, we also present ATS's jitter performance for both strict and relaxed delay requirement scenarios. Unlike ATS and EUDDA, meeting traffic delay requirement is not a concern in FTS. Therefore, in this case, a single result is reported for each communication direction in Table 4.

We can notice from Table 4 that jitter performance of EUDDA for both downlink and uplink traffic is noticeably better than that of other two solutions in $D_R = 10\,ms$ scenario. This good performance is reached since EUDDA applies FL-SIMP in a strict delay requirement scenario. In FL-SIMP, after entering into sleep mode, an ONU uses a fixed length of sleep interval, which is decided by the OLT and ONU after taking into consideration several important parameters including operator's imposed traffic delay requirement (D_R) for the ONU. Consequently, EUDDA not only can improve jitter performance compared to other solutions but also meet traffic delay requirement (see Fig. 10 and Fig. 11) in each strict delay requirement scenario.

Table 4 shows that jitter performance in ATS for the strict delay requirement case is worse than EUDDA. An explanation of this is as follows. ATS always uses Eq. (1) to calculate sleep interval length after deciding on value of T_{min} and T_{max}. According to this solution, sleep interval length increases exponentially as long as there is no traffic for a particular ONU. During low traffic arrival period, sleep interval length could reach up to its maximum value (T_{max}). Therefore, during that period, if ATS fails to choose an appropriate T_{max} value, long sleep interval lengths of an ONU could result in violating assigned traffic delay requirement. This also results in worsening jitter performance compared to EUDDA in which 100% of traffic satisfy delay requirement (see Fig. 10(a) and (b)). Similarly, FTS, which uses fixed value of T_{min} and T_{max} without being aware of delay requirement of traffic, shows the worst performance in terms of traffic delay and jitter among all three solutions when $D_R = 10\,ms$ (strict delay requirement scenario) as we can observe from Fig. 10(b) and Table 4, respectively.

We can also notice uplink traffic jitter performance from Table 4. Similar to the downlink traffic jitter results, EUDDA is the one showing the least jitter value among uplink traffic jitter performances of all three solutions when $D_R = 10\,ms$. Note that both ATS and FTS experience higher uplink delay than EUDDA in a strict delay requirement scenario (see Fig. 11(a) and (b)). This results in increasing jitter value in ATS and FTS compared to the uplink jitter result of EUDDA when $D_R = 10\,ms$.

When delay requirement is relaxed ($D_R = 40\,ms$), FTS is showing the least downlink traffic jitter value among all three solutions, as can be noticed from Table 4. Note EUDDA applies EI-SIMP in a relaxed delay requirement scenario. In this case, both EUDDA and ATS increase sleep interval length deliberately in order to maximize energy saving while meeting traffic delay requirement. Consequently, this results in increasing delay of downlink traffic in ATS and EUDDA compared to FTS (see Fig. 12(a)), thereby providing poorer jitter performance compared to FTS.

The same table presents uplink traffic jitter performance when $D_R = 40\,ms$. As mentioned earlier, EUDDA deliberately chooses a larger value of T_{min} and T_{max} in order to maximize energy saving in a relaxed delay requirement scenario. This results in worsening uplink frame delay performance of EUDDA compared to FTS and ATS (but, none of the traffic violates delay requirement boundary in EUDDA (see Fig. 12(b))). And the same cause is responsible for increasing jitter value of uplink traffic in EUDDA when $D_R = 40\,ms$ compared to ATS and FTS, in which an ONU uses early wake-up mechanism.

Table 4: Jitter performance comparison.

Solution	**FTS**	**EUDDA**		**ATS**	
		$D_R = 10\,ms$	$D_R = 40\,ms$	$D_R = 10\,ms$	$D_R = 40\,ms$
Downlink jitter	5.280 ms	2.133 ms	8.956 ms	2.644 ms	7.097 ms
Uplink jitter	2.188 ms	2.147 ms	9.282 ms	2.164 ms	2.175 ms

Note that even ATS and EUDDA show inferior jitter performance in a relaxed delay requirement scenario (i.e. $D_R = 40\,ms$) compared to FTS, their jitter performance is still acceptable for most of the traffic with relaxed delay requirement (e.g. VBR (Variable Bit Rate) traffic, which is regarded as a relaxed delay requirement traffic [7], has jitter performance requirement ranging from 30 ms to 50 ms [32]).

4.3. ONU Energy Consumption Performance

In this subsection, we compare an ONU's energy consumption performance in EUDDA, ATS and FTS. We quantify energy consumption performance as the portion that EUDDA, FTS and ATS expend compared to the solution in which an ONU is always active (e.g. IEEE 802.3ah [8]). We refer to the solution as Always Active (AA) solution. Figure 13 and Fig. 14 represent ONU's energy consumption performance for a strict and relaxed delay requirement scenario, respectively.

4.3.1. Energy Consumption Performance in Strict Delay Requirement Scenarios

Figure 13(a) depicts energy consumption performance of those three different solutions when $D_R = 5\,ms$. Note that this is a strict delay requirement scenario. Therefore, EUDDA's DRAPS algorithm selects FL-SIMP, so that all frames can meet delay requirement (see Fig. 10 and 11). Energy consumption results presented in Fig. 13(a) (for every 2 hours) show that in most cases EUDDA outperforms the other solutions. Global energy consumption results (24 hours' energy consumption) relative to AA solution for this strict delay requirement case are: 36.4% for FTS, 42% for ATS and 33.6% for EUDDA. These results impart that EUDDA is the most energy-efficient solution among all three solutions even in a strict delay requirement scenario.

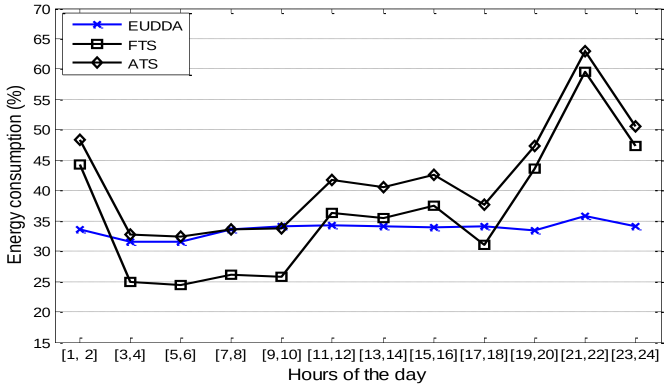
Figure 13(b) presents energy consumption performance evaluation results for those solutions for another strict delay requirement scenario ($D_R = 10\,ms$). Although this is a strict delay requirement scenario, it is still bit relaxed delay requirement case than the previous one. Therefore, EUDDA has bigger opportunity to save energy in this case. Similar to the previous delay requirement case, EUDDA is consuming the least amount of energy compared to two other solutions as can be observed from Fig. 13(b). Therefore, in both of these strict delay requirement scenarios, EUDDA appears as the most energy-efficient solution among these three solutions. The following paragraphs explain why EUDDA outperforms the other solutions even in a strict delay requirement scenario.

As previously mentioned that EUDDA's DRAPS algorithm uses FL-SIMP for strict delay requirement case. It is worth to mention here that the average sleep interval length of an ONU under FL-SIMP is smaller than that of ATS and FTS, in which an ONU's sleep interval length increases exponentially up to a T_{max} value. Consequently, energy saving performance in FTS and ATS is supposed to be superior than that of our proposed solution (EUDDA). However, interestingly, looking at the results presented in Fig. 13(a) and (b), we can realize that EUDDA can still provide better energy saving performance compared to other two solutions. The reason behind this is that EUDDA's sleep mode associated control message exchange policy plays significantly important role in reducing number of control messages, thereby minimizing energy consumption ([25] explains how number of control messages could affect energy-efficiency of an ONU with sleep mode enabled). EUDDA's PSS algorithm, which runs on both the OLT and ONU side, contributes in reducing sleep mode associated control message exchange between the OLT and ONU significantly (see Fig. 8). Additionally, similar to [6, 7], an ONU in EUDDA does not exchange any control message with the OLT as long as it is in sleep mode. This is possible because, the OLT always knows an ONU's sleep pattern.

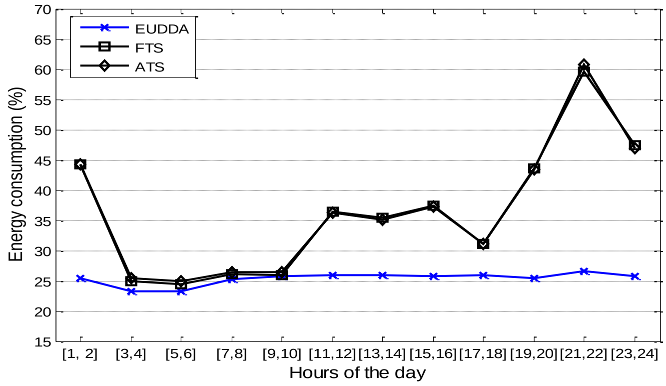
Note that, unlike ATS and FTS, EUDDA does not use early wake-up mechanism. An ONU should leave *Sleep state* instantly to make uplink bandwidth request on arrival of uplink traffic when it uses early wake-up mechanism [5, 7, 13, 18, 19]. Note that every time an ONU leaves *Sleep state*, it spends 2 ms for transition [11, 13] (an ONU's energy consumption during this transition is the same as in the *Active state* [17]). The frequency of occurring early wake-up rises during high arrival region of uplink traffic of a day. Consequently, an ONU in ATS and FTS ends up spending large portion of energy due to spending significant amount of time in transition. By contrast, proposed EUDDA takes into consideration both uplink and downlink traffic delay requirement, and arrival rate while deciding on sleep mode associated parameters for an ONU. This contributes not only to meet both uplink and downlink traffic delay requirement but also to minimize energy consumption of ONUs.

4.3.2. Energy Consumption Performance in a Relaxed Delay Requirement Scenario

In this part, we explain energy consumption performance results of those three solutions under a relaxed delay requirement scenario. Figure 14 depicts energy consumption performance when $D_R = 40\,ms$. As this



(a) $D_R = 5\ ms$.



(b) $D_R = 10\ ms$.

Fig. 13: ONU energy consumption in different solutions compared to AA solution under strict delay requirement scenarios.

is a relaxed delay requirement scenario, EUDDA aims at saving energy as much as possible. Note, EUDDA's DRAPS algorithm chooses EI-SIMP, in which an ONU's sleep interval length exponentially increases, for a relaxed delay requirement scenario. Therefore, all these three solutions rely on Eq. (1) to measure sleep interval lengths of an ONU (in all three solutions, an ONU's sleep interval length grows exponentially from a T_{min} to T_{max} value). Looking into Fig. 14, we can observe that an ONU can reduce energy consumption significantly when proposed EUDDA is in place. Energy consumption in EUDDA ranges between 18% to 23% of AA solution, in which an ONU remains active (on) always. Whereas, energy consumption in ATS and FTS account for approximately 19% to 36% and 24% to 60%, respectively. Global energy consumption results relative to AA solution in this relaxed delay requirement case are: 19.3% for EUDDA, 25.7% for ATS and 36.4% for FTS.

When EI-SIMP is imposed, the main role of DRAPS algorithm is to find a suitable value T_{min} and T_{max} that can maximize ONU's energy saving as much as possible without violating traffic delay requirement. Proposed DRAPS seeks to find a large value of T_{min} in high traffic arrival region ($\lambda > \lambda_{Th}$), so that an ONU can sleep longer. And consequently, number of idle listening time and transition period from *Sleep state* of the ONU reduces, thereby improving energy saving performance. On the other hand, it sets $T_{max} = T_{max\_Threshold}$. ATS does the same in high traffic arrival region.

Conversely, in low traffic arrival region ($\lambda \leq \lambda_{Th}$), ATS seeks to find a suitable T_{max} value, setting $T_{min} = T_{min\_Threshold}$ ($T_{min\_Threshold}$ is the lowest possible value of T_{min}). Indeed, selecting sleep interval length of an ONU from $T_{min\_Threshold}$, which is usually a small value ([7] and [25] assume 3 ms), is definitely expending ONU's energy unnecessarily in a relaxed delay requirement scenario like $D_R = 40\ ms$. Note that the smaller the value T_{min}, the smaller the sleep interval lengths an ONU can have (see Eq. (1)), and thus the higher the

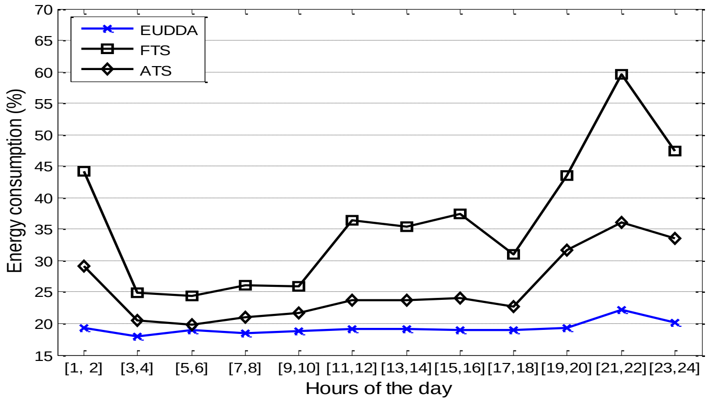


Fig. 14: ONU energy consumption in different solutions compared to AA solution under a relaxed delay requirement scenario.

idle listening time and transition period from *Sleep state* (during idle listening time and transition period an ONU spends energy unnecessarily). Therefore, unlike ATS, EUDDA's DRAPS algorithm seek to find a large value of T_{min} in low traffic arrival region also. Additionally, it finds a suitable value of T_{max} to restrict growth of an ONU's sleep interval length up to a certain point, so that traffic delay requirement is not violated. By contrast to ATS and EUDDA, the value of T_{min} and T_{max} remain always same in FTS. Consequently, there is no energy saving performance improvement in FTS even in a relaxed delay requirement scenario (see Fig. 14). Additionally, not surprisingly, energy saving performance of both FTS and ATS is adversely affected due to early wake-up mechanism in this relaxed delay requirement scenario.

5. Conclusion

It is practical to consider that a TDM-PON system should give more importance in meeting traffic delay requirement over its energy saving performance. It is because a PON operator cannot sacrifice traffic QoS for the sake of improving network energy-efficiency. Therefore, we aimed at developing a solution that not only meets traffic delay requirement but also saves energy. Our proposed EUDDA has two sleep mode management polices: FL-SIMP and EI-SIMP. These are selected based on operator's imposed delay requirement for an ONU. Towards this end, proposed EUDDA appears as a very energy-efficient solution while meeting traffic delay requirement. EUDDA can meet two contradictory goals (energy saving and satisfying delay requirement) because it uses a novel DRAPS algorithm, which selects appropriate sleep mode associated parameters for ONUs under two different sleep mode management policies (FL-SIMP and EI-SIMP). Aside from that, EUDDA follows novel sleep mode associated control message exchange approach with the objective of reducing number of control messages exchange between the OLT and ONUs, thus maximizing energy saving. Performance evaluation of EUDDA shows that it can meet delay requirement of 100% of uplink and downlink traffic under both strict and relaxed delay requirement scenarios. Moreover, the overall global energy saving performance shows that EUDDA consumes 33.6% and 19.3% of AA solution, in which an ONU remains always active (on), in a strict and relaxed delay requirement scenario, respectively.

Acknowledgment

This work was supported by grant No. EEWS-2014-N01140044 from Climate Change Research Hub Project of the KAIST EEWS Research Center (EEWS: Energy, Environment, Water and Sustainability).

References

[1] R. Hodges, W. White., Go Green in ICT, in: NASCIO White Paper, 2008.

[2] M. Gruber, O. Blume, D. Ferling, D. Zeller, M. Imran, E. Strinati, EARTH - Energy aware radio and network technologies, in: Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on, 2009, pp. 1–5.
[3] A. Nikoukar, I.-S. Hwang, C.-J. Wang, M. S. Ab-Rahman, A. T. Liem, A SIEPON based transmitter sleep mode energy-efficient mechanism in EPON, Optical Fiber Technology 23 (2015) 78 – 89.
[4] P. Chowdhury, M. Tornatore, S. Sarkar, B. Mukherjee, Building a Green wireless-optical broadband access network (WOBAN), Lightwave Technology, Journal of 28 (16) (2010) 2219–2229.
[5] S. S. Newaz, Á. Cuevas, G. M. Lee, N. Crespi, J. K. Choi, Improving energy saving in time-division multiplexing passive optical networks, Internet Computing, IEEE 17 (1) (2013) 23–31.
[6] J. Zhang, N. Ansari, Toward energy-efficient 1G-EPON and 10G-EPON with sleep-aware MAC control and scheduling, Communications Magazine, IEEE 49 (2) (2011) s33–s38.
[7] S. S. Newaz, Á. Cuevas, G. M. Lee, N. Crespi, J. K. Choi, Adaptive delay-aware energy efficient TDM-PON, Computer Networks 57 (7) (2013) 1577 – 1596.
[8] IEEE, 802.3ah-2004 - IEEE standard for information technology– local and metropolitan area networks– part 3: CSMA/CD access method and physical layer specifications amendment: Media access control parameters, physical layers, and management parameters for subscriber access networks (2004).
[9] J. Baliga, R. Ayre, K. Hinton, W. V. Sorin, R. S. Tucker, Energy consumption in optical IP networks, Journal of Lightwave Technology 27 (13) (2009) 2391–2403.
[10] S.-W. Wong, L. Valcarenghi, S.-H. Yen, D. Campelo, S. Yamashita, L. Kazovsky, Sleep mode for energy saving pons: Advantages and drawbacks, in: GLOBECOM Workshops, 2009 IEEE, 2009, pp. 1–6.
[11] D. Ren, H. Li, Y. Ji, Power saving mechanism and performance analysis for 10 Gigabit-class passive optical network systems, in: Network Infrastructure and Digital Content, 2010 2nd IEEE International Conference on, 2010, pp. 920–924.
[12] L. Zhang, Y. Liu, L. Guo, X. Gong, Energy-saving scheme based on downstream packet scheduling in ethernet passive optical networks, Optical Fiber Technology 19 (2) (2013) 169 – 178.
[13] R. Kubo, J.-i. Kani, H. Ujikawa, T. Sakamoto, Y. Fujimoto, N. Yoshimoto, H. Hadama, Study and demonstration of sleep and adaptive link rate control mechanisms for energy efficient 10G-EPON, Optical Communications and Networking, IEEE/OSA Journal of 2 (9) (2010) 716–729.
[14] R. Kubo, J. ichi Kani, Y. Fujimoto, N. Yoshimoto, K. Kumozaki, Adaptive power saving mechanism for 10 gigabit class pon systems, IEICE Transactions on Communications E93.B (2) (2010) 280–288.
[15] L. Zhang, Y. Liu, L. Guo, X. Gong, Energy-saving scheme based on downstream packet scheduling in ethernet passive optical networks, Optical Fiber Technology 19 (2013) 169 – 178.
[16] GPON power conservation, ITU-T Recommendations –Supplement 45 (G.sup 45) (May 2009).
[17] S. Herreria-Alonso, M. Rodriguez-Perez, M. Fernandez-Veiga, C. Lopez-Garcia, On the use of the doze mode to reduce power consumption in EPON systems, Lightwave Technology, Journal of 32 (2) (2014) 285–292.
[18] IEEE Standard for Service Interoperability in Ethernet Passive Optical Networks (SIEPON), IEEE Std 1904.1-2013 (2013) 1–834.
[19] ITU-T Recommendation G.988, ONU management and control interface (OMCI) specification (2012).
URL `https://www.itu.int/rec/T-REC-G.988/en`
[20] Y. Yan, L. Dittmann, Energy efficiency in Ethernet passive optical networks (EPONs): Protocol design and performance evaluation, Journal of Communications 6 (2011) 249–261.
[21] J.-H. Jung, W.-H. Yang, Y.-C. Kim, A bandwidth allocation algorithm for improving QoS in EPON with sleep mode, in: Information Networking (ICOIN), 2012 International Conference on, 2012, pp. 324–328.
[22] F. Y. M. Alaelddin, S. S. Newaz, J. Lee, M. R. Uddin, G. M. Lee, J. K. Choi, Performance analysis of TCP traffic and its influence on ONUs energy saving in energy efficient TDM-PON, Optical Fiber Technology 26, Part B (2015) 190 – 200.
[23] A. Dhaini, P. han Ho, G. Shen, Toward green next-generation passive optical networks, Communications Magazine, IEEE 49 (11) (2011) 94–101.
[24] Y. Yan, S.-W. Wong, L. Valcarenghi, S.-H. Yen, D. Campelo, S. Yamashita, L. Kazovsky, L. Dittmann, Energy management mechanism for Ethernet passive optical networks (EPONs), in: IEEE International Conference on Communications (ICC), 2010, pp. 1–5.
[25] S. S. Newaz, Á. Cuevas, G. M. Lee, N. Crespi, J. K. Choi, Evaluating energy efficiency of ONUs having multiple power levels in TDM-PONs, Communications Letters, IEEE PP (99) (2013) 1–4.
[26] Y. Xiao, Energy saving mechanism in the IEEE 802.16e wireless MAN, Communications Letters, IEEE 9 (7) (2005) 595–597.
[27] Smart Choices for the Smart Grid, alcatel/Lucent Technology White Paper, (Feb. 2011).
URL `http://www.smartgrids-cre.fr/media/documents/Alcatel-Lucent_SmartChoicesfortheSmartGrid_TWP_EN.pdf`
[28] G. Sankaran, K. Sivalingam, ONU buffer elimination for power savings in passive optical networks, in: IEEE International Conference on Communications (ICC), 2011, pp. 1–5.
[29] B. Skubic, D. Hood, Evaluation of ONU power saving modes for gigabit-capable passive optical networks, Network, IEEE 25 (2) (2011) 20–24.
[30] N. Radzi, N. Din, M. Majid, M. Al-Mansoori, The bandwidth and jitter study in EPON upstream traffic, in: Research and Development (SCOReD), 2013 IEEE Student Conference on, 2013, pp. 171–174.
[31] B. Dusza, C. Wietfeld, Performance evaluation of IEEE 802.16e mobile WiMAX for long distance control of UAV swarms, in: Wireless Information Technology and Systems (ICWITS), 2010 IEEE International Conference on, 2010, pp. 1–4.
[32] M. Marchese, QoS-oriented Technologies, John Wiley & Sons, Ltd, 2007, pp. 15–44.