

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/321057119>

# Formal Methods for Energy-Efficient EPONs

Article · November 2017

DOI: 10.1109/TGCN.2017.2772832

CITATIONS

7

READS

138

3 authors:



**Sophia G. Petridou**

University of Macedonia

54 PUBLICATIONS 419 CITATIONS

[SEE PROFILE](#)



**Stylianos Basagiannis**

United Technologies Research Center - UTRC Ireland

43 PUBLICATIONS 362 CITATIONS

[SEE PROFILE](#)



**Lefteris Mamas**

University of Macedonia

100 PUBLICATIONS 882 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



IoT Mobility research [View project](#)



Cross-Layer Control of Data Flows (CORAL) - WISHFUL H2020 Open Call 2 [View project](#)

# Formal Methods for Energy-Efficient EPONs

Sophia Petridou, *Member, IEEE*, Stylianos Basagiannis, *Member, IEEE*, and Lefteris Mamatras, *Member, IEEE*

**Abstract**—Energy-efficient Passive Optical Networks (PONs) have gained significant interest since they are estimated to be the largest energy consumers among the wired access networks for the next ten years. In Ethernet PONs (EPONs), the equipment placed at the customer premises, i.e., the Optical Network Units (ONUs), has been shown to be responsible for almost 65% of the total EPON power consumption. Sleep mechanisms, implemented at ONUs' side, can contribute to the EPONs' energy-efficiency. However, the trade-off between energy saving and Quality of Service (QoS) requirements should be carefully tuned, especially when the downstream transmission is considered, to achieve the desirable results. In this paper, we propose a general framework that exploits formal methods as an approach for EPONs' energy-efficiency. The idea is to build a holistic model representing both the state machine of the ONU in its details as well as the ONU's communication with the Optical Line Terminal (OLT) under EPONs' specifications. Verification results reveal that, guided by the QoS constraints and considering the transitions' cost, an "aggressive" policy defining long sleep periods of the 100 ms order can be up to 24,9% beneficial for energy-efficiency compared to a "conservative" policy with sleep periods of the 10 ms order.

**Index Terms**—Energy-efficiency, Ethernet Passive Optical Networks, fast sleep, formal methods.

## I. INTRODUCTION

Energy consumption in data communication networks has attracted significant research attention due to the tremendously increasing number of broadband users [1]–[3]. While traditionally, the discussion of energy efficiency was associated with wireless networks and, especially, with wireless sensor networks, due to the battery-powered devices [4], recently, such an issue has also been raised for wired access networks, i.e., the last segment of the connection that links the service provider's central office (CO) to end users [5].

Among various access technologies including xDSL-based wired access, fiber to the node (FTTN), and point-to-point optical access networks, Passive Optical Networks (PONs) is a dominant one; it can support a broad range of applications including triple play (voice, data and video services) over a single fiber, whilst it consumes less energy per transmitted bit due to the proximity of optical fiber to the end users and the passive nature of the remote nodes [5]–[9]. Nevertheless, further energy-consumption reducing is required since it is estimated that PONs, such as Ethernet PON (EPON) systems, will be the largest energy consumers among the wired access networks for the next ten years [5].

Within EPONs, the Optical Network Units (ONUs) are the most energy-consuming devices, as they are responsible for

almost 65% of the total EPON power consumption [5]. A common approach to improve EPONs energy efficiency is to turn off (completely or partially) the ONUs, which are rarely used at their full potential, in a cyclic manner, known as fast sleep [10]–[13]. According to [13], [14], the ONU's transmitter and receiver can switch to sleep state separately, with the sleep control of transmitters to be relatively easier than that of receivers, because the arrival of the upstream traffic can trigger them to wake up. However, the broadcast nature of the EPON downstream transmission entails that an ONU has to receive and check all downstream packets, even when these are not destined to itself. Thus, a sleep mechanism for receivers is challenging; the ONU could not waste energy receiving broadcast data, but if its receiver switches to sleep state due to the absence of downstream traffic, it should wake up once the Optical Line Terminal (OLT) has traffic for it.

A discussion of sleep mechanisms for EPONs' energy-efficiency which are based on ONUs' fast sleep follows. Two interesting tasks in such mechanisms are to determine the conditions that trigger the sleep state, as well as the sleep state duration [15], [16].

### A. Related Work

SIEPON (Service Interoperability in Ethernet Passive Optical Networks) standard [17] describes mechanisms and protocols for reducing the ONU power consumption based on power saving states. However, the parameters that determine the conditions that trigger the sleep state and the sleep state duration are outside the scope of the standard, leaving room for further investigation. In the literature, there have been approaches focused on the physical or data link layer and some of them target both layers via hybrid solutions [10]. The schemes that operate on the data link layer are either OLT or ONU-triggered based on a control message flow [11], [18], [19] or self-triggered based on mutual OLT and ONU consent [13], [20].

Energy-saving solutions of the latter case have the advantage of minimum or no overhead, since they address the downstream challenge through the modification of the bandwidth allocation (BA) algorithms running at both sides of communication. Lee *et al.* [21] propose fixed bandwidth allocation (FBA) for the downstream traffic under light network load. Using FBA, the ONUs are allocated with fixed and a priori known time slots. Thus, they can transit to the sleep state during the time slots assigned to the rest ONUs, without any OLT notification. However, FBA suffers from bandwidth under- or over-allocations. Yan *et al.* [22] provide the solution of simultaneously scheduling the downstream and upstream traffic, since the receivers and transmitters operate at different wavelengths. The idea is that an ONU can receive

S. Petridou and L. Mamatras are with the Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece, e-mail: {spetrido, emamatras}@uom.edu.gr.

S. Basagiannis is with United Technologies Research Centre, Cork, Ireland, email: basagis@utrc.utc.com.

its downstream traffic over the time slots that its upstream traffic is scheduled and they are assigned to it by the OLT. While this scheme works efficiently under symmetric traffic, it deteriorates bandwidth utilization when downstream traffic outweighs upstream traffic (e.g., downloading, broadcast TV).

In [13], [20], Zhang *et al.* achieve high bandwidth utilization by exploiting dynamic bandwidth allocation (DBA). They use a semi-Markov chain and consider three possible states for the ONU's operation, namely the active, listen and sleep state. In their solution, the ONU's receiver transits from the active to the listen and, then, from the listen to the sleep state based on the downstream traffic scheduling performed by the OLT. In particular, if the ONU does not receive downstream traffic for a time period  $x$ , it transits to the sleep state for a time period  $y$ . The ONU can interrupt the listen period  $x$  because of incoming downstream traffic, but once it transits to the sleep state, it should wait for the sleep period  $y$  to be expired.

A major issue is that this solution handles the listen and sleep periods, represented as  $x$  and  $y$ , in quite a conservative way. In particular, they prioritize QoS constraints, e.g., packets' delay, independent of the traffic identity, e.g., delay-insensitive traffic [23]–[25]. However, the trade-off between QoS requirements and desirable energy conservation results into shrinking energy saving even when this is not required. Another issue is that the analysis of Zhang *et al.* [13], [20] does not consider the ONU's time-recovery overhead. Typically, the ONU's transition from the sleep to the active state (wake-up) is not instantaneous, but it requires quite a long time ( $\sim 2ms$ ). This overhead is twofold: the ONU must recover the OLT clock and regain the network synchronization [12].

In practice, assuming that transition time from the sleep to the active state is negligible leads to a coarse view of the time issue with impact both on time-related results, e.g., delay, and on devices' lifetime. Ignoring, for example, the ONU's time-recovery overhead advocates an energy-aware mechanism with short sleep periods and, thus, frequent power-state transitions which result in temperature variations that impact the ONU's lifetime. The authors in [26], [27] elaborate on the impact of the deployed sleep mechanism combined with the device lifetime in optical networks. Actually, not only device lifetime constraints but also maintenance costs, derived from the application of different power states, could be defined as cost-benefit metrics, similarly to [28], during the design of an energy efficient mechanism for EPONs.

Nowadays, these ideas could be combined with Software-Defined Networking (SDN) strategies, relevant to [29], [30], that receive feedback regarding applications' requirements and tune the energy-saving configuration for each ONU accordingly (e.g., the listen and sleep periods' duration). Logically-centralized controllers can be communicating with high-level management components, expressing global performance goals for the system, e.g., an Internet service provider may decide to save the maximum energy, while giving main priority to paid services. A thorough investigation of SDN proposals for optical networks can be found in [31]. Such aspects, though interesting, are challenging enough to deserve a stand-alone study.

To sum up, keeping a sleep control mechanism realistic,

physical-layer feedback is necessary, along with a model that will synchronize the ONU's state machine with the EPONs' specifications. This insight inspired us to investigate EPONs' energy-efficiency using formal methods; i.e., an approach that offers an automated mechanism for model-based analysis of systems whose behaviour can be abstractly described. The advantages of formal methods as an analytical approach, along with the proposed analysis novelty are discussed right afterwards.

## B. Contribution

In this paper, we propose a general framework that exploits formal methods as an approach for EPONs' energy-efficiency. The idea is to start with a formal representation of the system under consideration, an EPON in our case, and move on with the full state-space generation and exploration using sound analytical techniques, such as model checking, in order to derive quantitative results regarding the system's properties. The full state-space exploration of the model gives to networks' analysts and designers the advantage of verifying their solutions under a variety of parameters much earlier than simulation or experimentation. Simulation, by definition, starts with an input vector and, then, provides the reference output. This entails that simulation-based approaches can provide performance and energy-aware results, but, since they cannot explore the full state-space of the model is based upon, they might miss critical states where, for example, energy overshoots or performance is downgrading. Experimentation is the most realistic approach but also time- and cost-prohibited. Model checking balances between realism - due to the complete state-space generation - and experimentation complexity.

Formal verification is widely used nowadays in the industrial domain in order to enable system verification early or during the design cycle, providing the necessary guaranties for its correctness. Moreover, verification results spanning from probability to rewards' calculations constitute an ideal choice for a parallel evaluation of both functional (e.g., performance) and non-functional (e.g., energy consumption) requirements of the system. Such a premise - an industrial challenge nowadays - can bridge the gap of multi-disciplinary engineering when, for example, network engineers and protocol designers have to comply to certain system characteristics, often not visible to both of them.

Exploiting formal techniques, our goal is to build a holistic model which represents the ONU state machine, incorporating physical-layer details, and simultaneously synchronizes it with the EPON's specifications, defining by higher-layer details. Towards this goal, probabilistic model checking is used, since it can be applied to analyse systems that exhibit stochastic behaviour, e.g., communication protocols and computer networks [32]–[34]. The authors in [35]–[37] provide a quantitative analysis that depicts CPU cycles' consumption or protocol's transmission cost for energy-constraint devices.

To the best of our knowledge this is the first time a passive optical network is analysed using model checking. The EPON under consideration is modeled as a Continuous-Time Markov Chain (CTMC), hereafter EPON<sub>CT</sub> model, and

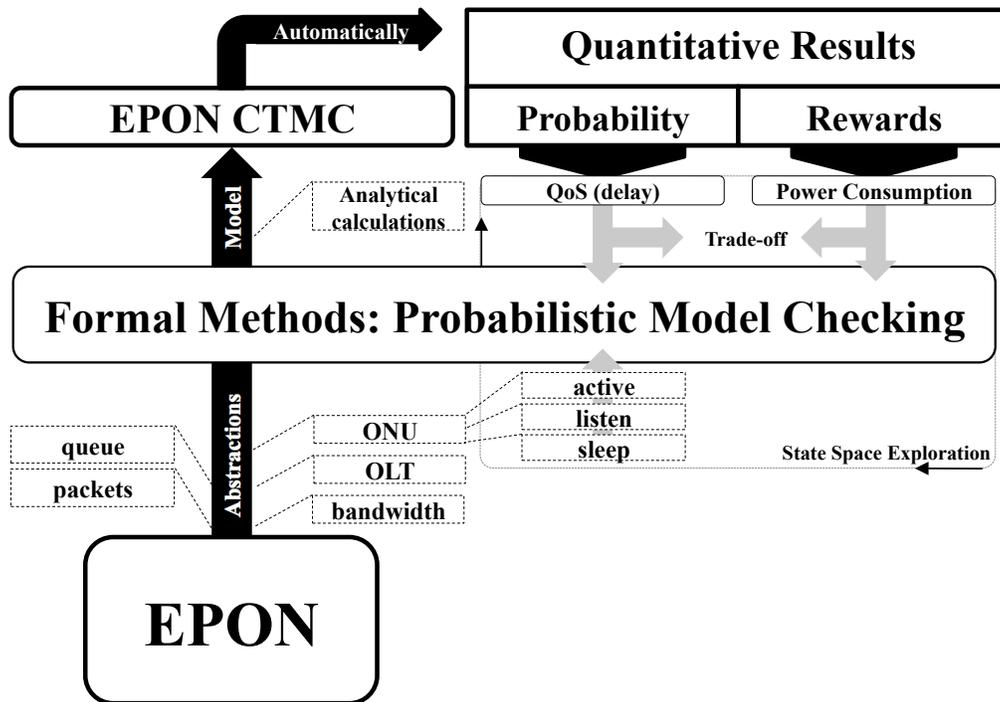


Fig. 1: An abstract view of the proposed analysis

then, its properties expressed as Continuous Stochastic Logic (CSL) formulas [38] are verified. CTMCs are frequently used in performance analysis, since they model continuous real time and probabilistic choice: one can specify the rate of making a transition from one state to another [32]. Probabilistic choice arises through race conditions when two or more transitions in a state are enabled. Thus, CTMCs are amenable to analytical treatment and numerical computation, e.g., ONU's power consumption over a finite time period.

Our main idea (first presented in [39]) can be generalized according to Fig. 1 and described as follows. The initial step was to model the ONU finite state machine. Apart from the three power states considered, namely the active, listen and sleep, the EPON<sub>CT</sub> model elaborates on transitions rates, derived by the physical-layer oriented bibliography [11], [12], [40], and especially on the wake-up delay which, according to [12], [15], should be taken into account by a sleep mechanism. The precision in transitions' rates highlights the impact of the listen and sleep periods' duration. Intuitively, short sleep periods are preferable when QoS requirements are stringent, e.g., schedule of delay-sensitive traffic, but they can be prolonged, guided by the QoS constraints, keeping the energy saving as the primary goal. Our analysis provides quantitative results verifying this intuition.

Next, the challenge was to align the model with the EPONs' specifications. For this purpose, a number of network parameters is considered, e.g., the rate of the downlink channel, the packets' length, the arrival and service rate. In addition, a queue is modeled at the OLT to buffer the packets arriving while the ONU is sleeping. Finally, the EPON<sub>CT</sub> model is augmented with rewards (or "costs") expressing the power consumption at each ONU state. This way, energy-related

results can be derived.

The novelty of the proposed work is summarized as follows:

- it exploits formal methods and especially model checking as a means of tuning the trade-off between energy saving and QoS requirements, e.g., packets' delay, in an EPON,
- it combines the ONU's state machine with the EPONs' specifications to build a configurable CTMC model,
- it demonstrates the impact of the listen and sleep periods on the aforementioned trade-off and specifies their duration in an EPON case study.

The remainder of this paper is organized as follows. Section II explains formal verification preliminaries emphasizing on CTMCs primitives. The proposed EPON<sub>CT</sub> model is described in Section III, while the results of the analysis are discussed in Section IV. Finally, Section V concludes the paper with some future work thoughts.

## II. FORMAL VERIFICATION FOR EPONs' ENERGY ANALYSIS

Model checking is a popular approach towards the validation and verification of computer-based systems [41], [42]. Its power is based upon a fully automated technique of verifying that the system under consideration - an EPON in our case - will meet its requirements, e.g. in terms of energy-efficiency or QoS. Typically, given a system model  $M$ , probabilistic model checking, which is a quantitative formal verification technique, proceeds to a systematic exploration of all generated states of  $M$  to verify its desirable reachability properties defined as  $\phi$  [42]. This way, an EPON sleep mechanism can be quantitatively verified through automatically checking all reachable states of its corresponding model  $M$ , developed in line with

TABLE I: CTMC symbols' notation

Symbol	Description
$M$	a continuous-time Markov chain
$S$	finite set of states
$\bar{s}$	initial state
$R$	transition rate matrix
$L$	labelling function
$\omega$	path in a CTMC
$E(s)$	exit rate of state $s$
$P(s, s')$	probability of $(s, s')$ transition
$Q$	infinitesimal generator matrix
$P^{u(M)}$	uniformized probability of a CTMC
$q$	uniformization rate
$\Pi_t$	transient probabilities' matrix at time instant $t$
$\gamma$	Poisson probability
$\underline{r}$	vector of state rewards
$\iota$	matrix of transition rewards
$R_{p \rightarrow r}[I^=t]$	instantaneous reward
$R_{p \rightarrow r}[C^{\leq t}]$	cumulative reward

the network's specifications and, then, enhanced with energy-aware parameters.

An EPON is a continuous real-time system mastered by packets' exchange between an OLT and ONU(s) over a fibre-optic line; and the sleep mechanism implemented at the ONUs' side implies a Markov chain functionality. Thus, from a probabilistic model checking perspective, CTMC is a suitable choice, since it forms a labelled transition system augmented with rates [32]. For the proposed EPON<sub>CT</sub> model the ONU's transition rates among its states, i.e., active, listen and sleep, and the packets' arrival and service rates are perfectly matched with the CTMCs primitives. A relevant model for dynamic power management of a Fujitsu disk drive is also using CTMC [43].

### A. Probabilistic Model Checking Primitives

This section discusses the necessary theoretical background and the approach followed to probabilistic model checking for energy-efficient EPONs. CTMC principles are also described, since we proceed with continuous instead of discrete time modeling, as it has been previously justified.

In theory, a CTMC contains a discrete state space  $S$  with transitions between its states. A transition is assigned with a rate of occurrence, i.e., the transition's rate. Analytically, a CTMC is defined as a tuple  $M = (S, \bar{s}, R, L)$  [32], where:

- $S$  is a finite set of states
- $\bar{s} \in S$  is the initial state
- $R : S \times S \rightarrow \mathfrak{R}_{\geq 0}$  is the transition rate matrix and
- $L : S \rightarrow 2^{AP}$  is the labelling function of atomic propositions  $AP$  that are true in  $S$ .

CTMCs have finitely many states that are discrete, a time parameter that ranges over  $\mathfrak{R}_{\geq 0}$ , and they do not allow non-determinism [44]. A path  $\omega$  in a CTMC is a non-empty sequence  $s_0 t_0 s_1 t_1 s_2 \dots$ , where, for all  $i \geq 0$ , there is a state  $s_i \in S$  for which  $R(s_i, s_{i+1}) > 0$  and  $t_i \in \mathfrak{R}_{> 0}$ . The value  $t_i$  denotes the amount of time spent in the state  $s_i$ , while  $\omega(k)$  represents the  $k$ th state of the path  $\omega$ , i.e.,  $s_k$ . The duration, i.e.,  $t_i$ , that the modeled system will reside on a specific state will follow an exponential distribution. Notation summary is given in Table I.

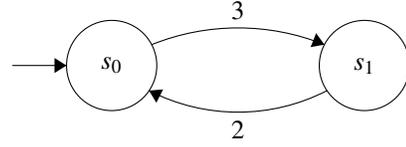


Fig. 2: A comprehension example of the CTMC principles

Provided that the proposed EPON<sub>CT</sub> follows the aforementioned primitives and generates its state space, the probability of taking a transition from a state  $s$  within  $t$  time units will be  $1 - e^{-E(s) \cdot t}$ .  $E(s)$  defines the exit rate of the state  $s$ ; in cases where  $R(s, s') > 0$  for more than one state  $s'$ ,  $E(s)$  will be equal to the sum of the rates leading to  $s'$ , i.e.,  $E(s) = \sum_{s' \in S} R(s, s')$ . In the latter case, a race condition exists and the probability of triggering the transition  $(s, s')$ , i.e.,  $P(s, s')$ , is defined by the embedded-Discrete Time Markov Chain (emb.DTMC) as follows [44]:

$$P(s, s') = \begin{cases} \frac{R(s, s')}{E(s)} & \text{if } E(s) > 0 \\ 1 & \text{if } E(s) = 0 \text{ and } s = s' \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Based on the transition rate matrix  $R$ , the infinitesimal generator matrix  $Q$  is defined as:

$$Q = \begin{bmatrix} -\sum_{s \neq 1} R(1, s) & R(1, 2) & \dots & R(1, s') \\ R(2, 1) & -\sum_{s \neq 2} R(2, s) & \dots & R(2, s') \\ \vdots & \vdots & \ddots & \vdots \\ R(s', s) & R(s', 2) & \dots & -\sum_{s \neq s'} R(s', s) \end{bmatrix}$$

where  $s'$  is the residing state and  $s \in S$ . Both transient probabilities defined in Eq. 1 and infinitesimal generator matrix  $Q$  are automatically calculated in probabilistic model checking; this is a clear advantage over a manual analytical approach.

Once the full state space has been generated, transient probabilities are calculated based on all possible paths leading from an initial state  $s_0$  to  $s'$  at a particular time instant  $t$ , i.e.,  $s_0 \rightarrow \Pi_t^M(s')$ . However, due to the high computational complexity of producing the transient probabilities' matrix  $\Pi_t$ , model checking proceeds with the uniformization of  $\Pi_t$ . According to [44], the uniformized probability of a CTMC is expressed as  $P^{u(M)} = I + \frac{Q}{q}$ , where  $q$  is the uniformization rate, i.e.,  $q \geq \max\{E(s) : s \in S\}$ , and  $I$  is the identity matrix. Eventually,  $\Pi_t = \sum_{i=0}^{\infty} \gamma(i, q \cdot t) \cdot [P^{u(M)}]^i$ , where  $\gamma(i, q \cdot t)$  is the  $i$ th Poisson probability with parameter  $q \cdot t$  [45];  $i$  expresses the steps occurring in time instant  $t$  given each step has a delay exponentially distributed with rate  $q$ .

To align these principles with the probabilistic analysis of the EPON<sub>CT</sub>, let us consider a proof of concept  $M$ , as the one depicted in Fig. 2. This model could be a two-state machine of an ONU with its receiver either in active, i.e.,  $s_0$ , or in sleep state, i.e.,  $s_1$ . The ONU starts in active state, and then it can transit to the sleep state with a rate  $R(s_0, s_1) = 3$ . While

TABLE II: EPON<sub>CT</sub> symbols' notation

Symbol	Description
$C$	rate of the downlink channel (1.25 Gbps)
$packets$	number of transmitted packets
$l$	packets length (1518 bytes)
$\lambda$	packet arrival rate
$\mu$	packet service rate ( $C/l$ )
$R_{l2s}$	rate of listen to sleep transition (1000/2.88)
$R_{s2l}$	rate of sleep to listen transition (1/2)
$R_{s2a}$	rate of sleep to active transition (1/2)
$d_{listen}$	listen period duration ( $d_{listen} = 1/R_{l2l}$ )
$d_{sleep}$	sleep period duration ( $d_{sleep} = 1/R_{s2s}$ )
$M_i$	module inside the model, $i = \{olt, q, onu\}$

in sleep state, it can head back to active state with a rate  $R(s_1, s_0) = 2$ . For this model  $M$ , the rate transition matrix  $R$  and the infinitesimal generator matrix  $Q$  will be:

$$R = \begin{bmatrix} 0 & 3 \\ 2 & 0 \end{bmatrix}$$

and

$$Q = \begin{bmatrix} -3 & 3 \\ 2 & -2 \end{bmatrix}$$

The probability distribution matrix for the uniformized DTMC (given a uniformization rate  $q = 3$ ) for the above  $M$  will be:

$$P^{u(M)} = \begin{bmatrix} 0 & 1 \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix}$$

Provided an initial distribution for the active state  $s_0$  at time  $t = 0$ , e.g.,  $\underline{\pi}(0, 0) = [1, 0]$ , the transient probabilities at time instant  $t = 1$  will be:

$$\begin{aligned} \underline{\pi}(0, 1) &= \sum_{i=0}^{\infty} \gamma(i, q \cdot t) \cdot \underline{\pi}(0, 0) \cdot [P^{u(M)}]^i \\ &= \gamma(0, 3) \cdot [1, 0] \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \gamma(1, 3) \cdot [1, 0] \cdot \begin{bmatrix} 0 & 1 \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix} + \\ &\quad \gamma(2, 3) \cdot [1, 0] \cdot \begin{bmatrix} 0 & 1 \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix}^2 + \dots \\ &\approx [0.40404, 0.59596] \end{aligned}$$

This outcome entails that starting from the active state  $s_0$ , the model remains at the same state with 0.4% probability after 3 time units, while it transits to sleep state  $s_1$  with probability 0.6. The higher probability of active to sleep transition is in line with the higher rate of taking this transition, i.e.,  $R(s_0, s_1) = 3$ , instead of remaining in active state  $s_0$ , i.e.,  $R(s_0, s_0) = 1$ , as depicted in Fig. 2 (rates equal to 1 are typically omitted from the state machine).

Compared to  $M$ , the proposed EPON<sub>CT</sub> considers a three-state machine for the ONU, since its receiver can transit among active, listen and sleep states. However the challenge in the ONU's state machine defined in EPON<sub>CT</sub> was to synchronize the rates among the three states with the rates defined by the EPON's specifications. In practice, the matrix  $R$  of the EPON<sub>CT</sub> model is calculated in line with the rates  $\lambda, \mu, R_{l2l}, R_{l2s}, R_{s2s}, R_{s2l}, R_{s2a}$ ;  $\lambda, \mu$  are network-related rates, and  $R_{s2s'}$  are rates associated with the state machine, since they express the rate to transit from state  $s$  to state  $s'$ . Table II gives a summary of rates' notation, while modeling aspects including these rates are discussed in Section III.

TABLE III: ONU power consumption at different states [12]

	active	listen	sleep
power consumption	3.85W	1.28W	0.75W

Having the full state-space of EPON<sub>CT</sub>, model checking contacts probabilistic analysis which allows us to derive quantitative results computing the probability that some behaviour of our model is observed (Fig. 1), e.g., the probability that the ONU will be in sleep state. However, to verify our model and calculate probabilities of residing successfully to a final state, it is apparent that the above analysis needs to be automated. Formal verification compared to a manual analytical approach provides the principles of analysing a system with stochastic behaviour, taking into account a variety of parameters that keeps the model close to the real system and all its possible behaviours.

### B. Energy-Aware Properties

As depicted in Fig. 1, besides probabilistic results, model checking enriches the state space with weights which enables the outcome of results related to network metrics, e.g., the expected power consumption and packets' delay. Typically, a CTMC model can be augmented with *rewards* structures, which assign real-valued quantities to the model states or transitions, e.g., either "bonus" or "cost". A reward structure is a tuple  $(\underline{q}, \iota)$  [32], where:

- $\underline{q} : S \rightarrow \mathfrak{R}_{\geq 0}$  is a vector of state rewards, and
- $\iota : S \times S \rightarrow \mathfrak{R}_{\geq 0}$  is a matrix of transition rewards.

Let us consider that a random variable  $X$  denotes the reward structure (or function) which assigns "costs" to a model's state or transition, e.g., the ONU is defined to consume 3.85 W while in the active state inside the EPON<sub>CT</sub> [12]. Then,  $X = \text{energy}$  and the expected reward  $Exp(X)$ , defined as:

$$Exp(X) = \sum_{\omega \in \Omega} X(\omega) \cdot P(\omega) \quad (2)$$

expresses the quantity calculated upon  $X$  for all state space's paths  $\omega$  and the probabilities  $P(\omega)$  assigned to them. In the aforementioned example,  $Exp(X)$  would provide the "expected power consumption" while the ONU is in active state, once the whole state space of EPON<sub>CT</sub> has been built.

Among four different types of reward properties supported in CTMC models [32], we employ:

- *Instantaneous*  $R_{\bowtie r} [I^=t]$ : the expected value of the reward at time-instant  $t$  is  $\bowtie r$ , where  $\bowtie \in [\leq, <, \geq, >]$ . Starting from a state  $s$ , the reward at time  $t$  will be given by the recursive function:  $Exp(s, X_{i=t}) = P^{u(M)} \cdot Exp(s, X_{i=t-1})$ , where  $Exp(s, X_{i=0}) = \underline{q}(s)$ .
- *Cumulative*  $R_{\bowtie r} [C^{\leq t}]$ : the expected reward cumulated up to time-instant  $t$  is  $\bowtie r$ , where  $\bowtie \in [\leq, <, \geq, >]$ . Starting from a state  $s$ , the reward up to time  $t > 0$  will be given by the recursive function:  $Exp(s, X_{C \leq t}) = \underline{q}(s) + \sum_{s' \in S} P^{u(M)}(s, s') \cdot (\iota(s, s') + Exp(s', X_{C \leq t-1}))$ , where  $Exp(s, X_{i=0}) = 0$ .

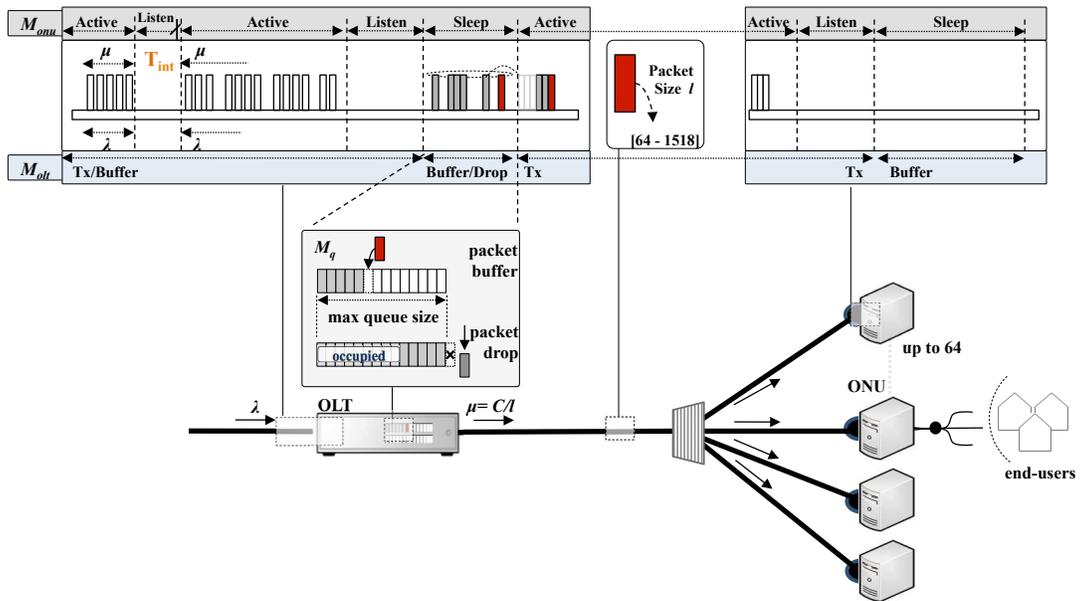


Fig. 3: The components of the EPON<sub>CT</sub> Markov chain model

In our analysis, instantaneous rewards provide the expected packets' delay, while cumulative rewards calculate the expected power consumption. Time property  $t$  for these rewards is typically derived through probabilistic results.

Rewards functionality is precise for the proposed energy-efficiency analysis. In particular, once the EPON<sub>CT</sub> model is designed, it is augmented with two reward structures. The first one is the energy reward structure that “brings” power-consumption “costs” into the model. Table III summarizes the power consumption of the ONU receiver in its different states, as derived by [12]. The delay reward structure “loads” the model's state space with “costs” related to packets' delay, i.e., queuing, transmission and “wait-to-wakeup” delay [13], [20]. The last one expresses the delay that packets face when they arrive and the ONU is sleeping.

Model checking enables the verification of the EPON<sub>CT</sub> model's properties, which are encoded as Continuous Stochastic Logic (CSL) formulas [38]. In CSL, for example,  $\mathcal{P}_{=?}(\phi)$  evaluates the probability of the path formula  $\phi$ .  $R_{\text{prior}}[I^{-t}]$  and  $R_{\text{prior}}[C^{\leq t}]$  are also CSL formulas. Queries  $Q_x$  in Section IV are specific examples of the above  $\mathcal{P}$  and  $R$  properties used to obtain probabilistic and rewards-aware results, respectively, through the full-state space exploration.

### III. EPON MODELING USING CTMC

From the model checking perspective, the ONU state machine and its optical communication with the OLT are modelled as a CTMC within the PRISM model checker [46]. Our EPON<sub>CT</sub> model, comprises three (3) modules, namely  $M = \{M_{olt}, M_q, M_{onu}\}$ . In line with Fig. 3:

- $M_{olt}$  corresponds to the OLT, which broadcasts a number of *packets* and observes the ONU state,
- $M_q$  represents a queue for modeling the packets' arrival, their buffering once they cannot be immediately received by the ONU and their dropping when the queue is full,

TABLE IV: Average transition times for ONU state transitions [11], [12], [40]

current/next	active	listen	sleep
active	-	<i>ns</i>	-
listen	<i>ns</i>	$d_{listen}$	$2.88\mu s$
sleep	$2ms$	$2ms$	$d_{sleep}$

- $M_{onu}$  models the ONU, which receives the packets that have been transmitted. This component may typically have several states of operation, each of which is characterized by a service rate. In general, states with zero service rate are called sleep states.

Each module  $M_i$ ,  $i = \{olt, q, onu\}$ , is defined as a pair of  $(Var_i, C_i)$ , where  $Var_i$  is a set of integer-valued local variables and  $C_i$  is a set of commands which drive the behaviour of the module. Each command  $c \in C_i$  is written as  $[g \rightarrow \lambda_1 : u_1 + \dots + \lambda_n : u_n]$ , composing of: a synchronization label  $[g]$ , a guard  $g$  and a set of pairs  $(\lambda_j, u_j)$ ,  $1 \leq j \leq n$ . The guard  $g$  is a predicate over the set of all local variables  $Var$  and each update  $u_j$  corresponds to a possible transition of module  $M_i$ . In CTMC model specification, the constants  $\lambda_j$  determine the rates attached to the transitions (i.e., they specify the transition time from state  $s$  to  $s'$  as described in Section II) [41]. In our model, we elaborate on different  $\lambda_j$ , where  $\lambda_j \in \{\lambda, \mu, R_{l2l}, R_{l2s}, R_{s2s}, R_{s2l}, R_{s2a}\}$ . Introduced in Section II, Table II provides the model's notation, where  $\lambda$  and  $\mu$  correspond to packets' arrival and service rate, and  $R_{s2s'}$  expresses the transition rate from state  $s$  to  $s'$ .

The EPON<sub>CT</sub> model considers three possible ONU states, namely active, listen and sleep state, inspired by the scheme proposed in [13], [20]. However, contrary to [13], [20], where the transit time from the sleep to the active state is considered to be negligible and, thus, is not taken into account, the proposed analysis elaborates on transition times which are derived by the studies [11], [12], [40] and defined in Table IV.

```

ctmc
// listen and sleep periods' duration in ms
2 Define d_listen, d_sleep;
// transition rates in ms
4 Set Rl2l=1/d_listen;
5 Set Rl2s=pow(10,3)/2.88 ;
6 Set Rs2s=1/d_sleep;
7 Set Rs2l=1/2 ;
8 Set Rs2a=1/2 ;
...
10 module ONU
// model begins with ONU in listen state
12 Set current_state=listen;
...
// transitions from listen state
15 [listen2listen] (current_state=listen) &
(previous_state=active) & (queue=empty) -> Rl2l:
(current_state=listen) & (d_listen=expired);
16 [listen2sleep] (current_state=listen) &
(d_listen=expired) & (queue=empty) -> Rl2s:
(current_state=sleep) & (d_listen=reset);
// transitions from sleep state
18 [sleep2sleep] (current_state=sleep) &
(previous_state=listen) -> Rs2s: (current_state=sleep)
& (d_sleep=expired);
19 [sleep2active] (current_state=sleep) &
(d_sleep=expired) & (queue=empty) -> Rs2a:
(current_state=active) & (d_sleep=reset);
20 [sleep2listen] (current_state=sleep) &
(d_sleep=expired) & (queue=empty) -> Rs2l:
(current_state=listen) & (d_sleep=reset);
...
22 end module
    
```

Listing 1: An indicative example of ONU transitions' modeling in pseudo code

This way we can appropriately configure the parameters that specify the listen and sleep periods, i.e.,  $d_{listen}$  and  $d_{sleep}$ , which significantly affects the trade-off between energy saving and packets' delay. Listing 1, which contains an indicative example of ONU transitions' modeling (using pseudo code), starts with defining  $d_{listen}$  and  $d_{sleep}$  (line 2) that consist two core parameters of the proposed EPON<sub>CT</sub> model. Next, the transition rates  $R_{s2s'}$  are specified (lines 4 – 8) taking into account the information of Table IV.

In detail, at any given time, the ONU can only be in one of the aforementioned three states, shown in Fig. 4 and modeled with the *current\_state* variable (line 12). As depicted in Fig. 3, the ONU receives downstream packets only in the active state with service rate  $\mu$ . Once the ONU does not receive data during a scheduling cycle, it transits to the listen state for a period of  $d_{listen}$  duration. The time period  $d_{listen}$  specifies the rate to stay in the listen state, i.e.,  $R_{l2l} = 1/d_{listen}$ , as defined in line 15 of Listing 1. The transit time from the active to the listen state is of the nanoseconds' order, according to Table IV, and, thus, the rate  $R_{a2l}$  is not considered in the model. In the case of packets' arrival during  $d_{listen}$ , the ONU goes back to the active state in nanoseconds' time (similarly to  $R_{a2l}$ , the rate  $R_{l2a}$  does not considered in the model), otherwise it transits from the listen to the sleep state. In practice,  $d_{listen}$  is the EPON<sub>CT</sub> parameter that determines the condition that triggers the sleep state. A long listen period would deteriorate energy saving, but too short periods would force the ONU to make early transitions to the sleep state increasing the packets' delay.

According to [11], the listen to sleep transition, described in

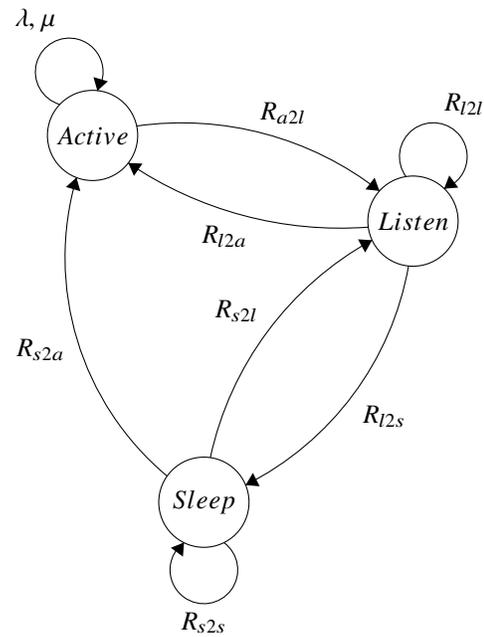


Fig. 4: The ONU finite state machine

line 16, takes  $2.88\mu s$  which is the time required by the ONU to turn off its transceiver. This implies that  $R_{l2s} = 1000/2.88$ . When  $d_{listen}$  expires, the ONU transits to the sleep state for a period of  $d_{sleep}$  duration, which defines the rate to stay in the sleep state, i.e.,  $R_{s2s} = 1/d_{sleep}$  (line 18). During this period, the received downstream packets can either be buffered or ignored. Our model implements a finite queue for packets' buffering. Upon  $d_{sleep}$  expiration, the ONU needs a significant amount of time, i.e.,  $2ms$ , to turn on its transceivers and to synchronize with the network [11], [12], [40]. In case of packets' arrival during  $d_{sleep}$ , the ONU transits from sleep to active with the rate  $R_{s2a}$  (line 19), otherwise, from sleep to listen with the rate  $R_{s2l}$  (line 20). Both rates are equal to  $1/2$  (lines 7, 8). Sleep state duration severely affects the trade-off under consideration, since short sleep periods benefit the packets' delay, but they limit the potential energy saving.

Listing 1 shows how the elements of Table II and IV are included in the proposed EPON<sub>CT</sub> model, indicating that the finite state machine of the ONU is synchronized with the OLT-ONU communication under EPONs specifications. Once the transitions' rates have been precisely defined, the goal of the analysis is to properly tune the  $d_{listen}$  and  $d_{sleep}$  parameters in order to adjust the trade-off between QoS requirements, in our case the packets' delay, and the desirable energy saving.

#### IV. RESULTS

We employ the PRISM model checker [46] for the design and analysis of the EPON<sub>CT</sub> Markov model. The results that follow are derived by a dual-core  $2.4 GHz$  machine with  $4 GB$  of RAM. Table V provides information about the state space produced when the PRISM builds the proposed model for different number of *packets*. The rest of network parameters, i.e., packet arrival and service rates  $\lambda$  and  $\mu$ , as well as, the transition rates and duration of listen and sleep period do

TABLE V: EPON<sub>CT</sub> state space results

Transmitted packets [11]	Total states of $S$	Transitions	Iterations	Time (sec)
$10^3$	215	399	24	0.01
$10^4$	8 195	16 299	135	0.2
$10^5$	93 695	187 299	1 035	2.25
$10^6$	948 695	1 897 299	10 035	34.5

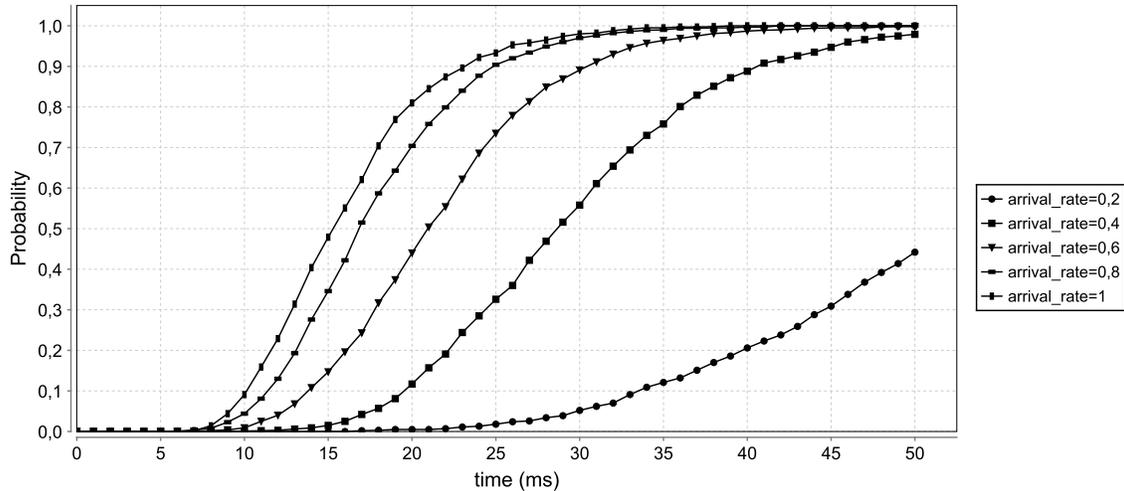


Fig. 5: Probability of OLT transmitting a number of packets (1000) successfully for different packet arrival rates  $\lambda$  ( $10^2$  packets/ms)

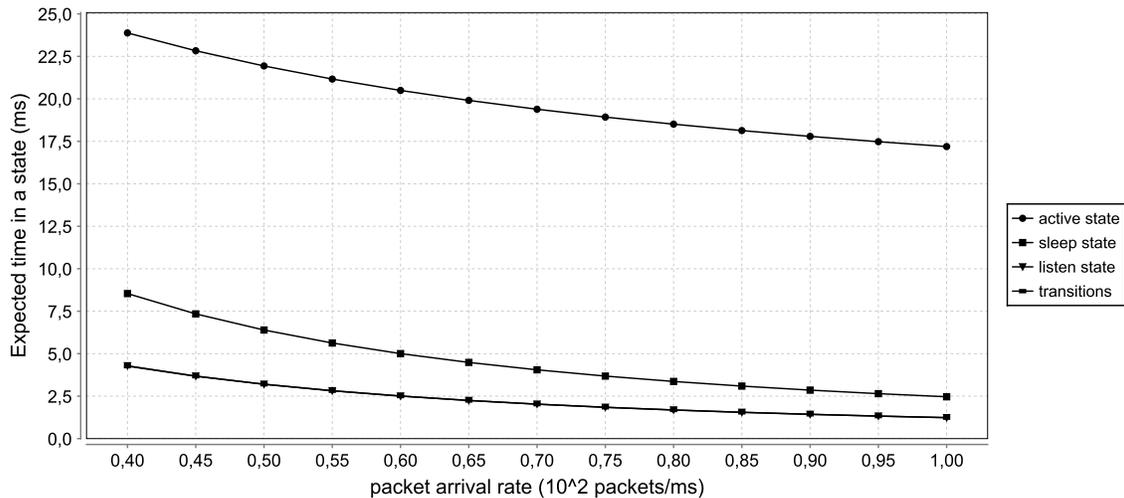


Fig. 6: Time that ONU spends in active, sleep, and listen state, as well as in transitions among them as a function of packet arrival rate  $\lambda$

not affect the model's state space. The columns of Table V concern the total number of the produced states  $S$ , along with the number of transitions between them, and the iterations and time needed to solve the CTMC model. The state space's magnitude denotes the depth of our analysis. In addition, verification's requirements, in terms of hardware resources and time, show a clear advantage over cost- and time-demanding experimentation-based approaches.

Once the model is built, model checking enables the verification of the EPON's properties. We use properties of the form  $\mathcal{P}_{=?}(\phi)$  to evaluate the probability of the path formula  $\phi$ ,

i.e., the probability of certain events. In addition, we use both instantaneous and cumulative reward properties, described in Section II-B, to derive the expected packets' delay and the expected energy saving, respectively. The results that follow provide a realistic representation of EPON's actual performance, since they are derived through exhaustive verification of state space (Table V). This is a benefit over simulation-based techniques which only evaluate a finite number of traces.

The proposed model mastered by the downstream transmission of a number of *packets*. Thus, at a first place, it is desirable to measure the probability that all *packets* will

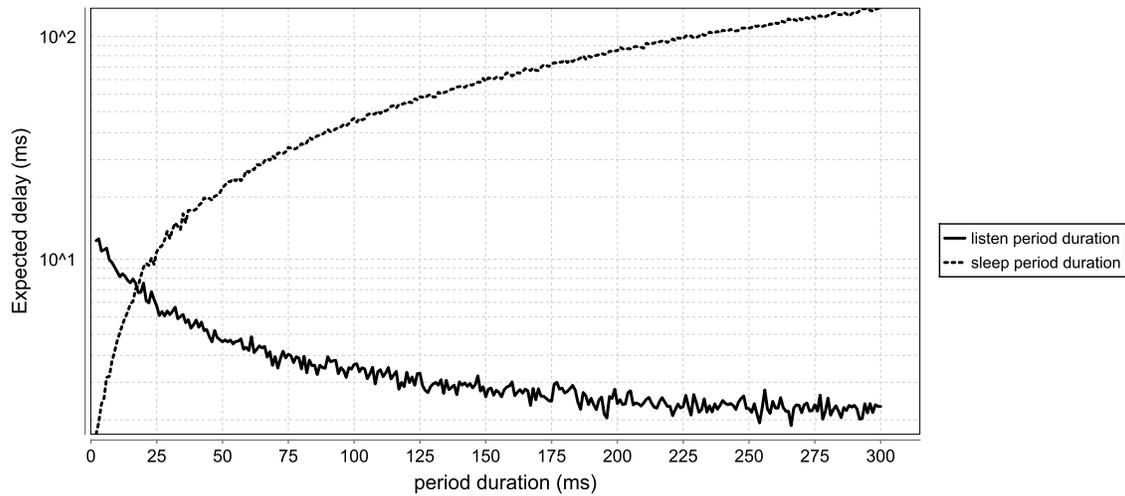


Fig. 7: The impact of listen and sleep period duration (ms) over the expected delay (ms) when packet arrival rate is  $\lambda = 0.3 \times 10^2 \text{ packets/ms}$

be transmitted by the OLT and received by the ONU in a certain amount of time. Apparently, time (ms) depends on the network parameters, i.e., the packets' arrival rate  $\lambda$ , the rate of the downlink channel  $C$ , as well as the packets' length  $l$  ( $\mu = C/l$ ). We define the CSL query:

$$Q_1 : \mathcal{P} =? [F \leq C_0 \text{ finish}], C_0 = 50, \text{ packets} = 1000 \\ \lambda = 0.2 \dots 1, \mu = 1, d_{listen} = 2 \text{ ms}, d_{sleep} = 4 \text{ ms}$$

whose explanation is "which is the probability that 1000 packets will be transmitted and received successfully within 50 ms when packet arrival rate varies from  $0.2 \times 10^2$  to  $1 \times 10^2 \text{ packets/ms}$ ". For the property defined in  $Q_1$ , the verification process will search the full state space produced (i.e., 215 states and 399 transitions according to Table V), in order to find final states before  $C_0$ , for which the formula "finish" will be true. In our EPON<sub>CT</sub> model the "finish" formula represents the Boolean expression which controls that all packets (1000 in query  $Q_1$ ) have been transmitted and received successfully.

Results of  $Q_1$  are depicted in Fig. 5, where curves corresponding to higher arrival rates reach probability 1, while those of  $\lambda = 0.4$  and  $0.2$  depict that verification has been completed by 98% and 44%, respectively, for  $C_0 = 50 \text{ ms}$ . This indicates that formula "finish" will eventually become true, but, as it is expected, the curves shift to the left with the increase of  $\lambda$ , denoting that model concludes sooner.

Ideally, an EPON with  $C = 1.25 \text{ Gbps}$  downstream rate requires 10 ms only to transmit a total of 1000 packets. Our model, which implements both the packets' arrival at OLT and OLT-ONU transmission along with the sleep scheme of ONU, requires, for example, almost 40 ms to reach probability 0.89 when  $\lambda = 0.4 \times 10^2 \text{ packets/ms}$  and  $l = 1518 \text{ bytes}$ . When  $\lambda = 0.4$ , packets arrive at OLT in 25 ms, which entails that roughly 35 ms out of 40 ms are required for packets' arrival and transmission. In addition, due to the sleep scheme, the ONU spends as much as 20.5% and 10.7% of the time in sleep and listen states, correspondingly, and 10.3% in transitions,

which explains the delay of the curve to reach probability 1. Time that ONU spends in each state as well as in transitions among them in line with the packet arrival rate  $\lambda$  is depicted in Fig. 6. The same figure confirms that time in sleep or listen states decreases with the increase of  $\lambda$ , because the ONU remains in active state to receive packets that are destined to itself. The decline in active state time is due to the model's residing in the final state.

#### A. Sleep-aware configuration

In Fig. 5 and 6, we define listen and sleep periods equal to 2 ms and 4 ms, respectively, and, it is reasonable that, this way, we cause extra delays in the model. Since listen and sleep states, introduced at the ONU side for energy saving, cause delays in packets' exchange, they should be configured to tune this trade-off. Next, our model provides quantitative results to demonstrate the impact of  $d_{listen}$  and  $d_{sleep}$  parameters on the cost metrics. In particular, we derive the energy saving percentages putting the ONU into the sleep state, while satisfying certain levels of delay.

For the delay performance, our model is augmented with the reward structure  $X = \text{delay}$  which counts queuing and transmission delay as well as the "wait-to-wakeup" delay [13], [20]. The packets that arrive while the ONU is sleeping have to wait until it wakes up and transits to the active state. The average waiting time equals to  $d_{sleep}/2 \text{ ms}$ , i.e., half the sleep period duration, and is defined as "wait-to-wakeup" delay [13], [20]. For the outcome of Fig. 7 we run the query  $Q_2$  twice:

$$Q_2 : \mathcal{R}\{\text{"delay"}\} =? [I = T], T = 100, \\ \text{ packets} = 10000, \lambda = 0.3, \mu = 1$$

For the first run, we keep  $d_{sleep} = 20 \text{ ms}$  and change  $d_{listen}$  on the interval  $0 \dots 300 \text{ ms}$ , while for the second run, we define  $d_{listen} = 8 \text{ ms}$  and change  $d_{sleep}$  on the aforementioned range. This query corresponds to an instantaneous reward property and expresses the expected delay caused in packets' transmission at  $T = 100 \text{ ms}$  of models' operation. Defining

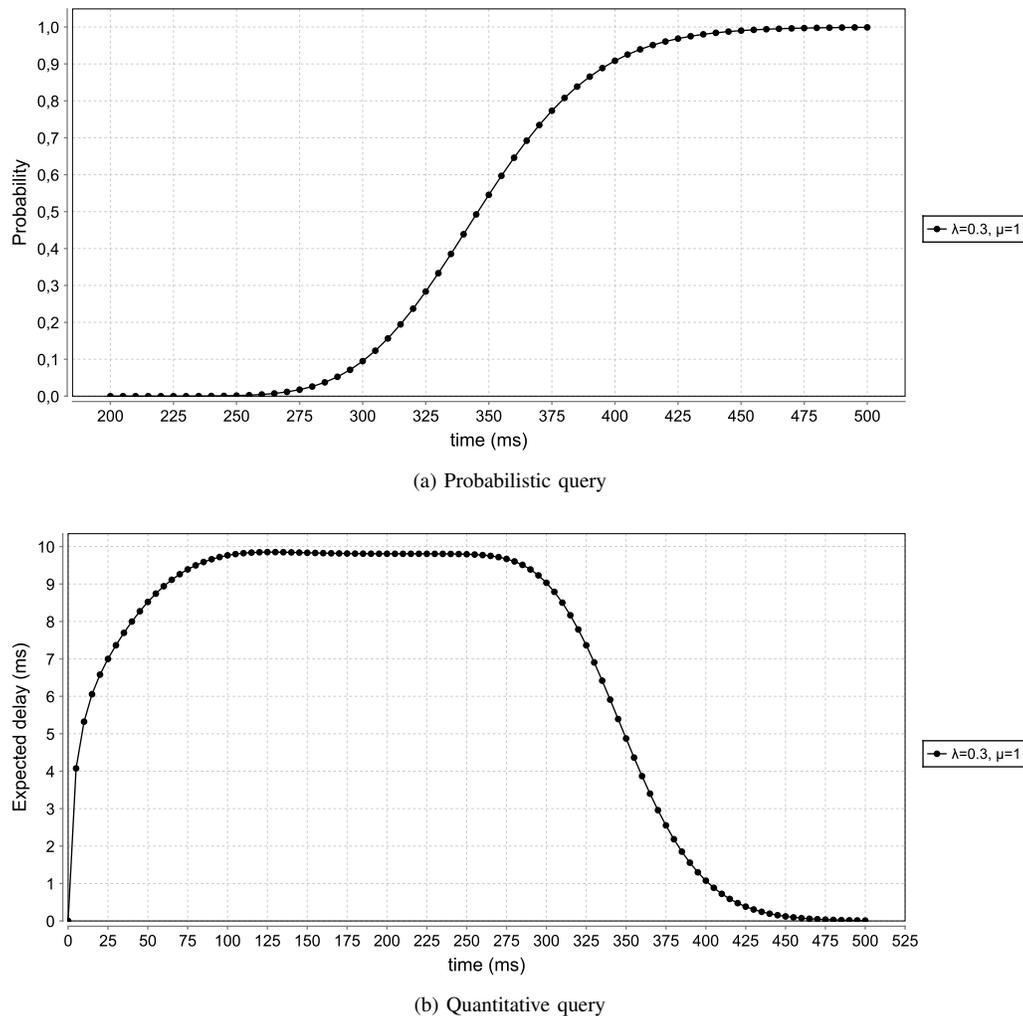


Fig. 8: Defining the time property  $T = 100 \text{ ms}$  of query  $Q_2$

the time property  $T$  in query  $Q_2$  is not an obvious choice. Firstly, it requires the knowledge of the time that the model “finishes”. And then, to be fair, we should detect the queue at its peak.

Thus, for a number of  $packets = 10\,000$ , which arrive with a rate  $\lambda = 0.3 \times 10^2 \text{ packets/ms}$  and are serviced with a rate  $\mu = 1 \times 10^2 \text{ packets/ms}$ , a probabilistic query similar to  $Q_1$  produces Fig. 8a. The curve in Fig. 8a shows that the model concludes in  $450 \text{ ms}$ . This information is used in a quantitative query similar to  $Q_2$  which calculates the expected delay over time. The outcome of Fig. 8b denotes that during an initial period, i.e.,  $0 - 100 \text{ ms}$ , the average delay is increasing due to the increase of the queue size, while after  $300 \text{ ms}$  is decreasing, since the queue is getting empty. The choice of  $T = 100 \text{ ms}$  is apparently fair since it indicates the time of maximum delay.

Back to the main results of query  $Q_2$ , Fig. 7 demonstrates that delay decreases with the increase of the listen period duration ( $d_{sleep}$  is fixed at  $20 \text{ ms}$ ). This is owed to the decrease of the sleep state probability  $P_{sleep}$ . For example,  $P_{sleep} = 1, 0.62$  and  $0.24$  for  $d_{listen} = 5, 55$  and  $200 \text{ ms}$ , respectively, according to Fig. 9. This graph

is produced by the “assistant” query  $Q_{2.1} : \mathcal{P} =? [F \leq C_0 \text{ sleep}]$ ,  $C_0 = 100$ ,  $packets = 10\,000$ ,  $\lambda = 0.3$ ,  $\mu = 1$ ,  $d_{listen} = \{0 \dots 300\} \text{ ms}$ ,  $d_{sleep} = 20 \text{ ms}$ . The “wait-to-wakeup” time causes longer delays for packets arriving during the sleep state compared to those arriving during the active or listen state. Thus, the decrease of  $P_{sleep}$  entails the decrease of the average expected delay. On the contrary, delay increases with the increase of the sleep period duration ( $d_{listen}$  is fixed at  $8 \text{ ms}$ ), since “wait-to-wakeup” time dominates the overall delay. In Fig. 7,  $d_{listen}$  and  $d_{sleep}$  are changed to cause delay of  $10$  up to  $100 \text{ ms}$ , which characterize voice and video traffic classes, correspondingly [23]–[25].

Afterwards, based on Fig. 7, we study the impact of  $d_{listen}$  and  $d_{sleep}$  on the expected energy saving. More specifically, we run the query  $Q_3$  twice:

$$Q_3 : R\{\text{“energy”}\} =? [C \leq C_0], C_0 = 800, \\ packets = 10\,000, \lambda = 0.3, \mu = 1$$

For  $d_{sleep}$  changing, we define  $d_{listen} = 8 \text{ ms}$ . We picked the minimum value which causes delay of  $10 \text{ ms}$  (according to Fig. 7), since, intuitively, higher values of listen period lead to the increase of power consumption. Looking at ONU’s finite

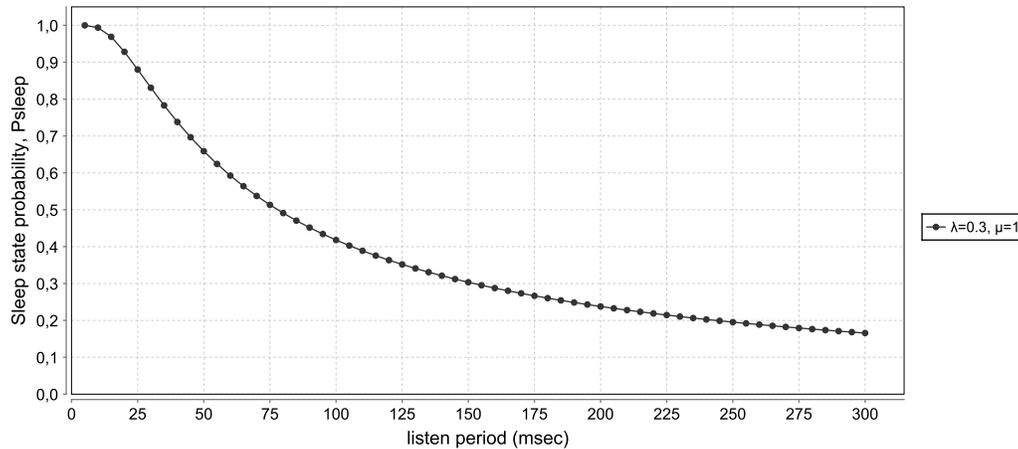


Fig. 9: The sleep state probability  $P_{sleep}$  in line with the listen period duration (ms)

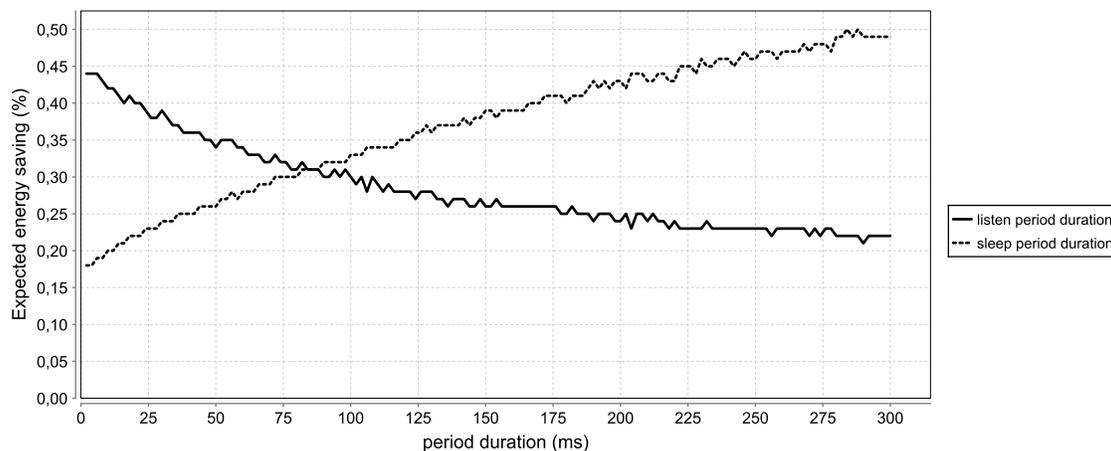


Fig. 10: The impact of listen and sleep period duration (ms) over the expected energy saving when packet arrival rate is  $\lambda = 0.3 \times 10^2 \text{ packets/ms}$

state machine, in Fig. 4, it is expected that increasing the time the ONU stays in listen state and providing the option to interrupt that period going back to the active state in case of packets' arrival, we will have many transitions between active and listen states (major power consumers according to Table III). In addition, as explained above, the increase of  $d_{listen}$  drops the sleep state probability  $P_{sleep}$ , which confirms the intuition that active and listen state probabilities are increasing. For a number of  $packets = 10000$ , which arrive with a rate  $\lambda = 0.3 \times 10^2 \text{ packets/ms}$  and are serviced with a rate  $\mu = 1 \times 10^2 \text{ packets/ms}$ , the query  $Q_3$  corresponds to an cumulative reward property, which calculates the expected energy saving within 800 ms of models' operation, time needed for model to be finished under the aforementioned parameters. Fixing  $d_{listen}$  at 8 ms we observe how the expected energy saving is increasing with the increase of  $d_{sleep}$ . Obviously, the more the ONU getting to sleep the higher the benefit in terms of energy, as depicted in Fig. 10.

However, which is the threshold of  $d_{sleep}$ , so that not only gaining in energy saving, but at the same time taking care of performance constraint, i.e., delay? Fig. 7 provides us with

the maximum value of  $d_{sleep}$  which causes delay of 100 ms. Thus, we run the same query  $Q_3$  for  $d_{listen}$  being changed and  $d_{sleep} = 200$  ms. In this case, we observe how the expected energy saving is decreasing with the increase of  $d_{listen}$ .

### B. Tuning the trade-off

Fig. 7 and 10 clearly depict how the parameters of listen and sleep period durations compete each other over the tradeoff between the energy saving and delay. In our results, we define the queries  $Q_2$  and  $Q_3$  considering the use case of voice and video traffic. However, it is apparent, that our EPON<sub>CT</sub> model can be also exploited in order to provide configurations for different traffic classes.

In particular, the results of Fig. 7 can be exploited as follows. Considering the first class of traffic, i.e., voice, the configuration  $(d_{listen}, d_{sleep}) = (8 \text{ ms}, 20 \text{ ms})$  seems to provide delay equal to 10 ms [23]–[25]. To verify this setup, we run the query  $Q_4$ :

$$Q_4 : \mathbb{R}\{\text{"delay"}\} = ? [I = T], T = 100, \text{ packets} = 10000, \lambda = 0.01 \dots 1, \mu = 1$$

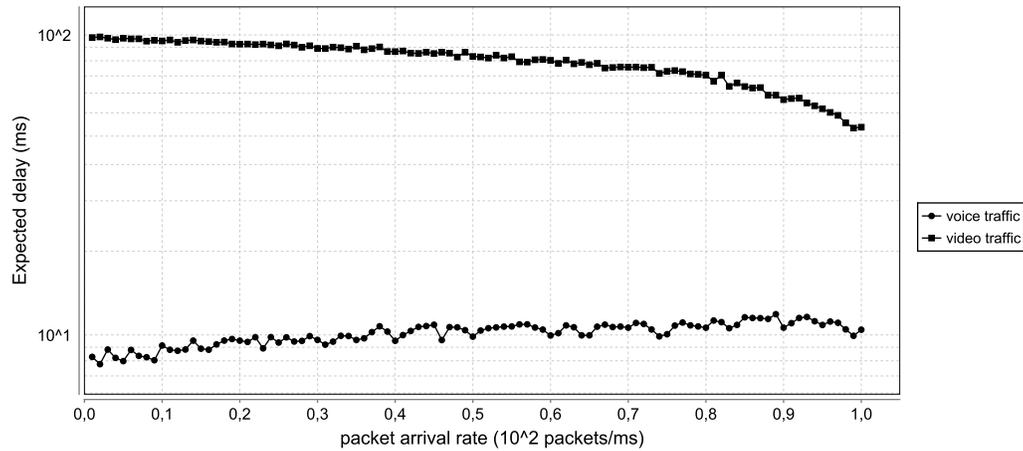


Fig. 11: The expected delay for: i)  $d_{listen} = 8\text{ ms}$  and  $d_{sleep} = 20\text{ ms}$  (voice traffic) and ii)  $d_{listen} = 4\text{ ms}$  and  $d_{sleep} = 200\text{ ms}$  (video traffic), as a function of packet arrival rate  $\lambda$

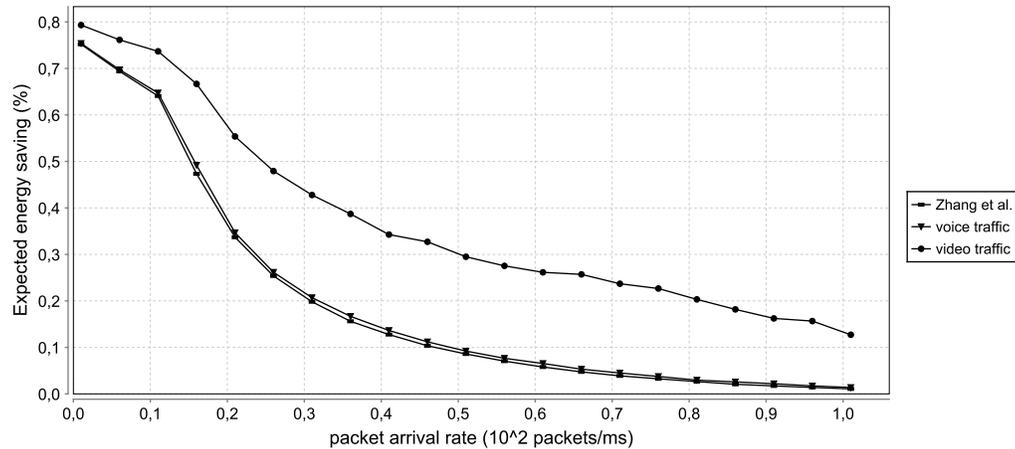


Fig. 12: The expected energy saving for: i)  $d_{listen} = 8\text{ ms}$  and  $d_{sleep} = 20\text{ ms}$  (voice traffic) and ii)  $d_{listen} = 4\text{ ms}$  and  $d_{sleep} = 200\text{ ms}$  (video traffic), as a function of packet arrival rate  $\lambda$

for different packets' arrival rates. In Fig. 11, the curve denoted as voice traffic shows, indeed, that the delay is less than  $10\text{ ms}$  for low arrival rates and slightly increases as  $\lambda$  becomes greater than  $0.5 \times 10^2\text{ packets/ms}$ . Having chosen a short sleep period, the overall average delay is dominated by the queuing delay, which increases with the traffic rate increase.

On the contrary, the curve denoted as video traffic, in Fig. 11, expresses the delay caused when  $(d_{listen}, d_{sleep}) = (4\text{ ms}, 200\text{ ms})$ . In this case, we achieve delay less than  $100\text{ ms}$  [23]–[25], which gets decreasing with the increase of the arrival rate. With a high value of  $d_{sleep}$ , the overall average delay is dominated by the “wait-to-wakeup” delay. However, with the increase of the traffic rate, the sleep state probability decreases, and the “wait-to-wakeup” time also decreases. Therefore, the overall average delay is reduced.

In the aforementioned settings, we have chosen the maximum value of  $d_{sleep}$  (based on Fig. 7) so that we do not exceed the constraint of  $100\text{ ms}$  delay, but at the same time we benefit in terms of energy. According to Fig. 10,  $200\text{ ms}$  of sleep period can contribute as much as 43% in energy saving.

Moreover, we dropped the listen period from  $8\text{ ms}$  to  $4\text{ ms}$ , since based on Fig. 10, this way we can expect the maximum energy saving of 44%. It is worthwhile to mention that, when the delay constraint is strict enough, it is imposed to retrieve the configuration of  $d_{listen}$  and  $d_{sleep}$  exclusively elaborating on their impact over it, i.e., the delay (e.g., Fig. 7). As long as the use-case loosens the time constraint, the energy saving achievements (e.g., Fig. 10) can be prioritized in combination with the delay ones.

Finally, having achieved the delay constraint for the use cases considered (i.e., voice and video traffic), we can derive the results regarding the potential energy saving that we can expect. We run the query  $Q_5$ :

$$Q_5 : R\{\text{“energy”}\} = ? [C \leq C_0], C_0 = 800, \\ \text{packets} = 10000, \lambda = 0.01 \dots 1, \mu = 1$$

for our two settings, corresponding to the use cases of voice and video traffic, and one more for a setting derived by [20]. Since, our EPON<sub>CT</sub> model is inspired by the sleep control

scheme proposed by Zhang *et al.*, it is desirable to show the novelty of our analysis' findings.

The analysis of Zhang *et al.* [13], [20] prioritizes QoS constraint, and, thus, it achieves conservative results in terms of energy saving, with the increase of the packet arrival rate. For example, it starts with 70% energy saving when the arrival rate equal to 6% of the service rate, but shortly after it drops below 20%, as depicted in Fig. 12. An explanation is that the sleep period is quite short, e.g., up to 20 ms, which entails many transitions among the ONU's states. In fact, apart from the impact in energy conservation, delay impact is also expected, but this is not reflected in their results, since the wake-up time is considered negligible in [13], [20]. Our results regarding voice are very similar to that of [20], since in this case, QoS requirements are quite stringent, i.e., 10 ms delay, and thus a short sleep period is preferable.

It is reasonable that all curves in Fig. 12 get decreasing with the increase of the packet arrival rate, since sleep state probability decreases and, thus, the ONU consumes more energy, switching between the active and listen states. However, we observe that, video-traffic results keep the energy conservation at significantly high levels. When the "conservative" policy of the short sleep period, i.e.,  $d_{sleep} = 10$  ms, leads to 20% of energy saving, the "aggressive" policy of the long sleep period, i.e.,  $d_{sleep} = 200$  ms, achieves 43% saving, while it goes down to 20% when the arrival rate equal to 80% of the service rate. This is due to the fact that the QoS constraint, i.e., 100 ms delay, stops being so stringent.

## V. CONCLUSIONS AND LOOKING AHEAD

This paper exploits the power of formal methods to automatically verify an EPON energy-efficiency policy. Verification results reveal that, guided by the QoS constraints and considering the transitions' cost, an "aggressive" policy of long sleep periods, e.g., 100 ms, can be up to 24,9% beneficial for energy-efficiency compared to a "conservative" policy with sleep periods, e.g., 10 ms. Specifying the sleep period duration is a practically useful part of the proposed quantitative analysis, since it could be served as an "inspired" input in a simulation or experimentation analysis. In addition, this outcome is supported by recent studies which bring into light that repeated power-state transitions of ONU result in temperature variations which deteriorate its lifetime.

The proposed analysis is a first step towards an application-aware energy-saving solution for EPONs. Our future plans include a further investigation of the involved performance trade-offs with respect to particular applications and their QoS requirements. For example, real-time video streaming applications may require a more conservative energy-saving strategy associated with improved jitter and delays, while bulk download can be delay-tolerant and operate in bursts, i.e., significantly improving energy efficiency. Furthermore, an extension of our proposed model to support more properties of the real environment, e.g., more ONUs (4 or 16), devices' lifetime parameters or maintenance costs would be valuable. Towards this direction, formal methods' approach allows to increase the model's specification and complexity for

a better realism, which remains a challenge in pure analytical approaches.

## REFERENCES

- [1] K. J. Christensen, C. Gunaratne, B. Nordman, and A. D. George, "The next frontier for communications networks: power management," *Comput. Commun.*, vol. 27, no. 18, pp. 1758–1770, Dec. 2004.
- [2] M. Gupta and S. Singh, "Using low-power modes for energy conservation in Ethernet LANs," in *Proc. IEEE INFOCOM*, May 2007, pp. 2451–2455.
- [3] N. Ansari and J. Zhang, *Media access control and resource allocation: For next generation passive optical networks*. Springer Sci. & Bus. Media, 2013.
- [4] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen, "A survey of energy efficient network protocols for wireless networks," *Wireless Networks*, vol. 7, no. 4, pp. 343–358, Jan. 2001.
- [5] R. S. Tucker, "Green optical communications Part II: Energy limitations in networks," *IEEE J. Sel. Topics Quantum Electron.*, vol. 17, no. 2, pp. 261–274, Mar.-Apr. 2011.
- [6] C. Lange, M. Braune, and N. Gieschen, "On the energy consumption of FTTB and FTTH access networks," in *Nat. Fiber Optic Engineers Conf.* Optical Society of America, Feb. 2008.
- [7] J. Zhang and N. Ansari, "An application-oriented fair resource allocation scheme for EPON," *IEEE Syst. J.*, vol. 4, no. 4, pp. 424–431, Dec. 2010.
- [8] Y. Zhang, P. Chowdhury, M. Tornatore, and B. Mukherjee, "Energy efficiency in telecom optical networks," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 4, pp. 441–458, Jul. 2010.
- [9] P. G. Sarigiannidis, S. G. Petridou, G. I. Papadimitriou, and M. S. Obaidat, "IGFS: A new MAC protocol exploiting heterogeneous propagation delays in the dynamic bandwidth allocation on WDM-EPON," *IEEE Syst. J.*, vol. 4, no. 1, pp. 49–56, Mar. 2010.
- [10] L. Valcarenghi, D. P. Van, P. G. Raponi, P. Castoldi, D. R. Campelo, S.-W. Wong, S.-H. Yen, L. G. Kazovsky, and S. Yamashita, "Energy efficiency in passive optical networks: where, when, and how?" *IEEE Netw.*, vol. 26, no. 6, pp. 61–68, Nov.-Dec. 2012.
- [11] N. M. Bojan, D. P. Van, L. Valcarenghi, and P. Castoldi, "An energy efficient ONU implementation," in *Sustainable Internet and ICT for Sustainability (SustainIT)*. IEEE, Oct. 2012, pp. 1–4.
- [12] S.-W. Wong, L. Valcarenghi, S.-H. Yen, D. R. Campelo, S. Yamashita, and L. Kazovsky, "Sleep mode for energy saving PONs: Advantages and drawbacks," in *IEEE Globecom Workshops*, Nov. 2009, pp. 1–6.
- [13] J. Zhang, M. T. Hosseinabadi, and N. Ansari, "Standards-compliant EPON sleep control for energy efficiency: Design and analysis," *IEEE J. Opt. Commun. Netw.*, vol. 5, no. 7, pp. 677–685, Jul. 2013.
- [14] J. Zhang and N. Ansari, "Toward energy-efficient 1G-EPON and 10G-EPON with sleep-aware MAC control and scheduling," *IEEE Commun. Mag.*, vol. 49, no. 2, pp. s33–s38, Feb. 2011.
- [15] M. Fiammengio, "Sleep mode scheduling technique for energy saving in TDM-PONs," M.S. thesis, KTH, School of Information and Communication Technology, Stockholm, Sweden, 2011.
- [16] D. P. Van, L. Valcarenghi, M. Chincoli, and P. Castoldi, "Experimental evaluation of a sleep-aware dynamic bandwidth allocation in a multi-ONU 10G-EPON testbed," *Optical Switching and Networking*, vol. 14, no. 1, pp. 11–24, Aug. 2014.
- [17] *IEEE Standard for Service Interoperability in Ethernet Passive Optical Networks (SIEPON)*, IEEE Std. 1904.1, 2017.
- [18] J. Mandin, "EPON powersaving via sleep mode," presented at the IEEE P802.3av 10G-EPON Task Force Meeting, vol. 3, Sep. 2008, pp. 15–33. [Online]. Available: [http://www.ieee802.org/3/av/public/2008\\_09/3av\\_0809\\_mandin\\_4.pdf](http://www.ieee802.org/3/av/public/2008_09/3av_0809_mandin_4.pdf)
- [19] R. Kubo, J.-I. Kani, Y. Fujimoto, N. Yoshimoto, and K. Kumozaki, "Sleep and adaptive link rate control for power saving in 10G-EPON systems," in *IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov.-Dec. 2009, pp. 1–6.
- [20] J. Zhang and N. Ansari, "Standards-compliant EPON sleep control for energy efficiency: Design and analysis," in *IEEE Int. Conf. on Commun. (ICC)*, Jun. 2012, pp. 2994–2998.
- [21] S. S. Lee and A. Chen, "Design and analysis of a novel energy efficient ethernet passive optical network," in *9th Int. Conf. on Networks (ICN)*. IEEE, Apr. 2010, pp. 6–9.
- [22] Y. Yan and L. Dittmann, "Energy efficiency in ethernet passive optical networks (EPONs): Protocol design and performance evaluation," *J. Commun.*, vol. 6, no. 3, pp. 249–261, May 2011.

- [23] G. Kramer, B. Mukherjee, S. Dixit, Y. Ye, and R. Hirth, "Supporting differentiated classes of service in Ethernet passive optical networks," *J. of Optical Networking*, vol. 1, no. 8, pp. 280–298, 2002.
- [24] R. K. Mok, E. W. Chan, and R. K. Chang, "Measuring the quality of experience of HTTP video streaming," in *Int. Symp. on Integrated Network Manage. (IM)*. IFIP/IEEE, May 2011, pp. 485–492.
- [25] P. Calyam and C.-G. Lee, "Characterizing voice and video traffic behavior over the Internet," in *Int. Symp. on Comput. and Inform. Sciences (ISCIS)*, Oct. 2005.
- [26] L. Chiaraviglio, P. Wiatr, P. Monti, J. Chen, J. Lorincz, F. Idzikowski, M. Listanti, and L. Wosinska, "Is green networking beneficial in terms of device lifetime?" *IEEE Commun. Mag.*, vol. 53, no. 5, pp. 232–240, May 2015.
- [27] J. Li, Z. Zhong, N. Hua, X. Zheng, and B. Zhou, "Balancing energy efficiency and device lifetime in TWDM-PON under traffic fluctuations," *IEEE Commun. Lett.*, vol. 21, no. 9, pp. 1981–1984, May 2017.
- [28] T. Deshpande, P. Katsaros, S. B. Basagiannis, and S. A. Smolka, "Formal analysis of the DNS bandwidth amplification attack and its countermeasures using probabilistic model checking," in *13th Int. Symp. on High-Assurance Systems Engineering, (HASE)*. IEEE, Nov. 2011, pp. 360–367.
- [29] D. Chitimalla, S. Thota, S. S. Savas, P. Chowdhury, M. Tornatore, S.-S. Lee, H.-H. Lee, S. Park, H. S. Chung, and B. Mukherjee, "Application-aware software-defined EPON access network," *Photonic Network Commun.*, vol. 30, no. 3, pp. 324–336, Dec. 2015.
- [30] C. Li, W. Guo, W. Wang, W. Hu, and M. Xia, "Programmable bandwidth management in software-defined EPON architecture," *Optics Commun.*, vol. 370, pp. 43–48, Jul. 2016.
- [31] A. S. Thyagaturu, A. Mercian, M. P. McGarry, M. Reisslein, and W. Kellerer, "Software defined optical networks (SDONs): A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2738–2786, Jul. 2016.
- [32] M. Kwiatkowska, G. Norman, and D. Parker, "Stochastic model checking," in *Int. School on Formal Methods for the Design of Comput., Communication and Software Syst. (SFM)*. Springer, May-Jun. 2007, pp. 220–270.
- [33] S. Petridou, S. Basagiannis, and M. Roumeliotis, "Survivability analysis using probabilistic model checking: A study on wireless sensor networks," *IEEE Syst. J.*, vol. 7, no. 1, pp. 4–12, Mar. 2013.
- [34] N. Alexiou, S. Basagiannis, and S. Petridou, "Formal security analysis of near field communication using model checking," *Comput. & Security*, vol. 60, pp. 1–14, Jul. 2016.
- [35] S. Basagiannis, S. Petridou, N. Alexiou, G. Papadimitriou, and P. Katsaros, "Quantitative analysis of a certified e-mail protocol in mobile environments: A probabilistic model checking approach," *Comput. & Security*, vol. 30, no. 4, pp. 257–272, Jun. 2011.
- [36] I. Paparrizos, S. Basagiannis, and S. Petridou, "Quantitative analysis for authentication of low-cost RFID tags," in *IEEE 36th Conf. on Local Comput. Networks (LCN)*, Oct. 2011, pp. 295–298.
- [37] S. Petridou and S. Basagiannis, "Towards energy consumption evaluation of the SSL handshake protocol in mobile communications," in *Proc. 9th Annu. Conf. on Wireless On-demand Network Syst. and Services (WONS)*. IEEE, Jan. 2012, pp. 135–138.
- [38] A. Aziz, K. Sanwal, V. Singhal, and R. K. Brayton, "Model-checking continuous-time Markov chains," *ACM Trans. Comput. Logic*, vol. 1, no. 1, pp. 162–170, Jul. 2000.
- [39] S. Petridou, S. Basagiannis, and L. Mamas, "Energy-efficiency analysis under QoS constraints using formal methods: A study on EPONs," in *IEEE Int. Conf. on Commun. (ICC)*, May 2017, pp. 1–6.
- [40] M. P. I. Dias and E. Wong, "Sleep/doze controlled dynamic bandwidth allocation algorithms for energy-efficient passive optical networks," *Optics express*, vol. 21, no. 8, pp. 9931–9946, 2013.
- [41] A. Bianco and L. de Alfaro, "Model checking of probabilistic and nondeterministic systems," in *Proc. 15th Conf. on Foundations of Comput. Technology and Theoretical Comput. Sci.* Springer, Dec. 1995, pp. 499–513.
- [42] C. Baier, J.-P. Katoen, and K. G. Larsen, *Principles of model checking*. MIT press, 2008.
- [43] Q. Qiu, Q. Qu, and M. Pedram, "Stochastic modeling of a power-managed system-construction and optimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 10, pp. 1200–1217, Oct. 2001.
- [44] M. Kwiatkowska, "Model checking for probability and time: from theory to practice," in *Proc. 18th Annu. Symp. on Logic in Comput. Sci.* IEEE, Jun. 2003, pp. 351–360.
- [45] M. Kwiatkowska, G. Norman, and A. Pacheco, "Model checking expected time and expected reward formulae with random time bounds," *Comput. & Math. with Applicat.*, vol. 51, no. 2, pp. 305–316, Jan. 2006.
- [46] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in *Int. Conf. on Comput. Aided Verification*. Springer, Jul. 2011, pp. 585–591.



an Associate Editor of the International Journal of Communication Systems.

**Sophia Petridou (M'08)** is a Lecturer at the Department of Applied Informatics, University of Macedonia, Greece. She received the Diploma and Ph.D. degrees in Computer Science from the Department of Informatics, Aristotle University of Thessaloniki, Greece in 2000 and 2008 respectively. Her main research interest are in the areas of MAC protocols of optical networks, wireless networks, probabilistic model checking of protocols and clustering algorithms. She has published around 30 papers in international journals and conferences. She serves as



journals and conference proceedings in the areas of formal methods, security and safety analysis.

**Stylianos Basagiannis (M'12)** is a principal research scientist working at United Technologies Research Centre in Cork, Ireland (UTRC) from 2011. He holds a Ph.D. (2009), an MSc (2005) and a BSc (2004) in Computer Science with specialization in formal verification and cyber security. He has participated in more than 14 R&D projects related to Cyber-Physical Systems' analysis, cyber-security and embedded systems. He is a patent holder of 3 invention disclosures (9 pending) and he has published more than 30 research articles in international



Energy Efficient Communication. He participated in several international research projects, such as NECOS (H2020), Dolfin (FP7), Autonomic Internet (FP7), UniverSELF (FP7) and others. He published around 50 papers in international journals and conferences. He served as a General co-chair for the IFIP WWIC 2016 conference, General co-chair for the INFOCOM SWFAN 2016 workshop, TPC co-chair for the INFOCOM SWFAN 2017, IFIP WWIC 2012 and IEEE E-DTN 2009 conferences / workshops and as a guest editor for the Elsevier Ad Hoc Networks Journal.

**Lefteris Mamas** is an Assistant Professor at the Department of Applied Informatics, University of Macedonia, Greece. Before that, he was a researcher at the University College London, Space Networking Center / Democritus University of Thrace and DoCoMo Eurolabs in Munich. He received his Ph.D. from the Department of Electrical and Computer Engineering, Democritus University of Thrace in Greece. His research interests lie in the areas of Software-Defined Networks, Network Function Virtualization, Mobile Edge / Fog Computing and