UNIVERSITA’ DEGLI STUDI DI PISA

FACOLTA’ DI INGEGNERIA

Corso di Laurea Specialistica in Ingegneria delle Telecomunicazioni

Tesi di Laurea Specialistica

"Methods for improving energy efficiency in TDM PONs"

RELATORI:

Prof. Stefano Giordano

Ing. Rosario Giuseppe Garroppo

Ing. Luca Valcarenghi

Ing. Paolo Monti

CANDIDATO:

Michele Chincoli

Anno Accademico 2010/2011

CONTENTS

Abstract

Even though Information and Communications Technologies (ICT) are currently consuming between 2% and 4% of the electricity consumed worldwide, the number of efforts devoted to reduce the communications network energy consumption is increasing. This is mainly due to the foreseen growth of ICT even in substitution of personal travel. Access networks are the network segment that currently consumes the highest percentage of energy. Even though the utilization of optical technologies can potentially reduce the energy consumed by current ADSL modems, the further reduction of the energy consumed by passive optical access networks (PON) is attracting a lot of interests.

Previous studies showed that, in PONs, the majority of the energy in consumed by the customer premises equipments, i.e. the Optical Network Units (ONUs), because of the many idle periods used only for synchronization. For this reason the target of our work is to save energy by exploiting cyclic sleep periods in the ONU. In particular the Sleep and Periodic Wake-up (SPW) technique is considered. The SPW mechanism is managed by the OLT and the choice of the sleep period for the ONUs can be based on different parameters. In this work two approaches are considered for deciding the sleep period: interarrival-based and service-based.

The interarrival-based approach has been previously presented. In this thesis a simulator based on Opnet Modeler is built to verify the validity of the previously presented results. Then a novel service-based sleep time scheme is designed and evaluated. The novelty of our work resides in presenting a service-based saving energy technique with variable sleep period to maximize the energy efficiency guaranteeing the maximum tolerable delay of the applications subscribed by the ONU.

The main difference between the two approaches is how the sleep period is set. Following SPW technique, the OLT sets the sleep period according to traffic conditions such as average frame interval and queue length in the interarrival-based algorithm, and class of service (CoS) in the service-based algorithm. In the interarrival-based the sleep period is fixed, instead in the service-based the sleep period changes in function of the delay constraints of subscribed services to guarantee the service performance.

The simulation results in the interarrival-based approach are very similar to the published ones. In case of low and high bandwidths, the values of average power are matched, instead the values of average queuing delay differ because of reasonable different assumptions. The increasing trend are the same in both results.

The service-based approach resulted in the average frame delay, which exploits the maximum tolerable delay maximizing the energy efficiency. The SPW technique with service-based approach was presented in the Optical Fiber Communication Conference and Exposition (OFC) 2012 in Los Angeles.

Riassunto analitico

Sebbene le Tecnologie di Informazione e Comunicazione (ICT) consumino ad oggi tra il 2% e il 4% del consumo di elettricità mondiale, il numero di sforzi mirati alla riduzione del consumo energetico delle reti di comunicazione è in aumento. Questo è maggiormente dovuto alla prevista crescita di ICT anche in sostituzione agli spostamenti fisici. Le reti di accesso sono la porzione di rete che attualmente consuma la più alta percentuale di energia. Anche se l'uso di tecnologie ottiche possono potenzialmente ridurre l'energia consumata dai correnti modem ADSL, la conseguente energia consumata dalle reti di accesso passive (PON) attrae molto interesse.

Studi passati mostrano che, nelle PON, la maggior energia è consumata dalle apparecchiature di utenza, per esempio, le unità di rete ottica (ONU), a causa dei molti periodi di inattività usati solo per la sincronizzazione. Per questo motivo, l'obiettivo del nostro lavoro è il risparmio energetico sfruttando periodi ciclici di sleep nelle ONU. In particolare la tecnica Sleep and Periodic Wake-up è presa in considerazione. Il meccanismo SPW è gestito dall'OLT e la scelta del periodo di sleep per le ONU si può basare su diversi parametri. In questo lavoro due approcci sono considerati per decidere il periodo di sleep: interarrival-based e service-based.

L'approccio interarrival-based è stato presentato in precedenza. In questa tesi un simulatore basato su Opnet Modeler è implementato per verificare la validità dei risultati precedentemente presentati. Successivamente un nuovo schema service-based con periodi di sleep è stato progettato e valutato.

L'originalità del nostro lavoro consiste nella presentazione di una tecnica per risparmio energetico service-based con periodi di sleep variabile per massimizzare l'efficienza energetica garantendo il massimo ritardo tollerabile delle applicazioni a cui l'ONU è abbonato.

La principale differenza tra i due approcci riguarda come il periodo di sleep è impostato. Seguendo la tecnica SPW, l'OLT imposta il periodo di sleep in base alle condizioni di traffico come il tempo d'interarrivo medio e la lunghezza della coda nell'approccio interarrival-based, e come la classe di servizio (CoS) nell'approccio service-based. Riguardo l'interarrival-based il periodo di sleep è fisso, invece nel service-based il periodo di sleep cambia in funzione del limite di ritardo imposto delle applicazioni per garantire le prestazioni di servizio.

I risultati delle simulazioni nell'approccio interarrival-based sono molto simili a quelle pubblicate. Nel caso di basse e alte bande, i valori di potenza media combaciano, mentre i valori di ritardo di accodamento medio differiscono a causa di diverse assunzioni. L'andamento delle curve è lo stesso.

L'approccio service-based con risultati riguardo il ritardo medio dei pacchetti, sfrutta il massimo ritardo tollerabile per massimizzare l'efficienza energetica.

La tecnica SPW con approccio service-based è stato presentato all'Optical Fiber Communication Conference and Exposition (OFC) 2012 a Los Angeles.

1 – INTRODUCTION

1.1 - Overview

The thesis work proposes the development of new sleeping techniques for energy efficiency in Time Division Multiplexing (TDM) Passive Optical Networks (PONs).
The work was made at the NEGONET, group at the Royal Institute of Technology (KTH) in Kista (Stockholm) and at the laboratories of the Institute of the Communication, Information and Perception Technologies (TeCIP) of the Scuola Superiore Sant'Anna.
Saving energy is an important field in which many researchers are working to find new and improved techniques. It's not only an academia discipline of study but it's also considered by telecom providers with the aim to reduce their operative expanse (OPEX).
As reported in previous research, it becomes obvious that home networks consume the largest share of energy in Telecommunication networks, but concerning the network sections, highest energy consumption shares are observed in the fixed and mobile access networks because of the huge number of distributed network elements in the field. This is mainly due to the high number of involved elements (i.e., the customer premises equipments — CPEs).
In this work, energy efficient techniques are developed for the Ethernet PON architecture with Time Division Multiplexing (TDM).
The tool of simulation used to assess the proposed technique performance is OPNET Modeler.

1.2 - PASSIVE OPTICAL NETWORK

The general structure of a modern telecommunication network consists of three main portions: backbone (or core) network, metro/regional network, and access network (Fig. 1).

The PONs reside in the family of the access networks, called also first-mile networks (or last-mile from a point of view of the core network). The access network is the portion of the whole network developed between the Central Office (CO) to the customer premises.

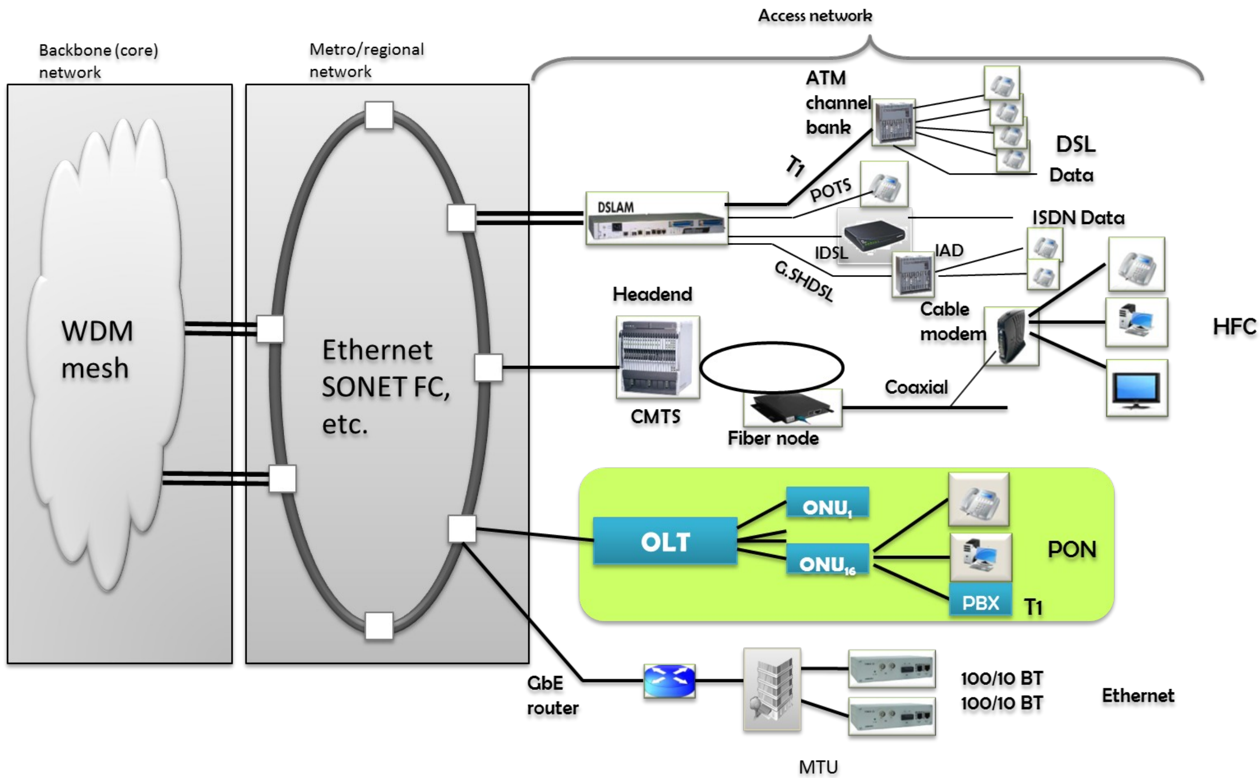


Fig.1 – Structure of a Modern Telecommunication Network

The optical fiber has the advantage of high bandwidth, low loss, and low noise, thus it is commonly used in the core and backbone network.

Because of the continue bandwidth requirements, the rise of customers and the diffused use of services (e.g., video-on-demand, gaming online and peer-to-peer data transferring), the providers are expanding the usage of optical fiber to the access networks. Mostly the PONs are referred to the Fiber-To-The-x (FTTx) system, whose “x” is the variable name (e.g., home, cabin, building,...) depending on how the fiber is close to the users.

After more than 20 years of active research, according to [1], PONs are having important deployments in Asia, North America and, less, in Europe.

A PON is a power splitting Point to Multi-Point (PS-P2MP) network with a remote passive splitter, called Remote Node (RN), to share broadcast downstream traffic to the recipients. Basically, the architecture is composed of a shared optical fiber with a unit of management to the CO, the Optical Line Terminal (OLT), and Customer Premises Equipments (CPEs) to the home network, the Optical Network Units (ONUs).

Each OLT is shared by up to 32 ONUs for a maximum separation of 20 km between the OLT and ONU. In the commercial PONs a split ratio of 1:16 or 1:32 is commonly used. The PON architecture can be designed in different structures.

About downstream direction, from the OLT to the ONUs, the Time Division Multiplexing (TDM) PON (Fig. 2) and the Wavelength Division Multiplexing (WDM) PON (Fig. 3) are presented. The TDM-PON function is to transmit in separated time slots the traffic to each ONU and the RN used is a passive splitter that broadcast all the packets to all the ONUs. On the other hand, the WDM-PON sends in private channels the traffic to each ONU by different wavelength. The RN, in this case, is a WDM coupler.

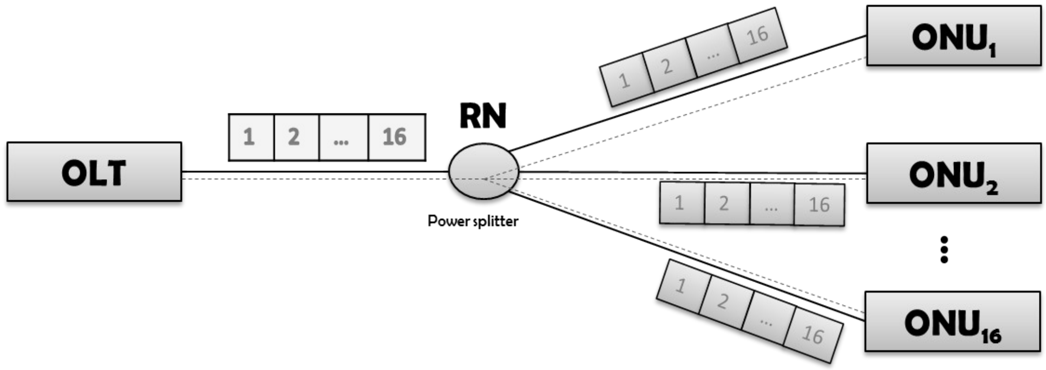


Fig. 2 – TDM-PON architecture

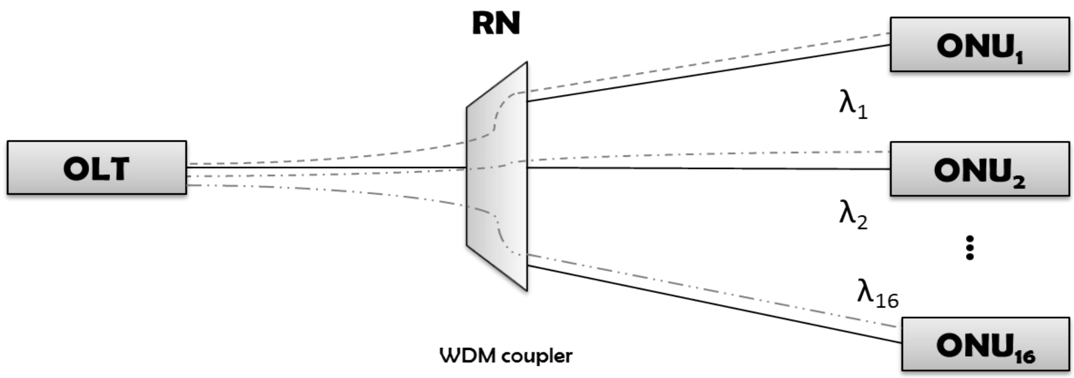


Fig. 3 – WDM-PON architecture

The OLT and the ONUs can communicate by two optical fibers or one optical fiber. If two optical fibers reside between the units then the upstream and downstream traffic flow in distinct physical channels and two splitters, for TDM architecture, or two multiplexer couplers, for WDM, are used. Besides, the upstream and downstream traffic are in the same optical fiber and so some solutions are applied. One solution is to separate the different optical signals by simple 3-dB 1:2 directional couplers at the OLT and ONU, using only one wavelength, but the near-end cross talk (NEXT) disturb can occur. In a similar way, the Time Division Duplex (TDD) can be realized, in which the OLT and ONU take turns to use the fiber in a ping-pong fashion for upstream and downstream transmissions. In this situation the NEXT effect is avoided. The last solution is to use the Wavelength Division Duplex (WDD), where upstream and downstream traffic are separated by two different wavelengths. An example of commercial TDM PON with WDD is in the Fig. 4.

In this architecture, an OLT is connected to the ONUs via a 1:32 splitter. The maximum transmission distance covered is usually 10–20 km. The upstream and downstream traffic are separated by 1.3 μm and 1.49 μm wavelengths respectively. Different OLTs are situated in the CO connected to a switch or cross-connect to the backbone network. The OLTs are constructed by line cards in a chassis in whom also the switch can be hosted.

The portion between one OLT and ONUs is called PON section. The signals transported in this section can be encoded and multiplexed in different formats and schemes depending on the PON standard implemented. Nevertheless, beyond the PON section, standard format signals are used for client interface hand-off, switching, and cross-connect. The most common standard interface used today is the Ethernet interface.

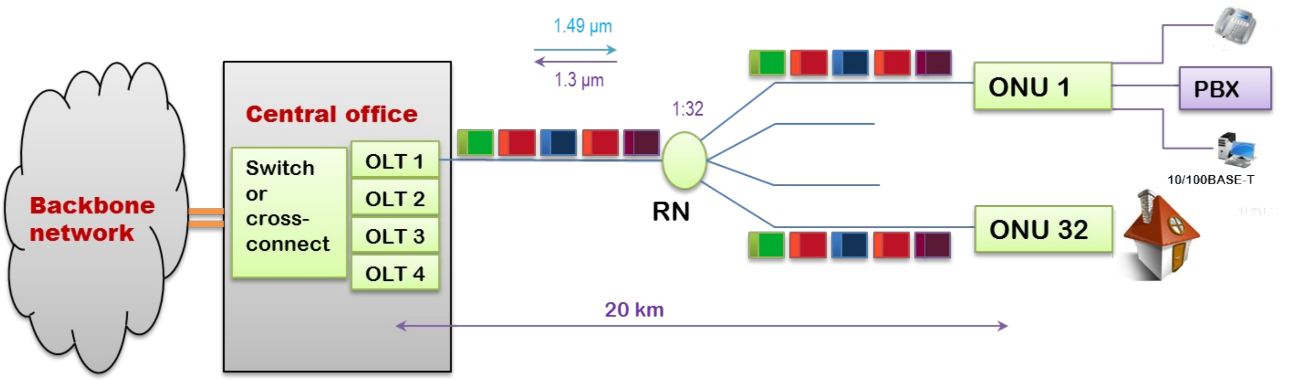


Fig. 4 – Example of Commercial TDM-PON architecture

In a TDM-PON architecture, the traffic, from or to one ONU, is frame interleaved and it is identified by a Logical Link ID (LLID) carried in the preamble. The downstream transmissions are broadcast point-to-multipoint (P2MP), instead the upstream transmissions are multipoint-to-point sharing the same physical medium.

The Time Division Multiple Access (TDMA) is included in a PON architecture to

avoid collisions between frames of each ONU, whose transmit traffic in a shared optical fiber to the OLT. The mechanisms is that each ONU has a specific time slot to transmit own traffic and thus it allows to the OLT to have only one receiver.
PON standardizations are represented into two big groups: Ethernet PON (EPON) standardized in 802.3ah by IEEE and Gigabit-Capable PON (G-PON) standardized in the G.984.x and G.987.x ITU-T Recommendations .
The EPON has the main characteristic to encapsulate IP packets in Ethernet frame. On the other hand, it is used a new framing mechanism called G-PON encapsulation mode (GEM), which is based on the original idea of generic framing procedure (GFP).
The EPON architecture is considered in this work and detailed in the next section.

1.2.1 - EPON

EPON is a PON-based network that carries data traffic encapsulated in Ethernet frames as defined in the IEEE 802.3 standard.
Ethernet covers the physical layer and data link layer of the open system interconnect (OSI) reference model. Fig. 5 shows a comparison of the layering model of the traditional point-to-point (P2P) Ethernet and the point-to-multipoint (P2MP) EPON architecture.
The most important difference is the exchanging sublayer of Media Access Control (MAC) to Multipoint Media access Control (MPMC). The MPMC has the task to manage the Multipoint Control Protocol (MPCP) to coordinate the access to the shared PON medium among EPON ONUs .
The stacks of the ONU and OLT differ to the MPCP entity, the first one is a slave and the second one is a master. The MPCP manages by MAC control messages the assignment of the transmission window to each ONU so the upstream transmission can flow without collisions. There are two modes of operation of MPCP: autodiscovery (initialization) and normal operation. In the autodiscovery

mode, the OLT learns the MAC address and the RTT of new joined ONUs if they are detected. Instead, in the normal mode, the OLT assigns the transmission window to the discovered ONUs.

In a PON, the communications from each ONU to a destination pass through a splitter, a passive component that forwards all the frames only to the OLT. For this motivation, the transmissions of each ONU can't be detected from other ONUs.

To resolve this issue and to ensure seamless integration with other Ethernet networks, devices attached to the EPON medium will use an extended reconciliation sublayer (RS), which will emulate the point-to-point (P2P) medium.

The P2P emulation function is achieved by modifying the preamble in front of the MAC frame to include a LLID that identifies each ONU. The modified preamble with the LLID is used in the PON section.

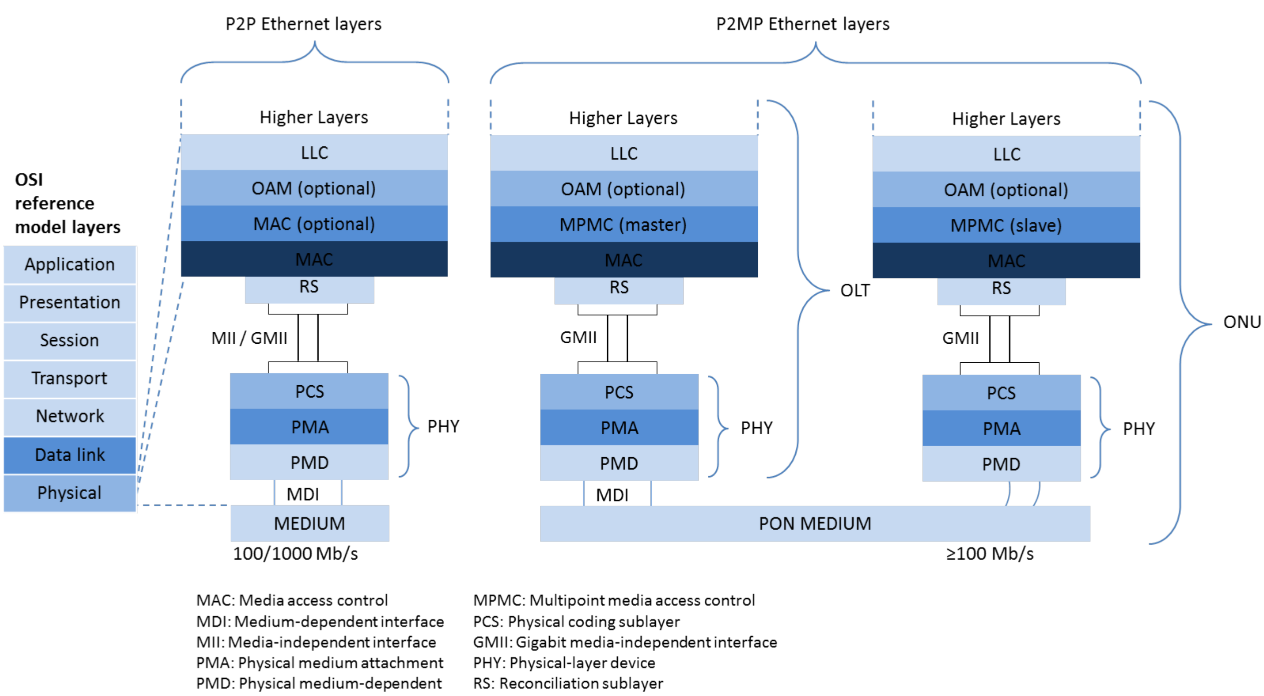


Fig.5 – P2P Ethernet architecture and P2MP EPON architecture

The PON is a centralized system and the OLT has the role of master. This architecture allows to have simple and chip ONUs that doesn't know the state of the network and a simple topology P2P from the ONUs to the OLT. These advantages weren't possible in a distributed system because, in the upstream case, an ONU has to monitor the network by communications with the other ONU to know their state of transmission and so send the traffic alternatively without collisions.

In a centralized PON system is not applicable the Carrier Sense Multiple Access/Collision Detection (CSMA/CD) because of the long distance between the OLT and the ONUs (up to 20 km). In the situation of a collision, the ONU could be informed of it after an high delay since the detection to the OLT by sending a jam frame. Thus, the mechanism used not to have collisions is established in MPCP and it consists of a polling scheme exchanging grant and request messages. The MPCP facilitates the implementation of an algorithm to take decisions about the allocation of bandwidth to give for each ONU. The algorithm is known as Dynamic Bandwidth Allocation (DBA); it isn't specific and it's out of scope for the standard. The choice of the DBA is left to equipment vendors.

In the mechanism the OLT would allow only one ONU to transmit at any given time. It manages the assignation of a transmission timeslot setting the startTime and the length values of the slot. The values for startTime and length are decided upon by a DBA agent or scheduler, located in MAC control client, a sublayer outside the scope of IEEE 802.3ah. Afterward the values are passed to the gate process in which a GATE message is built and sent to the ONUs.

The ONU parses and demultiplexes the GATE message which is passed to the gate process that manages the upstream transmission according to the timeslot allocated. In an EPON system, the downstream physical link maintains continuous signal stream and clock synchronization. In the upstream direction, in order to maintain a common timing reference with the OLT, ONUs use loop timing for the upstream burst mode transmission, i.e. the clock for upstream signal transmission is derived from the downstream received signal.

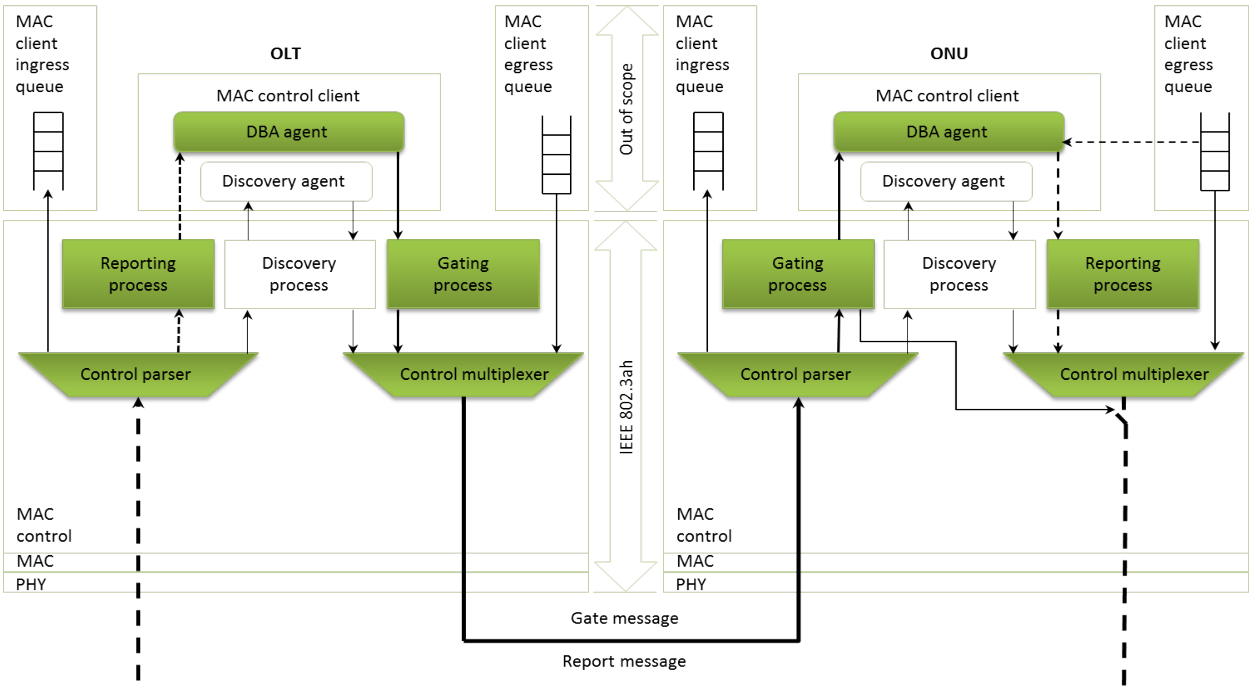


Fig.6 – MPCP scheme

Finally the ONU informs the OLT its local conditions for the next timeslot allocation. It sends a REPORT message containing the amount of traffic to be transmitted in the next polling cycle. REPORT frames can be sent only in previously assigned timeslots. At the OLT, the received REPORT frame is parsed and demultiplexed to the OLT's reporting process, which, in turn, passes it to the DBA agent [2]. The complete scheme of MPCP is shown in the Fig. 6.

1.3 - Energy consumption

It has been widely recognized that reducing power consumption in data communication networks is becoming an important goal not only for reducing CO2 emissions but also to reduce Capital Expenditures (CAPEX).
Previous research have already studied the impact of Telecommunication networks on the global energy consumption. The [3] estimates that the Internet currently consumes about 0.4% of electricity consumption in broadband-enabled countries. This percentage is in growing up following the increasing trend of access rates and it could rise to 1% considering improved saving energy techniques.
Moreover in [3], it is shown that between different access network technologies (Passive Optical Network - PON, Asymmetric digital subscriber line - ADSL, Fiber to the Node - FTTN and Point-to-Point - PtP), the most energy efficient between the access networks is PON for low access rates and PtP for very high access rates.
The ONUs contribute for over 65% to the total PON power consumption. In other words, the ONUs are the major target for energy saving in PONs and their energy efficiency will be much improved if idle ONUs can be properly turned off.
The power consumption of the Internet in the short to medium term future will be dominated by the customer premises equipment and this should be an area of design attention. A key parameter that will affect growth of energy consumption of the network is the rate at which the energy efficiency of the network equipment in the infrastructure improves with successive generations of new technology. The overall energy efficiency of the network will depend on how rapidly network operators choose to replace older equipment with newer technology.
In accord with [4] the traffic volume is predicted to grow at tremendous rates during the next few years, approximately one decade. As shown in Fig. 7, the overall subscriber number is expected to increase moderately and the subscribers tend to use higher-valued services and switch to higher access bit rates over time.

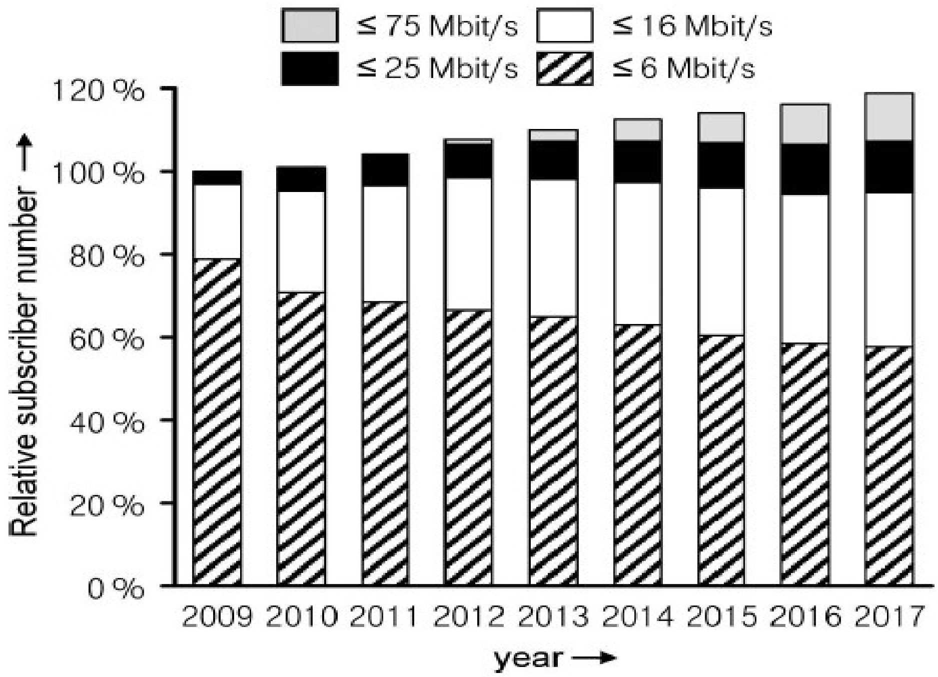


Fig.7 – Forecast of the relative subscriber number changing with the amount of access rate until 2017

The energy demand has been forecasted until 2017 with respect to the reference year (2009). It can be observed that the overall energy consumption increases by approximately 5% per annum (see Fig. 8) in the period under consideration. However, it does not increase to the same extent as the traffic volume. This behavior results from the different proportionality characteristics of different network segments. The energy consumption in home and access networks is proportional to the number of connected subscribers, whereas the consumption in network segments with aggregated traffic (aggregation and core networks) is proportional to the traffic volume.

In the network segments under the responsibility of the network operator, the heaviest energy-consumption increases are estimated in IP/MPLS core networks as well as data centers. The access networks (fixed and mobile) consume largest shares of the overall energy consumption of telecommunication networks owing to lots of active network elements that are widely distributed throughout the field. However, fixed optical access networks are forecast to be the largest contributors to fixed optical communications networks for the next ten years yet [5].

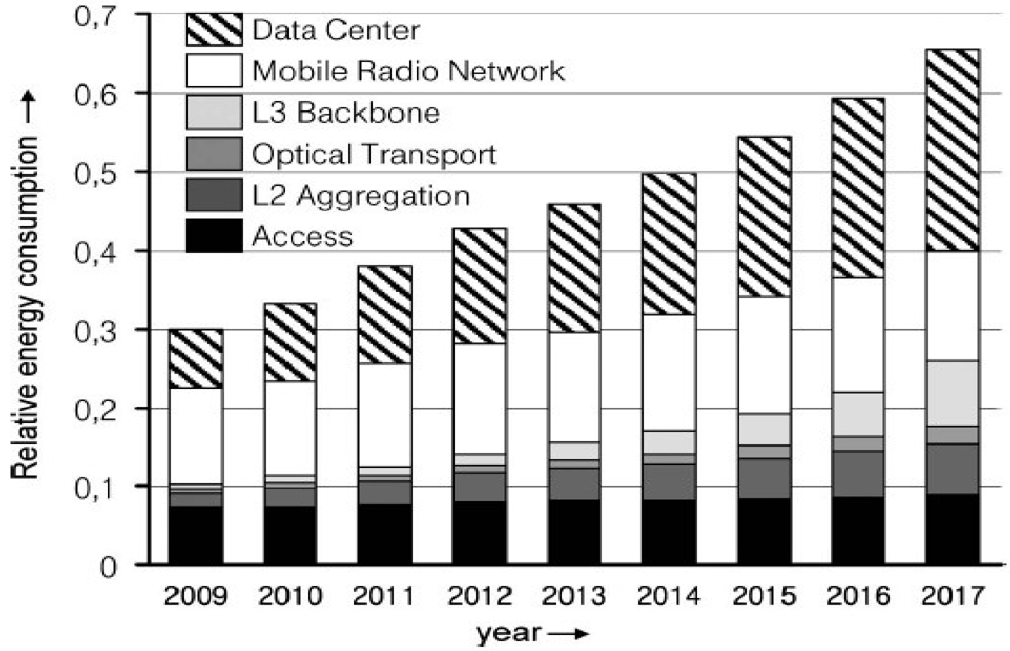


Fig.8 – Forecast of relative energy consumption for each portion of a telecommunication network until 2017

The main reasons why we should care about energy conservation, as explained in [6] are: current energy inefficiencies, enable greater deployment and benefits in the event of a disaster. The high energy cost for the Internet comes about because networking devices expend a great deal of energy even when idle. For the high amount of electricity in many parts of the world, electricity is a scarce resource and this poses one of the barriers to widespread Internet deployment. Furthermore networking equipment in a disaster-hit area will rely on their uninterruptable Power Supplies (UPS) batteries for operation. However, if we can have some form of low-power operating modes, these batteries will last longer. Thus, hospitals, police, and other agencies in the disaster-hit area will be able to access data stored in the affected area for longer.

2 - APPROACHES FOR SAVING ENERGY IN PASSIVE OPTICAL NETWORKS

2.1 – Classification

Today's widespread PON standards are the Gigabit capable passive optical network (GPON ITU-T Rec. G.984.x) with capacity of 1Gb/s, the 10-Gigabit-capable passive optical network (XG- PON ITU-T Rec. G.987.x) with capacity of 10 Gb/s, the Ethernet Passive Optical Network (EPON IEEE 802.3-2008 section 5) with capacity of 1Gb/s, and the 10 Gb/s Ethernet Passive Optical Networks (10G-EPON IEEE 802.3av-2009) with capacity of 10 Gb/s.

The current approaches proposed so far by standard bodies (i.e., ITU-T and IEEE), industry, and academia for implementing energy efficient optical access networks are classified based on the architectural layer. As depicted in Fig. 9, they are implemented in: physical layer, data link layer, and hybrid.

This classification is consistent with both the common GPON layers defined in ITU-T Rec. G.984.1 and with the IEEE 802.3-2008 layered architecture.

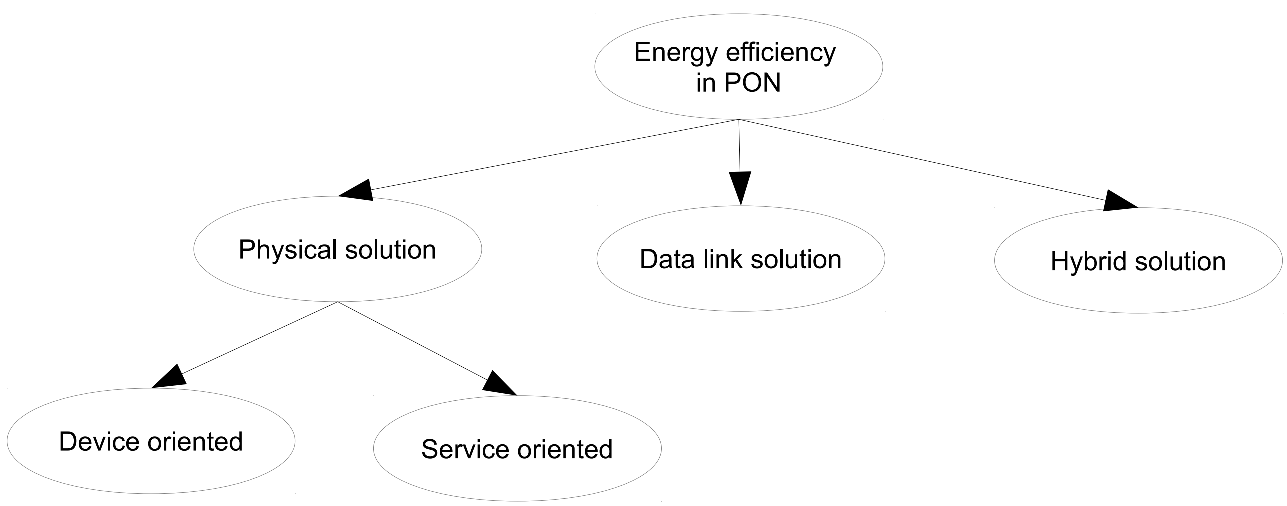


Fig.9 – Classification of energy efficient optical networks

2.2 - Physical layer solutions

Physical layer approaches include solutions aiming at reducing PON energy consumption by targeting the physical layer of the IEEE 802.3 protocol architecture (i.e., the Physical Medium Dependent (PMD) sublayer, the Physical Medium Attachment (PMA) sublayer, and the physical coding sublayer (PCS)). They can be further divided into: device oriented approaches and service oriented approaches. Device oriented approaches aim at reducing the energy consumption of the devices enabling the services provided by each sublayer. Device oriented approaches include equipping transceivers with link rate adaptation (i.e., adaptive link rate (ALR) such as in copper Ethernet), optimizing device energy consumptions, utilizing new modulation formats for data transmission that are more energy efficient. Service oriented approaches aim at improving the performance of the service provided by a sublayer (e.g., the clock recovery within the PMA sublayer) to enable upper layer solutions (e.g., sleep mode). They are often utilized in combination with data link layer solutions to implement hybrid solutions.

The implementation of device-oriented physical layer solutions that reduce physical device energy consumption is strongly advised by standard bodies and European Union (EU) sponsored research centers. For example, the Institute for Energy of the Joint Research Center of the European Commission Directorate General has published a “Code of Conduct on Energy Consumption of Broadband Equipments”(BBCoC) [7].

In the BBCoC, forthcoming year power consumption targets for both Customer Premises Equipments (CPEs) and network equipments are provided. Three CPE operation states are identified in the BBCoC: on-state, off-state and idle-state. A device is in the on-state if all of its components (i.e., interfaces) are fully operational transmitting and receiving a minimum amount of traffic. A device is in the off-state if all of its interfaces are not providing any function. A device is in the idle-state if all of its interfaces are not processing or transmitting a significant amount of traffic.

The BBCoC considers EPON, GPON, 10G-EPON, and XG-PON. By utilizing the data provided in [7] it is possible to compute the target power consumption of a typical 10G- EPON ONU with one Gigabit Ethernet interface, two Plain Old Telephone Service (POTS) interfaces and one Multimedia over Coax Alliance (MoCA) interface. The values summarized in Tab. I show that, even if the targets are met, more than 50% of the power budget is consumed by the ONU central functions (that include also Gigabit Ethernet switch functionalities) and the WAN interface.

Function	Tier 2011-2012 1.1.2011-31.12.2012		Tier 2013-2014 1.1.2013-31.12.2014	
	Idle-state	On-state	Idle-state	On-state
Central functions plus WAN interface	5.6	8.8	5.3	7.7
1 GE interface	0.3	0.9	0.2	0.6
2 FXS (only one interface is on in idle-state)	1x0.5	2x1.5=3.0	1x0.3	2x1.2=2.4
1 MoCA	2.0	2.5	1.8	2.2
Total	8.4	15.2	7.6	12.9

Table I - Power consumption [W] of 10G-EPON ONU (1GE, 2POTS, 1 MoCA)

In [8] is explained one solution within the IEEE P802.3az Energy Efficient Ethernet (EEE) task force (now closed) . EEE uses a Low Power Idle (LPI) mode to reduce the energy consumption of a link when no packets are being sent. The scenario represented is the Ethernet technology with Unshielded Twisted Pair (UTP). For 100 Mb/s and higher data rates, Ethernet transmitters transmit continuously, and when there is no data they transmit an auxiliary signal called IDLE that is used to keep transmitters and receivers aligned. This means that most of the elements in the interfaces are active at all times leading to a large energy consumption. The main function of the LPI is to stop the transmissions when no data are detected and to resume them with new arrivals. Thus, the IDLE signals are not sent during the period of LPI, therefore the alignment of the transmitter and receiver of the current conditions is realized by sending the signals in a small period every large intervals. The energy consumption when the device is in low power mode can be as low as 10 percent that of the active mode.

Another solution is the Adaptive Link Rate (ALR) discussed in [9]. The ALR performs the saving energy by adaptive transmission rates depending on the traffic arrival. A transmitter consumes less energy with low transmission rates so it means that ALR is a smart choice to save energy consumption. The [9] presents a scenario in which the terminals operates at low frequency with the advantage to use Dynamic Voltage Scaling (DVS) that reduce the operating voltage. This allows power to scale cubically, and hence energy consumption quadratically, with operating frequency. In the context of a network link, this translates into sending packets at a constant rate equal to the average arrival rate. However under non-uniform traffic this can result in arbitrary delays and hence it is instead looked for an optimal schedule of rates (i.e.,the set of rates at which the link should operate at different points in time) that minimize energy while respecting a specified constraint on the additional delay incurred at the link.

In the electronic industry as well as in the optical one, component integration is the main device-oriented physical layer solution to reduce energy consumption. As shown in Table II, electronic integration allows to decrease component power consumption. For example, the integration of CDR, LA, and SERDES allows to almost halve the power consumption with respect to utilizing discrete components in EPON.

Variables	EPON	GPON	10G-EPON	XG-PON
	Power [mW]			
CDR	545	520	356	356
CDR + LA	410	410	350	NA
CDR + SERDES	910	790	NA	
CDR + LA + SERDES	610	610		

Table II - Power consumption reduction by means of integration

Several patent applications proposing device-oriented physical layer solutions have been also submitted. One of these is presented in [10] that extends by a low-power sleep logic circuit an amount of time that battery power is available to

an optical network terminal (ONT) after the AC main power has failed.
The low- power logic circuit monitors the main external power source and the battery status, after mains failure, and it selectively switches off ONU interfaces based on the external power source and battery status.
New proposal devices are planned in accord with the device-oriented solutions for decreasing the transmitted power along the fiber in current PON. The research [11] proposes a photonic device, to replace of the Remote node (RN) in a PON, that has a sufficient intelligence to reconfigure the network. It is an active node and for this reason consumes more power than the RN but it has the feature to switch between two states. It consumes energy during periods of network reconfiguration reaching the desired state system. For the rest of the time, the device is able to maintain the current state system without consumption of energy. The use of an active node allow to dynamically reallocate power and wavelength to enhance performance, have remote re-configurability, to enhance security countermeasures, implements a novel tri-state latching mechanism to maintain semi-passive operation even with intelligent remote nodes. However, this solution decreases the required optical power budget only, thus the benefits in terms of energy consumption are not expected to be high.
Concerning about service-oriented physical layer solution, [12] exposes the use of the Manchester modulation instead of the standard Non-return to zero (NRZ). The idea is to keep a commercially available ONU receiver module unchanged and evaluate the energy consumption with the two modulation formats. The system with Manchester modulation can save about 60% of energy because the ONU synchronizes its clock with the OLT clock both in frequency and phase quicker than if the NRZ modulation format is utilized. Another service-oriented physical layer solution can exploit optically powered fiber networks [13]. In optically powered fiber networks remote devices (e.g., sensors) accumulate and store energy within their idle time, and then perform their functions for a short time and send the acquired information. Similar solutions can be applied for remotely powering up a control circuit that wakes up ONUs after a sleep period.

2.3 - Data Link Layer Solutions

Data Link layer approaches target the Data Link layer of the IEEE 802.3 architecture (i.e., the MAC layer) and they are based on the possibility of switching network elements to low power mode. Although the availability of a low power mode must be provided by the physical layer, such approaches can be classified as Data Link approaches because they are based on extensions of the Multi Point Control Protocol (MPCP) and on modified Dynamic Bandwidth Allocation (DBA) algorithms. In principle, Data Link layer approaches require no physical layer modifications but low power mode support in the devices. Research in academia proposed several methods for dynamically switching to sleep mode both EPON and GPON.

ITU-T G.Sup45 [14] proposes three types of power conservation methods, namely power shedding, dozing, and sleeping (further divided in deep and fast sleep). The classification is made in base to the ONU receiver and transmitter behavior. Power shedding powers off or reduces power to non-essential functions and services (e.g., interfaces) while maintaining a fully operational optical link. Dozing powers off ONU transmitter when the ONU does not have upstream traffic to transmit while keeping the ONU receiver always ON. Deep sleep turns off the whole ONU transceiver and all the ONU functions and services, except activity detection. In deep sleep, the ONU loses the incoming downstream traffic and may wake up only when the ONU is switched on on local stimulus or at the expiration of a locally maintained timer. Fast sleep is based on sleep cycles that alternate sleep periods, when the whole ONU transceiver is turned off together with all the non-essential functions, and active periods, when the optical transceiver and the necessary supporting functions are turned on. The OLT learns the capabilities of the ONU to support a specific power saving mode through extensions of the Management and Control Interface (OMCI) management channel and it can also negotiate with the ONU which modes to select.

Transitions to and from a particular power save mode of the ONU can be supported by different signaling messages, such as PLOAM Power Save Mode (PSM) message or by the OMCI framework as proposed in ITU-T Rec. G.988. The qualitative comparison in [14] shows that the OMCI signalling is the most functionally powerful method and it does not require TC layer modification. On the other hand, PLOAM PSM signaling has the deepest impact on the TC layer (introduction of a new PLOAM message), but it may achieve completion of the mode transition faster.

For example, [15] use this signalling mechanism in one of the modes presented. Two power saving modes for GPON are shown: sleep mode and power shedding. The basic idea of sleep mode is to switch off all PON related circuitry, including optics, SERDES, MAC, relevant packet processing, and storage engines when no traffic has to be transmitted. The ONU monitors his activity by status indicators and in case of idle state it requests to enter in sleep mode by sending a sleep request PLOAM message to the OLT . The OLT replies with a sleep approval PLOAM message if the request is accepted, inserting the period time allowed to stay in that mode. Upon the reception of the approval message, the ONU enters in the sleep mode until the expiration of the time allowed or in case of upstream detection whom makes the unit to send a wake-up request PLOAM message to the OLT. If the reply of this message is a wake-up approval PLOAM message, the ONU quits from the sleep mode.

The power shedding, instead, permits to shut down some unnecessary services in order to save energy for preserving a certain number of hours (e.g., 8 hours) of battery power for useful services in emergency situations.

Power shedding is more effective than sleeping and depends on the percentage of sleeping in the whole period of activity. For example, if an ONU sleeping 80% of the time, would reduce the power consumption by 30% (3 Watts savings).

In general, sleep modes for GPON can reduce the power consumption by 10-30% and the power shedding mode in a stand-by fashion can get the power down by 50-80% .

However, in case of power shedding, all the packets arriving to the ONU User to Network Interface (UNI) are lost. Thus, power shedding can be applied only during

long times of ONU inactivity. Moreover, G.Sup45 proposes also a near term solution based on power shedding. The solution allows operators to turn off ONU services during times of the day when they are not needed.
Several vendors are also seeking data link layer solutions for reducing ONU energy consumption. This is witnessed by the large number of patent applications that have been recently submitted. For example, in [16] a cyclic sleep mode is proposed to be implemented in IEEE 802.3 style networks. Both OLT driven and ONU driven methods are considered, but the final decision of switching to sleep mode is taken by the OLT.
Extension to the EPON MPCP protocols to implement the handshake between the OLT and the ONU to switch to sleep mode are also proposed that introduce new messages such as SLEEP, SLEEP REQ, CHANGE SLEEP MODE. Many vendors are also advertising the introduction of power saving modes into their devices [17]. In this case, the ONU supports power saving modes for power consumption reduction during power outages and standby periods.
In [18] two data link layer solutions are presented for reducing energy consumption of EPON and 10G-EPON. The less aggressive solution defines sleep-mode control and sleep aware traffic scheduling to allow ONUs to sleep over one DBA cycle. The more aggressive solution allows ONU to sleep within one DBA cycle. In both solutions, four ONU power levels are defined based on the possibility of selectively turning off the ONU transmitter or the receiver. If both transmitter and receiver are off, even the ONU MAC and SERDES are turned off. In the sleep over one DBA cycle, transmitter or receiver are turned off if no upstream/downstream traffic exists. In the sleep within one DBA cycle approach, the ONU transmitter goes to sleep outside its assigned upstream slot. The ONU receiver sleep time, instead, is based on a downstream traffic scheduling performed by the OLT. The main issues are making the ONU aware of the scheduling to avoid early or late wake up of the ONU. Early wake up implies reduce energy saving while late wake up implies additional delay.
In [19] the performance evaluation of two power saving methods standardized in XG-PON (i.e., ITU-T Rec. G.987.x), namely cyclic sleep (fast sleep) and doze mode

(as defined in ITU-T G.Sup45) is performed through simulations. One of the key features in XG-PON power saving modes is the utilization of timers to regulate the duration of sleep periods. This solution simplifies the coordination between ONU and OLT because timers can be updated yet but not necessarily for each sleep period. Another feature is the utilization of the sleep (or doze) aware state in which ONU periodically (i.e., based on timers) wakes up to collect potential wake-up messages from the OLT. In addition to the timers, the XG-PON power saving solution utilizes scheduling message to schedule ONU into sleep or doze mode. Simulations performed in [19] show that, in cyclic sleep, a larger sleep period (determined by the value of the timer T_{sleep}) allows for larger power consumption reduction. However, if the traffic is low and the T_{sleep} is high, the delay of downstream message is very high (close to 90ms). On the contrary, upstream traffic delay is not affected by the sleep duration because ONU initiated wake-up does not depend on T_{sleep}. In doze mode, the absence of delay impairments on the downstream traffic are improved at the expense of reduced energy saving. Therefore, no upper bound is necessary for T_{sleep}, resulting in the possibility of longer sleep periods with, in turn, similar or even higher energy savings than cyclic sleep.

2.4 - Hybrid solutions

Hybrid solutions are the ones that combine physical and data link layer solutions to reduce energy consumption. One hybrid solution is proposed in [20] for 10G-EPON where EPON and 10G-EPON OLTs and ONUs coexist. The approach is based on combining sleep and periodic wake-up (SPW) (i.e., cyclic sleep in G.Sup45 terminology) and adaptive link rate (ALR).

In the SPW, the OLT requests an ONU to switch to sleep mode because of the absence of downstream traffic. The OLT sets the ONU sleeping time after which the ONU wakes up for requesting the OLT whether it can sleep further or not. In the ALR the downlink rate is switched between 1 Gb/s and 10 Gb/s based on the amount of traffic. It is again the OLT issuing a control message to the ONU for switching between 1Gb/s and 10Gb/s downstream transmission. The ONU replies with an acknowledgement message after having performed the receiver switching. Simulation results show that the hybrid mechanism is capable of providing larger power saving than the two methods in isolation. In addition, in [20] the successful implementation of the proposed method in a testbed is shown.

Another possible hybrid solution could be provided by combining MPCP protocol extensions for sleep mode, such as cyclic sleep [16], with physical layer approaches to remotely power the ONU [13]. In this way, the ONU wake up is directly triggered by the OLT by powering the ONU up.

Other studies propose the combination of modified ONU architectures with sleep mode during ONU idle slots for improving the clock recovery after the ONU wake up [21].

3 - SAVING ENERGY METHODS

3.1 – Introduction to the approach

The approach that this thesis proposes is based on a Sleep and Periodic Wake-up (SPW) control mechanism with variable sleep periods for Ethernet Passive Optical Networks (EPON). In accord to maximize the saving energy consumption at the Optical Network Units (ONUs), the network requirements (e.g., delay) are guaranteed.

The Optical Line Terminal (OLT) manages to save energy at the ONUs through the SPW technique using sleep cycles. The ONUs switch between the sleep mode or the active mode depending on the OLT's decisions.

The SPW technique is presented in [20] and explained in detail in the next sessions. [20] proposes a version in which the sleep period is set by an interarrival-based algorithm between two values.

The main idea of this work is to extend the SPW technique to a service-based algorithm in which the sleep period is set in order to maximize the energy savings for ONUs while guaranteeing the required maximum tolerable delay for different services.

3.2 - SPW operation

The SPW is an energy saving technique with fast sleep mechanism, which consists, if no downstream traffic is present, in a sequence of sleep cycles each composed of a sleep period T_s and an active period T_a. The fast sleep mechanism is classified in the data link approach to implement energy efficiency in PON. The concept in this approach is to power off the ONU's transceiver every time in which its transmitter and its receiver are both idle.

The OLT monitors constantly the behavior of the network and sets the state of the

ONU and the sleep cycle.

Depending on the current state, one ONU consumes different powers: P_a , if it is in active mode, P_s if it is in sleep mode. P_s is lower than but P_a is not zero because the unit doesn't power off completely.

We assume two different approaches in the implementation of SPW using two different alghoritms: interarrival-based and service-based.

The SPW with interarrival-based algorithm is proposed in [20]. The aim of its implementation is, firstly, to check the reliability of our work and, afterwards, to extend the technique to new ideas and new algorithms.

The main difference between the two approaches is how the sleep period is set. Following SPW technique, the OLT sets the sleep period according to traffic conditions such as average frame interval and queue length in the interarrival-based algorithm, and class of service (CoS) in the service-based algorithm. In the interarrival-based T_s is fixed, instead in the service-based T_s changes in function of the delay constraints of subscribed services to guarantee the service performance.

The basic model behavior, showed in Fig 10, is to exchange three kinds of message to make the ONU in the sleep mode or active mode: REQUEST, ACK and CONFIRMATION.

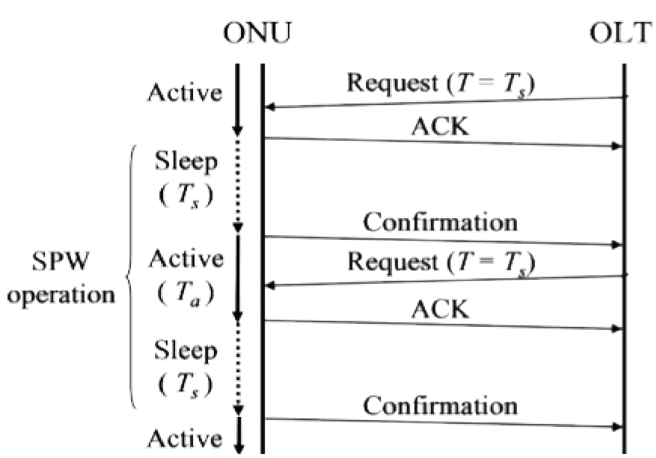


Fig.10 - SPW operation

The OLT sends a REQUEST message to the ONU to set it to sleep for a period T. The ACK message is the reply of the ONU to the REQUEST message and the CONFIRMATION message is sent from the ONU after the sleep period to check the presence of downstream traffic and to inform the OLT of his awake status.

3.3 - Interarrival-based algorithm

The sleep period T, in this approach, is invariable and can assume only two values, T_s and 0, to decide the mode that the ONU have to trigger.

The decision is taken at the OLT, based on the inter-arrival time of the frames that it receives. The downstream traffic is instantly monitored in every arrival frame by the OLT and the inter-arrival time is calculated in average by using an exponential smoothing calculation:

$$i_{a,k} = r \cdot i_{k-1} + (1-r) \cdot i_{a,k-1} \quad (1)$$

The average downstream frame interval $i_{a,k}$ was utilized in place of the instantaneous downstream frame interval, i_k, where subscript k denotes the value at time k. The value r denotes the smoothing factor and it's comprised between 0 and 1, $(0<r<1)$. The subscript k−1 denote the value at time k–1. $i_{a,k}$ is updated to every frame arrival.

Comparing the average inter-arrival time with a threshold, the OLT can predict the behavior of the downstream traffic. The first step of the algorithm is the transmission of the first frame received to the OLT, while the ONUs are in the active mode. The SPW is activated and the OLT sends a REQUEST with T set to T_s if the average inter-arrival time is more than the threshold, otherwise, if the average inter-arrival time is less than the threshold and SPW is not activated yet, nothing changes and the frames in downstream direction are sent, or else, if the mechanism is already enable, a REQUEST with T set to zero and the downstream traffic are sent and the ONUs remain in active mode until the reception of a new REQUEST.

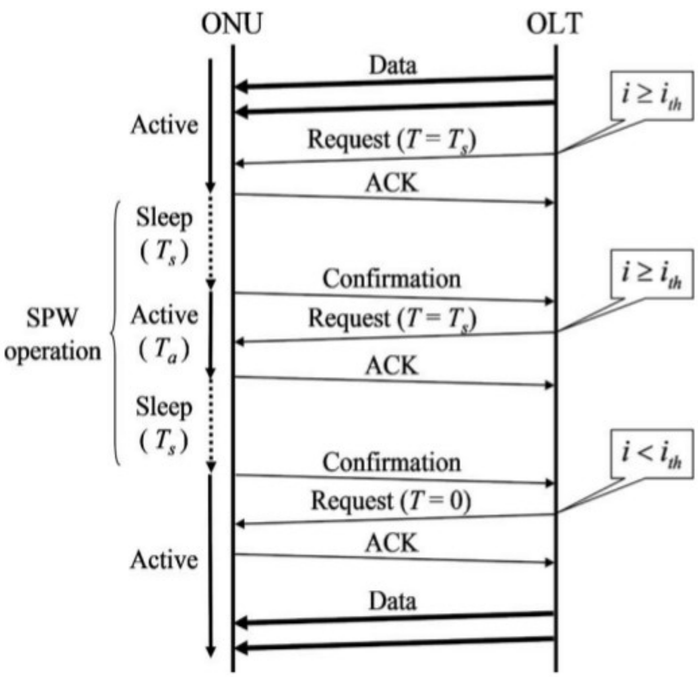


Fig.11 - SPW operation with interarrival algorithm

When the ONU receives a REQUEST, it replies with an ACK to the OLT and, if T is equal to zero, remains in active mode, otherwise it switches to the sleep mode and its transceiver is powered off for the time T_s. When the time expires, the specific ONU enters in active mode and sends a CONFIRMATION to the OLT after an overhead time T_{OH} that reflect the time which the ONU needs to wake up. At this time, the OLT controller takes not only the frame interval but also the queue length into account in determining when the ONU should enter the sleep mode. This is because the OLT has to send the frames buffered in the queue during the

sleep period T_s to the ONU as a matter of first priority. This means that the ONU does not perform SPW operation again until the downstream queue length in the OLT becomes zero.

After frame transfer is completed, in the absence of additional downstream traffic for the ONU, the OLT sends a REQUEST message including sleep period T_s to the ONU. On the other hand, in the presence of downstream traffic for the ONU, the OLT sends another REQUEST message whose sleep period is set to zero in order to keep the ONU active and forwards the downstream traffic. The REQUEST message is also used to acknowledge the CONFIRMATION message.

This process explained is the standard operation in case of the upstream traffic is absent. Indeed, if one ONU receives frames to send in upstream, can reject the

sleep mode. This happen when a REQUEST message arrives with T set to T_s . In this case, the ONU doesn't reply with an ACK but with a NACK, which means that the unit remains in active mode and it's ready to send upstream traffic.
Moreover, also in absence of upstream traffic, the ONU performs the transmission of its frames, DATA and SPW's control, by dynamic bandwidth allocation (DBA) managed in the OLT.

3.4 - Service-based algorithm

The concept of a service-based variable sleep period has been proposed in [22] to save energy in Optical Network Units (ONUs) while avoiding service degradation in the presence of traffic with demanding requirements (e.g., delay). According to this approach, each class of service (CoS) is assigned a specific sleep period. If multiple CoS are received at one ONU, the sleep period of the most demanding class is selected. The feasibility of using the concept of cyclic sleep with adaptable sleep length has also been experimentally proven in [23]. In this work, two methods of estimating packet inter-arrival time are combined with two strategies for choosing the length of the sleep period as a function of the estimated inter-arrival time. As a result, the authors in [23] present four methods for choosing the most suitable sleep period for a certain predicted value of inter-arrival time.
Usually one CoS is characterized by strict and absolute Quality of Service (QoS) constraints that must be met by the service provider to adhere to the Service Level Agreement (SLA) stipulated for users residing at the ONU. In addition, users (i.e., ONUs) may dynamically modify their service subscriptions (e.g., request a video streaming to watch a movie for some hours), possibly changing the overall delay requirements.
Utilizing the service-based variable sleep period concept, this paper proposes a method for maximizing the energy savings for ONUs while guaranteeing the

required maximum tolerable delay for different services. The approach differs from the one proposed in [22] because the chosen sleep period is a function of the maximum allowed delays rather than based on the estimated average inter-arrival time and the highest-priority class. Moreover, it differs from the approach proposed in [22] and [23] because the sleep time is a function of the CoS absolute delay constraints instead of the estimated packet inter-arrival time.

We assume that the Optical Line Terminal (OLT) is constantly aware of the services subscribed by each ONU. In addition, the OLT has a table in which services are categorized in terms of CoS with specific absolute bandwidths (i.e., data rates) and delay constraints. Such constraints can be derived from ITU-T standards such as [24,25] and statically loaded into the OLT. Leveraging on such information, the OLT dynamically builds a table containing the minimum bandwidth and maximum delay requirement per each registered ONU. The minimum bandwidth is the sum of all the minimum bandwidths (in this work, Poisson arrival process is assumed) while the maximum delay is the minimum of all the maximum acceptable delays required by the services to which the ONU is subscribed.

Tab. III shows several typical applications with their corresponding delay requirements and data rates. Note that the delays shown in this table are the strict delay requirement for the access segment only that can be inferred from contributing portion of access network to overall end-to-end delay of an Internet application presented in [25].

Service Class	Service Type	Maximum Delay [ms]	Data Rate Bi [b/s]
1	Web Browsing	220	30.5K
2	Internet Relay Chat	110	1K
3	Multimedia on Web	84	28.8-500K
4	Voice over IP	56	5.3-64K

Tab. III - Applications with their corresponding delay requirements, data rates and service class

The ONU implements a slightly modified version of Sleep and Periodic Wake-up

(SPW) (i.e., cyclic sleep) [20] whose behavior is depicted in Fig. 12. The main difference from SPW of the interarrival-based approach is that no threshold for the frame inter-arrival time is used to set the ONU sleep time. One period of activity T_a of the ONU used for exchanging Confirmation, Request, and ACK messages with the OLT is always followed by sleep period of duration T_{sl}. Moreover, whenever the OLT receives a Confirmation message from the ONU, it sends all data frames received before the Confirmation reception (e.g., frames n-2, n-1, and n in Fig.12) to the ONU. This operation requires a variable time T_{DS}.

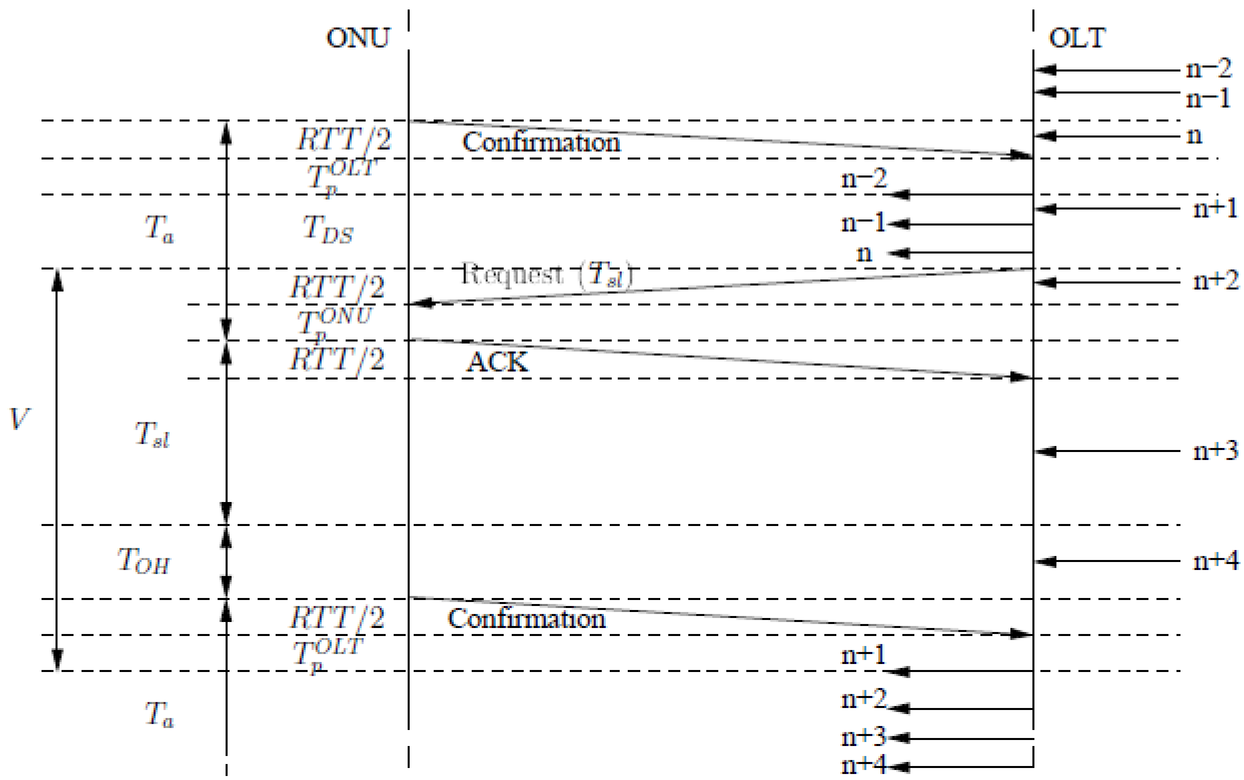


Fig.12 - SPW operation wiith the service-based algorithm

A first approximated model of the system OLT-ONU with the modified SPW is a M/G/1 queue with vacations [27]. The average waiting time of a frame in the queue is expressed by:

$$\overline{W}_{q,MG1} = \frac{\lambda \cdot \overline{S^2}}{2 \cdot (1 - \lambda \cdot \overline{S})} + \frac{\overline{V^2}}{2 \cdot \overline{V}} \quad (2)$$

where S is the average service time, $\overline{S^2}$ is the second moment of the service time, λ is the average frame arrival rate, $\overline{V^2}$ is the second moment of the vacation time and $\overline{V}$ is the average vacation time.

A more detailed model that has been considered is a polling system with gated service policy with identical stations [26]. The average waiting time of a frame in the queue $\overline{W}_{q,poll}$ is expressed by:

$$\overline{W}_{q,poll} = \frac{\overline{V^2} - \overline{V}^2}{2 \cdot \overline{V}} + \frac{N \cdot \overline{V} \cdot (1 + \lambda \cdot \overline{S})}{2 \cdot (1 - N \cdot \lambda \cdot \overline{S})} + \frac{N \cdot \lambda \cdot \overline{S^2}}{2 \cdot (1 - N \cdot \lambda \cdot \overline{S})} \quad (3)$$

where N is the number of ONUs. Assuming negligible processing time T_p at both ONU and OLT and transmission time of the control frames, the vacation can be written $V = T_{sl} + T_{OH} + RTT$, where T_{OH} is the clock and frame synchronization overhead and RTT is round trip time. For networks in which the service time is significantly smaller than the sleep time and $\lambda \cdot \overline{S} \ll 1$, Eq. (3) can be approximated as:

$$\overline{W}_{q,poll} \approx \frac{N \cdot \overline{V}}{2} \quad (4)$$

For example, in a scenario with 1250 byte frames, transmission rate of 10Gb/s, frame arrival rate λ = 62.5 frame/s (i.e., 625 kb/s), T_{sl} = 1 ms, T_{OH} = 2 ms,

N = 1, and RTT = 0.4 ms the three contributions to $\overline{W}_{q,poll}$ in Eq. (3) are 0, $1.7\text{x}10^{-3}$, $3.125\text{x}10^{-11}$, respectively. Because V is constant, it is possible to write:

$$\overline{W}_{q,\,poll} \approx \frac{V}{2} = \frac{T_{sl}+T_{OH}+RTT}{2} \quad (5)$$

Solving the equation (3) and considering only one station (N = 1), (3) can be written:

$$\overline{W}_{q,\,poll} = \overline{W}_{q\,,MG1} + \frac{\overline{V}\cdot\lambda\cdot\overline{S}}{1-\lambda\cdot\overline{S}} \quad (6)$$

The average queuing delay of the two systems can be approximately the same only if $\lambda\cdot\overline{S}\ll 1$ and $\overline{S}\ll T_{sl}$ with V constant. Then:

$$\overline{W}_{q,\,poll} \approx \overline{W}_{q\,,MG1} \quad (7)$$

Fig. 13 shows the algorithm for OLT to determine sleep period for an ONU. The algorithm is based on substituting $\overline{W}_{q,poll}$ with the maximum delay D_{max} and deriving T_{sl} as a function of the other quantities in Eq. (5).

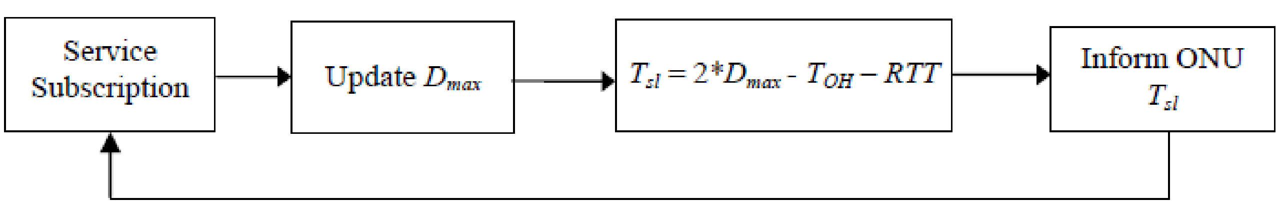


Fig.13 - Algorithm for the service-based method

The algorithm starts with the service subscription of each ONU. The OLT builds or updates the table for each ONU in which the service subscribed are categorized in terms of CoS with specific absolute bandwidths and constraint delay. Thus the maximum delay D_{max} of a service composition is calculated from the table and used in input to derive T_{sl}. The last step is to inform the ONU about the sleep period through a Request message.

4 - RESULTS

4.1 – Scenario

We built a model in OPNET Modeler, a network modeling and simulator tool, to reproduce the Ethernet Passive Optical Network (EPON) architecture and to evaluate different energy saving techniques, in particular using sleep cycles for Optical Network Units (ONU).

OPNET has a hierarchical architecture, in a manner that emulates real network systems. Specialized editors address issues at different levels of the hierarchy. This provides an intuitive modeling environment and also permits re-use of lower level models.

The bidirectional transmissions in a EPON are performed by a point-to-multipoint (P2MP) network between a unit called Optical Line Terminal (OLT) placed in the Central Office (CO) and one Optical Network Unit (ONU) interfaced to the client network.

A single fiber is used to serve multiple premises, exploiting passive components, like optical splitters, by a bus topology with 1x2 optical tap couplers to attach the nodes to the bus as follow in the Fig. 14.

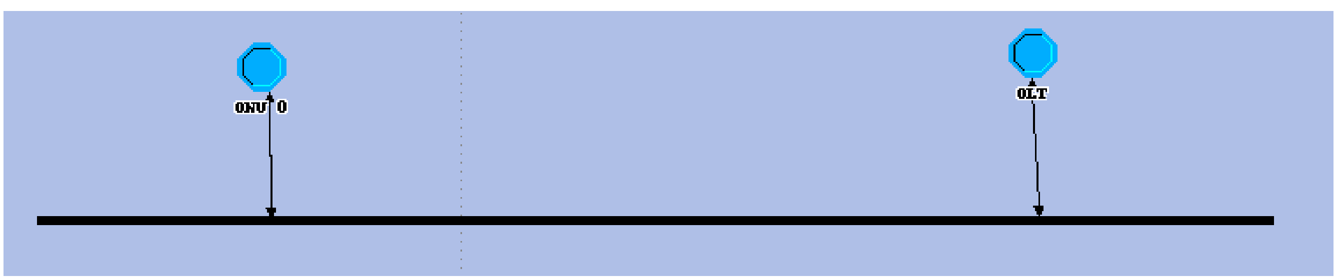


Fig.14 - Bus topology of the TDM-PON model

The bus topology is chosen to simplify our model. The bidirectional communication is the same as a tree topology with a passive splitter following the same policy in which the ONUs can transmit frames only to the OLT sharing the optical fiber and the OLT sends traffic in broadcast to all the ONUs using LLID. The policy is obtained in the simulator by a pipeline “closure stage” forbidding the communication between ONUs and the Dynamic Bandwidth Allocation (DBA) algorithm to schedule the upstream traffic of each unit.

We set the distance between the OLT and the ONU in terms of propagation delay T_{pr} to 0.2 ms.

Since we are studying a 10G-EPON, the transmission capacity of the bus link and the optical tap couplers is set to 10 Giga bit/s.

The traffic flows downstream (from the network to users) and upstream (from the users to the network) using two different wavelengths. All the upstream signals can operate in the same wavelength in a Time Division Multiple Access (TDMA) managed by the Multi-Point Control Protocol (MPCP) assigning to the ONUs the transmission windows. There is no frame loss, since all downstream frames coming when the ONU could not receive them were buffered at the OLT .

For sake of simplicity, only downstream traffic from one OLT to one ONU is considered in simulation.

In the SPW technique, the ONU can switch between the active mode and sleep mode. Depending on which mode is enabled, the unit consume different power, in particular, P_a equal to 10 W in active mode and P_s equal to 1 W in sleep mode. The time that the ONU needs to trigger the active mode from the sleep mode is T_{OH} that is set to 2 ms.

Other parameters and algorithms are specific for approach.

4.2 - Interarrival-based approach

In the interarrival-based approach, we assume that the frames are generated with arrival rate following Poisson distribution and constant size, set to 1250 Byte. The maximum downstream bandwidth for each ONU running the 10G downlink is set to 1 Gb/ s.

The OLT decides to switch the ONU in a specific mode based on the average interarrival time calculated to every arrival frame. The OLT compare the average inter-arrival time with a threshold i_{th}. We run three sets of simulation changing the value of the threshold: 10 ms, 20 ms and 40 ms.

For this approach the sleep period T_s is only one and set to 10 ms.

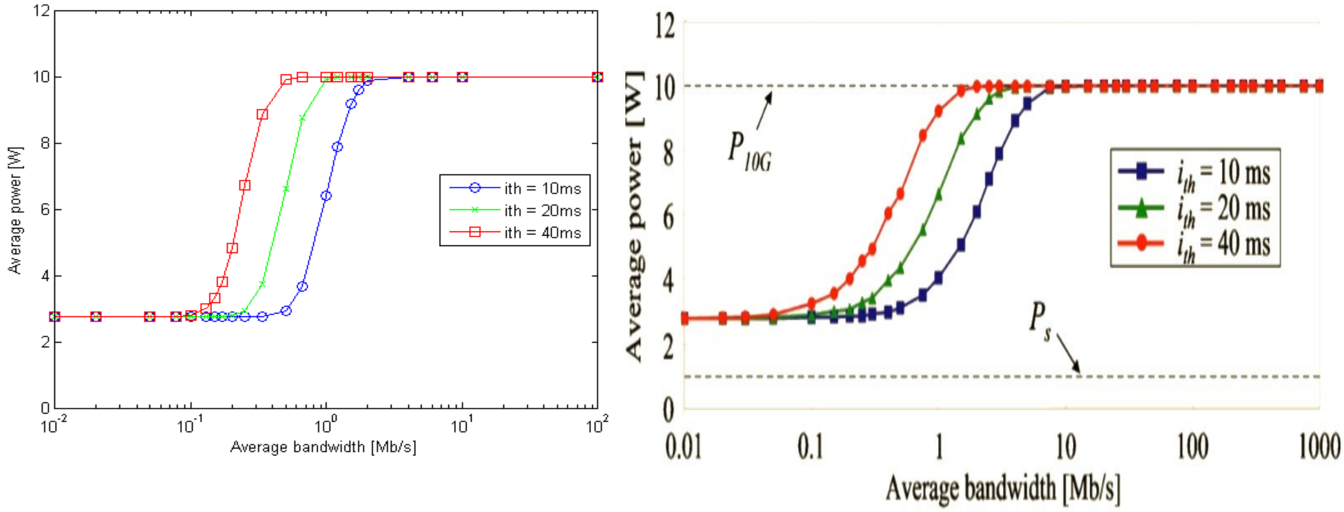
We studied the average power consumption at the ONU and the average queuing delay of our model fixing the average downstream bandwidth. The results,shown in the Fig. 15,16 , were collected inserting in input the average inter-arrival time of the frames. The set of inter-arrival time is: [1 0.5 0.2 0.13 0.1 0.077 0.067 0.059 0.05 0.04 0.03 0.02 0.015 0.01 8.33e-3 6.67e-3 5.88e-3 5e-3 2.5e-3 1.67e-3 1e-3 1e-4] second. The choice to use these values reside to have more contributions on the trend of the curves and less in the portion in which the curves reach the stationary.

The average power is calculated by

$$P_{ab} = \frac{T_a \cdot P_a + T_s \cdot P_s}{T_a + T_s} \qquad (8)$$

The time T_a is the duration of the ONU in active time. In absence of downstream traffic, it is the sum of the overhead time, T_{OH}, and the round-trip transmission delay between the OLT and ONU, equal to $2 \cdot T_{pr}$. Otherwise the active time is variable and equal to $2 \cdot T_{pr} + T_{DS}$, where T_{DS} is the time to send the frames in queue arrived during the previous sleep period.

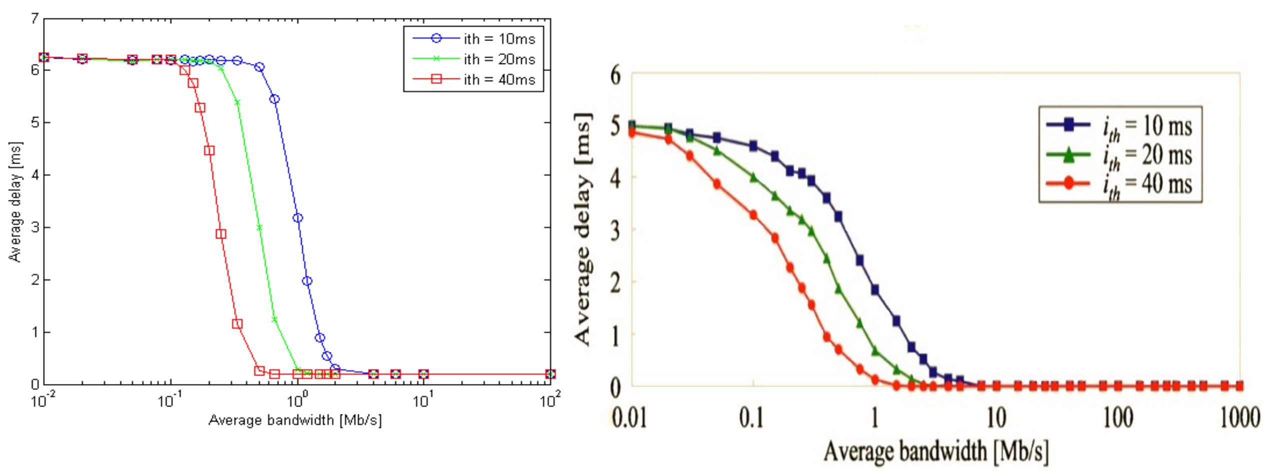
The objective to implement this approach is to compare the results of our simulations with the ones showed in the paper [20]. As shown in the Fig. 15 and Fig. 16 the results obtained are very close with the same increasing trend. The theoretical average power for low bandwidth was reached as no downstream traffic was presented and the SPW was activated for all the simulation time. Indeed for a value of 0.01 Mb/s, the average power is 2.74 W.



(a) **(b)**

Fig.15 - Comparison between the average power simulation results (a) and the published results (b)

On the other hand the maximum average power was reached for high bandwidth because the ONU was continuously awake. Between 1 Mb/s and 10 Mb/s in both the results the average power is 10 W the same value of power in which the ONU is in active mode.



(a) **(b)**

Fig.16 - Comparison between the average delay simulation results (a) and the published results (b)

Differently from [20] the maximum average delay was around 6.2 ms instead of 5 ms because the transition time from the sleep to the awake state as well as the RTT of the exchange messages were considered in the calculation of queuing delay. The same motivation applies to the lowest average delay.

4.3 - Service-based approach

In the scenario where the service-based approach is applied the data frames arrive at the OLT according to a Poisson process with arrival rate λ frames/s depending on the ONU service composition. We run two sets of simulation, in the first set only one source is used to generate traffic with the bandwidth of a service composition and in the second a source generates traffic for each service composition.

In both the sets, the simulations are run twice with different frame size f_s , one is constant and equal to 1250 bytes, the second one is uniformly distributed between 72 and 1526 bytes, respectively the Ethernet minimum and maximum frame size.

Without loss of generality, this paper limits the number of traffic classes to 4 and considers several class combinations as shown in the left part of Tab. IV. For classes with a range of data rates, the maximum one has been considered to compute λ. The values of λ^{-1} reported in Tab. IV have been computed by following formula:

$$\lambda^{-1} = \frac{8 f_s}{\sum_{i \in S} B_i} \qquad (9)$$

where S is the service composition.

The bandwidth for a service composition is the sum of the bandwidth B_i of all the

services that are part of it, assuming frames arrive following a Poisson process.
The evaluated performance metrics are the frame queuing delay D and the energy efficiency η. Here, the efficiency is defined as

$$\eta = 1 - \frac{E_{sleep}}{E_{nosleep}} \quad (10)$$

where E_{sleep} is the energy consumed by the ONU implementing the proposed sleep mode and $E_{nosleep}$ is the energy consumed by the ONU if it is always active.
The results are collected by post processing operations setting in input the T_{sl} and the average inter-arrival time values in input to the simulator for a specific service composition. The T_{sl} is calculated as

$$T_{sl} = 2*D_{max} - T_{OH} - RTT$$

where D_{max} is the maximum constraint delay of a service composition taken from the table that the OLT built per each registered ONU; T_{OH} is the time of overhead in which the ONU switches from the sleep mode to active mode; RTT is the round-trip time between OLT and ONU.
The inter-arrival time is calculated as in (9)
The output parameters of the system are the queuing delay for the frames with constant and uniform distributed size and the energy efficiency calculated as in (10). Moreover the confidence interval with confidence level of 0.9 is calculated for both output parameters.
The results about the first set of simulation are shown in the Tab. IV. In the right part of the table is indicated that in any service composition, ONU can save up to around 89% of energy while not violating any delay requirements. The advantage comes from the method of determining the sleep time that allows the ONU to maximize the sleep time, thus maximizing the energy savings, without violating the service delay constraints.

Input Parameters				Output Parameters				
Service Composition	λ^{-1} [ms/frame]	D_{max} [ms]	T_{sl} [ms]	$D_{constant}$ [ms]	$D_{uniform}$ [ms]	Efficiency η [%]	Conf. Interval $D_{constant}$, $D_{uniform}$ (conf.lev. 0.9) [ms]	Conf. Interval η (conf.lev. 0.9)
1+2+3+4	17	56	109	55.7	55.7	88.01	$\pm 1.8 \cdot 10^{-1}$	$\pm 3.0 \cdot 10^{-4}$
1+2	317	110	217	109.7	109.7	89.01	$\pm 3.7 \cdot 10^{-1}$	$\pm 1.3 \cdot 10^{-4}$
2+3	20	84	165	83.7	83.7	88.66	$\pm 0.7 \cdot 10^{-1}$	$\pm 3.2 \cdot 10^{-4}$
4	156	56	109	55.7	55.7	88.05	$\pm 5.4 \cdot 10^{-1}$	$\pm 1.9 \cdot 10^{-4}$

Table IV – Results per service composition with one generator of traffic

The reliability of the formula used for this system is compared with our simulations. The results are taken with a T_{sl} set and changing the average bandwidth of the traffic generated:

$$B = \frac{10}{2^x} Gbit/s \qquad (11)$$

with x = {0, 1, 2, 3, 4, 5, 6}.

The average inter-arrival time is calculated as in (9).

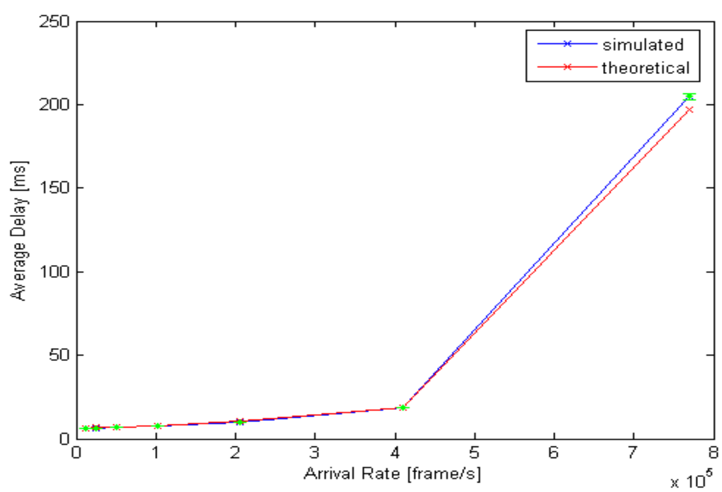


Fig.17 – Comparison between the theoretical and simulated average delay

How the Fig. 17 shows, the curves of the theoretical average delay and the simulated average delay are exactly the same until going close to the limit of transmission rate of 10Gb/s.

The last step is to verify that the results collected about the average delay D, considering only one generator, are the same considering one generator per service with the own arrival rate.

The second set of simulation is launched to verify that the simplification to use only one generator with the bandwidth of a service composition was correct and if the service with the maximum constraint still respect the acceptable delay. The results are shown in the Tables V, VI, VII.

Service Composition: 1+2+3+4 with T_{sl} = 109 ms								
Input parameters			**Output parameters**					
Service	λ^{-1} [ms/frame]	D_{max} [ms]	$D_{constant}$ [ms]	$D_{uniform}$ [ms]	Effic. η [%]	Conf. Int. $D_{constant}$ (conf.lev. 0.9) [ms]	Conf. Int. $D_{uniform}$ (conf.lev. 0.9) [ms]	Conf. Int. η (conf.lev. 0.9)
1	328	220	55.9	55.6	88.01	$\pm 7.8 \cdot 10^{-1}$	$\pm 7.8 \cdot 10^{-1}$	$\pm 3.0 \cdot 10^{-4}$
2	10000	110	55.3	55.1		± 4.9	± 5.3	
3	20	84	55.7	55.7		$\pm 1.9 \cdot 10^{-1}$	$\pm 1.9 \cdot 10^{-1}$	
4	156	56	55.9	55.7		$\pm 5.4 \cdot 10^{-1}$	$\pm 5.4 \cdot 10^{-1}$	

Table V - Results for a service composition 1+2+3+4 with a generator of traffic per service

Service Composition: 1+2 with T_{sl} = 217 ms								
Input parameters			**Output parameters**					
Service	λ^{-1} [ms/frame]	D_{max} [ms]	$D_{constant}$ [ms]	$D_{uniform}$ [ms]	Effic. η [%]	Conf. Int. $D_{constant}$ (conf.lev. 0.9) [ms]	Conf. Int. $D_{uniform}$ (conf.lev. 0.9) [ms]	Conf. Int. η (conf.lev. 0.9)
1	328	220	109.7	109.7	89.01	$\pm 3.8 \cdot 10^{-1}$	$\pm 3.8 \cdot 10^{-1}$	$\pm 1.3 \cdot 10^{-4}$
2	10000	110	109.7	109.6		± 2.1	± 2.1	

Table VI - Results for a service composition 1+2 with a generator of traffic per service

Service Composition: 2+3 with T_{sl} = 165 ms								
Input parameters			**Output parameters**					
Service	λ^{-1} [ms/frame]	D_{max} [ms]	$D_{constant}$ [ms]	$D_{uniform}$ [ms]	Effic. η [%]	Conf. Int. $D_{constant}$ (conf.lev. 0.9) [ms]	Conf. Int. $D_{uniform}$ (conf.lev. 0.9) [ms]	Conf. Int. η (conf.lev. 0.9)
2	10000	110	83.5	83.7	88.66	± 1.6	± 1.6	$\pm 1.9 \cdot 10^{-4}$
3	20	84	83.7	83.7		$\pm 0.7 \cdot 10^{-1}$	$\pm 0.7 \cdot 10^{-1}$	

Table VII - Results for a service composition 2+3 with a generator of traffic per service

The results in the tables show the correctness of our previous assumptions to consider the bandwidth of a service composition as the sum of the bandwidth of each service. We obtained values of average delay in both situations of constant and uniform distributed packet size in accordance with the delay constraints for each service in all combinations. The average delay for each service of a combination service respects the own constraint delay. Besides, the energy efficiency of each combination is the same as in the case of where only one generator is being used and so maximize to a value of around 89%.

The bad aspect of our assumptions is that the traffic generated is not typical of a real system, for this reason a possible future work is to use a distribution (e.g., Pareto distribution) for frame size more similar to the traffic of commercial telecommunication systems.

5 - CONCLUSIONS AND FUTURE WORK

This work focused on the reduction of energy consumption in passive optical access networks. Different schemes have been presented to save energy at the ONUs by cyclic sleep periods reducing the amount of idle periods. In particular an energy saving technique with Sleep and Periodic Wake-up (SPW) mechanism with two different approaches was implemented.

Firstly the interarrival-based approach presented in [20] was implemented to compare the published results with our simulations to verify the reliability of our model. The results obtained closely match and the differences is due to different assumptions. Then a new service-based variable sleep period method for PONs is implemented which can guarantee maximum tolerable delays while maximizing the energy efficiency at ONUs. The method for OLT to determine the sleep time for an ONU is based on a simple yet effective analytical model. The only control parameter used to calculate the sleep time is the strictest delay constraint among all the services the ONU subscribes to. Four service combinations were considered in the scenario and the simulation results were collected in two manners to generate the traffic for each service. In a first approach one source generates traffic with the bandwidth of a service composition as the sum of the bandwidth of each service (assuming that the traffic is in accord with Poisson distribution). Then a specific generator is implemented for each service. The results obtained are the same and confirm that the use of a generator with the service composition bandwidth is reasonable. The proposed method successfully exploits the acceptable delay to maximize the sleeping duration of ONUs resulting in a significant energy efficiency improvement for PON systems.

The work will continue with new ideas to improve and extend both approaches discussed previously. In particular the next step is to implement sources that generate more realistic traffic in accord with the current one that flows in real architectures. Moreover the work will be extended considering more ONUs and upstream transmissions.

APPENDIX A

PON OPNET simulator

A.1 - INTRODUCTION

This appendix explain the details of the implementation of a TDM-PON architecture in the simulator tool, Opnet Modeler.

In particular an EPON architecture with Interleaved Polling with Adaptive Cycle Time (IPACT) bandwidth allocation using the GATED discipline and the Sleep and Periodic Wake-up (SPW) operation for saving energy presented in [20] was implemented in our model.

With the gated algorithm, the OLT grants enough time to send all the traffic reported by each ONU.

The basic model behavior of the SPW is to exchange three kinds of message to make the ONU asleep or awake: REQUEST, ACK and CONFIRMATION. The OLT sends a REQUEST message to the ONU to set the period T in which the unit must sleep. The ACK message is the reply of the ONU to the REQUEST message and the CONFIRMATION message is sent from the ONU after the sleep period to check the presence of downstream traffic and to inform the OLT of his awake status. The traffic received by the OLT during a sleep period is sent after receiving a CONFIRMATION.

A.2 - Simplifications in the implementation of the model

The model of an EPON has been developed by making as many simplifications as possible without compromising the performance between our model and the real-life EPON system.

Concerning the two modes of operation of MPCP, some simplifications have been made about the autodiscovery operation, whose task is to register new ONUs and the clock synchronization between ONUs and OLT.

Registration in a EPON is managed by allocating the discovery slot, where unregistered ONUs can register themselves by issuing register messages. Regular upstream data traffic is forbidden during this interval. Collisions of registering messages coming from different unregistered ONUs can occur, which are resolved by repeated trials (in successive discovery windows) until they succeed.

Registration has the effect of lowering the available bandwidth and extending the polling cycle (it can be seen as an "additional" ONU whose upstream allocation is deterministic).

Since a good architecture should be designed so as to waste only little bandwidth, we considered the impact of autodiscovery in our statistics as negligible. Furthermore, autodiscovery overhead is independent of network load, so the penalty is fixed with respect to the amount of traffic generated by ONUs.

This means that the autodiscovery is not implemented, assuming that the number of ONUs, their LLID and the propagation delay between the OLT and each ONU are already known.

Those parameters could be obtained by calling the function <epon_get_config_kubo()>; its implementation can be found in the external source file "ak_epon_config_kubo", while the association of an ONU to its LLID is controlled by an attribute of the node which implements it.

A.3 - Implementation

OPNET has a hierarchical architecture, similar to the one of parallel real network systems. Specialized editors address issues at different levels of the hierarchy. This provides an intuitive modeling environment and also permits re-use of lower level models.
The OPNET simulator is composed of 4 different levels: Network, Node, Process and Proto-C.

A.3.1 - Network level

The Network level is the highest level of the simulator presented when a project is opened. The objects of this level are the nodes and the subnetworks within a geographic context for network model development that provides an intrinsic notion of distance. Several different types of communication link architectures are provided to interconnect nodes that communicate with each other. OPNET Modeler provides simplex (unidirectional) and duplex (bidirectional) point-to-point links to connect nodes in pairs and in a bus link to allow broadcast communication for arbitrarily large sets of fixed nodes.
The Passive Optical Network (PON) is a Multi-Point network developed in a tree topology, bus topology or ring topology in which the transmissions are performed between OLT and ONUs.
In the Network level of this model, the EPON architecture is presented by a bus topology with two kinds of node models, OLT and ONU, attached by 1x2 optical tap couplers as shown in the Fig.A1.

Fig.A1 Model configuration in Network level

The model is implemented in “Kubo.prj”.

A.3.1.1 - Bus

The link model adopted in this scenario is the bus link to emulate the shared fiber of a real EPON architecture. The features of a link in OPNET are described and implemented through pipeline stages. The bus transceiver pipeline consists of six stages: transmission delay, closure, propagation delay, collision, error allocation and error correction.

In this project, it's assumed that the errors in the link model are omitted and the collisions are managed by the MPCP, so only the transmission delay, closure and propagation delay stages have been considered and described in the next section.

The link model is defined in “ak_epon_link_kubo”.

A.3.1.1.1 - Transmission delay stage

The transmission delay stage is the first stage of the pipeline, and is specified by the “txdel model” attribute of the bus link. It is invoked immediately upon beginning the transmission of a packet.

This stage is invoked to calculate the amount of time required for the entire packet to complete transmission. The result is the simulation time difference between the beginning of transmission of the first bit and the end of transmission of the last bit of the packet.

The transmission delay stage is defined in “ak_epon_link_tx_delay_kubo”.

A.3.1.1.2 - Closure stage

The closure stage is the second stage of the pipeline, and is specified by the "closure model" attribute of the bus link. It is invoked once for each receiver attached to the bus. These invocations take place immediately after the return of the transmission delay stage, with no simulation time elapsing in between. The purpose of this stage is to determine whether a particular receiver is capable of receiving a transmission.

In detail, the sender and receiver of the packet are detected and the packet is dropped if:

• the sender and the receiver are the same entity;

• the sender and the receiver are both ONU.

Sender and receiver identities are specified in terms of Object id (Objid). Since the first packet that will enter the bus is forwarded by the OLT, the Objid of the sender will be set with OLT's Objid.

Implemented in "ak_epon_link_closure_kubo".

A.3.1.1.3 - Propagation delay stage

The propagation delay stage is the third stage of the pipeline, and is specified by the "propdel model" attribute of the bus link. It is invoked by each receiver that is attached to the bus; this invocation takes place after the return of the closure stage with no simulation time elapsing in between. The purpose of this stage is to calculate the amount of time required for the packet's signal to travel from the bus transmitter to the bus receiver.

In this model, the value is fixed and set in the external source file "ak_epon_config_kubo". The pipeline stage gets the value from the LLID encapsulated in the frame and calls the function "epon_get_config_kubo".

The implementation is in "ak_epon_link_delay_kubo".

A.3.2 - Node level

The Node level defines the features of a node, the main entity of communication in the Network level. The node models are created by objects called modules that describe specific behaviors. The principal modules in a node are the transmitter, receiver, processor and queue. The transmitter/receiver are like a gateway on the Network level that allow a transmission/reception between the node, to which they belong, and the other nodes. The queue and the processor objects can turn into programmable modules by changing the process model attribute. Normally, a processor module would be used in cases where a packet can be completely processed in response to the interrupt associated with its arrival or generation. If this is not the case, i.e., it is necessary to buffer the packet while awaiting a later event to complete processing, then a queue module, with its additional buffering resources, is likely to be more correct. This is particularly true if multiple packets must be buffered simultaneously.
Some standard process models are already included in Opnet but it's possible to create other models depending on what a network developer needs. The function of a module is given by the Finite State Machine (FSM) in the Process Level instantiated in a process model.
The standard process models used in the EPON model are the sink and the simple source. The sink model function is to get the packet's statistics, instead the simple source module generates packets and specifies the creation rate of the packets in terms of packets per second, the distribution of the packets’ size and the start/stop time of their generation.
The interaction between the modules is performed by three kinds of connections: packet streams, statistic wires and logical associations. Packet streams allow formatted messages called packets to be conveyed from one module to another by statistic wires. Statistic wires convey simple numeric signals or control information between modules, and are typically used when one module needs to monitor the performance or the state of another. Logical associations identify a binding between modules. Currently, they are allowed only between transmitters and receivers to indicate that they should be used as a pair when attaching the

node to a link in the Network Domain.
The node models created are the ONU node and the OLT node.

A.3.2.1 - OLT node

The OLT node, in Fig. A2, describes the main functions of the OLT: reception of downstream traffic from the core network, forwarding of downstream traffic to each ONU, reception of upstream traffic from each ONU, forwarding of upstream traffic to the core network, implementation of MPCP to manage the upstream traffic and implementation of the SPW technique to save energy and to manage the downstream traffic.
The downstream reception is modelled by using unique simple generator of packets where one specific source generates downstream traffic for one specific ONU included in the model.
The downstream forwarding and the upstream reception is managed by a transmitter and a receiver module interfaced on the bus link, instead the upstream forwarding is omitted.
The implementation of the MPCP and SPW is defined in the "olt" module in which the process model is "ak_epon_olt_manager_kubo" (Fig. A3).

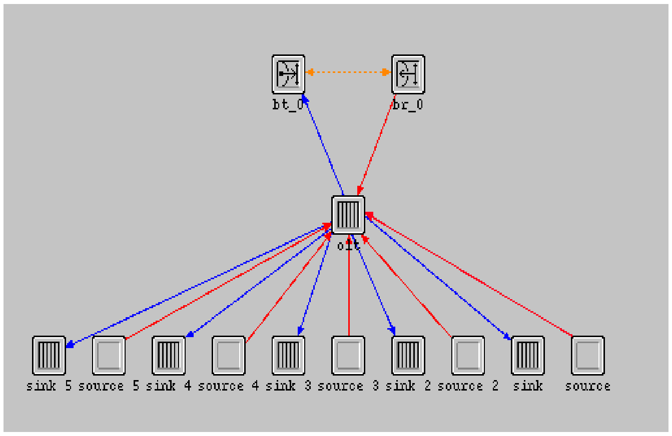


Fig.A2. OLT node

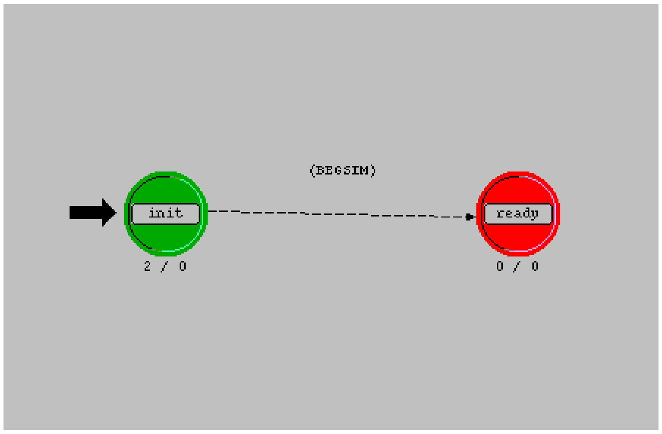


Fig.A3."ak_epon_olt_manager_kubo"

A.3.2.2 - ONU node

The ONU node, in Fig. A4, describes the main functions of the ONU: reception of downstream traffic from the OLT, forwarding of downstream traffic to the users, reception of upstream traffic from the users, forwarding of upstream traffic to the OLT, and implementation of MPCP and SPW.

The downstream reception and upstream forwarding are managed by a receiver and transmitter module interfaced on the bus link.

The downstream forwarding to the users is omitted and the upstream reception from the users is modelled using a generator of packets.

The implementation of the MPCP and SPW is defined in the “mac” module in which the process model is “ak_epon_onu_manager_kubo” (Fig. A5).

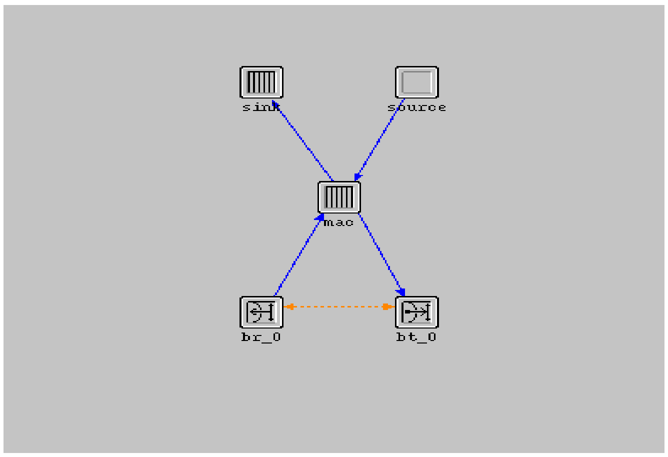


Fig.A4. ONU node

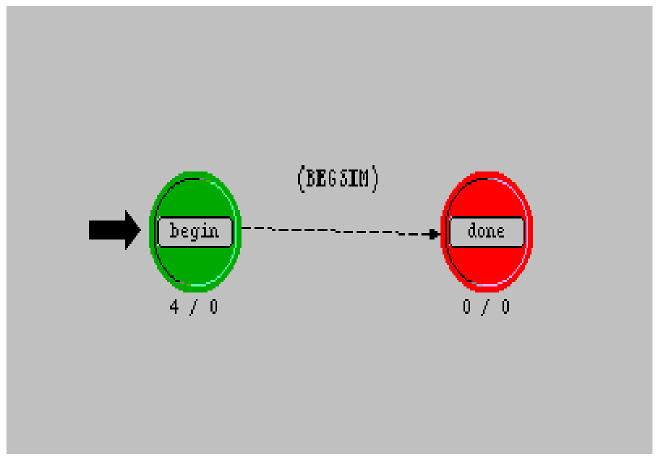


Fig.A5. “ak_epon_onu_manager_kubo”

A.3.3 - Process level and Proto-C level

Process models define the behavior of programmable modules (e.g., queue, processor). The tasks that these modules execute are called processes. A process is an instance of a process model and operates within one module. Because it has a set of instructions and maintains state memory, it is similar to an executing a software program.

More than one process can be placed in one module by using dynamic processes.

A root process is placed in the process model to manage the creation, invocation and destruction of new processes, called child, dynamically. A child process can become a parent and create other new processes.
Processes respond to interrupts, which indicate the occurance events of interest, such as the arrival of a message or the expiration of a timer. When a process is interrupted, it takes actions accordingly and then it stops, awaiting a new interrupt. It may also invoke another process; its execution is suspended until the invoked process finishes.
The process models are expressed in a language called Proto-C, which is specifically designed to support development of protocols and algorithms. Proto-C is based on a Finite State Machine (FSM), a library of high-level commands known as Kernel Procedures, and the general facilities of the C or C++ programming language. The FSM is composed of states that the process can enter and, for each state, the conditions that would cause the process to move to another state. The condition to change between states and the associated destination state is called a transition. For each transition there could be an action, a function written in Proto-C.
The modules programmed in this project are “olt” and “mac”, queue modules in the OLT and ONU node respectively.

A.3.3.1 - “olt” module

The “olt” module defines the implementation of OLT MPCP sublayer as a collection of the following processes (ignoring the ones involved with autodiscovery):

- Control Parser (CP): receives a frame from the lower layer and dispatches it to the upper layer (if it is a data frame) or to the Report Process (if it is a MPCP control frame or a ACK or a CONFIRMATION frame);
- Report Process (RP): parses the REPORT message and passes the relevant data to DBA_US (if a MPCP control frame is received) or forwards directly the frame to DBA_US (if an ACK or CONFIRMATION frame is received);

- DBA_US: implements the Dynamic Bandwidth Allocation (in our case, IPACT “gated”) for upstream traffic and passes the relevant data to the Gate Process (if a REPORT frame is received) or forwards the received frame to the DBA_DS (if an ACK or a CONFIRMATION frame is received) or forwards the REQUEST message to GP if it is received from DBA_DS;
- DBA_DS: implements the sleep mode technique, builds a REQUEST frame and forwards it to the DBA_US, informs the CM process about arrival frames from the upper layer thus it can put them in a queue, forwards a request to send downstream traffic to the MPCP-DU process;
- Gate Process (GP): upon the reception of a couple (start time, length), it builds a GATE frame and forwards it to the Control Multiplexer (if a REPORT frame is received) or forwards directly a REQUEST frame to the Control Multiplexer (if a REQUEST is generated);
- Control Multiplexer (CM): multiplexes the frames received from the upper layer and from the GP into the downstream direction, upon reception of a grant from MPCP-DU;
- Multi-Point Control Protocol-Data Unit (MPCP-DU): controls the activity of all the CMs present in the MPCP sublayer.

There is only one instance of DBA_US, DBA_DS and MPCP-DU per MPCP, while each instance of CP, RP, GP and CM is associated to one registered ONU, n of them if n is the number of registered ONUs.
The implementation of the MPCP sublayer follows the reference implementation, so one OPNET process is defined per each EPON process.
Since the number of the processes to be instantiated is known only at run time, the whole EPON stack is implemented by dynamic processes, where each OPNET process (covering the process which implements the Reconciliation Sublayer (RS) behavior) is a child process of a root process, which has the duty of instantiating all the others.
The root process, that is also the process model of the OLT node, is “ak_epon_olt_manager_kubo”.

A.3.3.2 - “mac” module

The queue module “mac” in the ONU node describes the ONU's architecture using dynamic processes similar to the OLT's module. The process model is “ak_epon_onu_manager_kubo”, the root process of our implementation, which manages the generation of the child processes in a way to emulate the ONU MPCP sublayer.

The process manager, at the beginning of the simulation, obtains the LLID from an attribute of the node and assigns it to each child process calling the pm_onu_kubo() function presented in “ak_epon_onu_all_kubo” external file.

The MPCP sublayer is defined as follows:

- Control Parser (CP): receives a frame from the lower layer and dispatches it to the upper layer (if it is a data frame) or to the Gate Process (if it is a MPCP control frame or a REQUEST frame);
- Gate Process (GP): provides the GATE reception and the GATE activation that control the time transmission, passing the relevant DATA to DBA_US and CM processes, or forwards the REQUEST frame to DBA_US if it is received;
- DBA_US: implements the Dynamic Bandwidth Allocation (in our case, IPACT “gated”) for upstream traffic and passes the relevant data to the Report Process (if a Gate frame is received) or forwards the frame received to the DBA_DS (if a REQUEST frame is received) or forwards the ACK message or the CONFIRMATION message to RP if they are received from DBA_DS;
- DBA_DS: implements the sleep mode technique, parses the REQUEST message, obtains the value of the sleeping time T, generates an ACK frame forwarding it to DBA_US and, if the sleeping time is not zero, generates a CONFIRMATION frame after the T value forwarding it to DBA_US;
- Report Process (RP): parses the message that it receives, if a GATE frame

arrived, upon the reception from DBA_US of the size of the traffic to be sent in the next polling cycle, builds a REPORT frame and forwards it to CM, or forwards an ACK or CONFIRMATION frame to CM if one of them is received from DBA_US;

- Control Multiplexer (CM): multiplexes the frames received from the upper layer and from the RP into the downstream direction, upon the start time of the transmission window allocated.

A.3.3.3 - Communication between processes

In this section it is explained how to coordinate communications between processes.

Analyzing the execution flow between the processes, a process should be able to:

- invoke a process associated to the same ONU (e.g., CP process associated to ONU i must invoke the RP process associated to i);
- pass to the invoked process a parameter (e.g., RP should pass to DBA_US the queue length reported) or a packet (e.g., GP should pass a frame to CM);
- know which ONU is associated to the invoked process, if any (e.g., DBA_US should know who's reporting to);
- invoke a process associated to a given ONU (e.g., DBA_US should pass information to the right GP process to create a Gate message).

To solve this issue in an homogeneous way, a C++ class called "process_kubo" is defined, which represents an OPNET process.

Process exposes some methods to send or receive a message (which can carry a frame and an argument) to a given process, to get its type and its associated LLID (NO IDX if there isn't one, like for DBA_US and MPCP-DU) or to associate an interrupt.

The "process_kubo" implements the message delivery as an interrupt scheduled on the receiver process, using an ICI to carry the body of the message. If a packet is to be delivered, the "ak_packet_ici_kubo" is used, or, otherwise, the

“ak_async_ici_kubo” is used. Since the message also includes the sender’s process, the receiver knows about the LLID associated to it.
To get the process reference of a given process, the interface “process_manager_kubo” was defined.
The “process_manager_kubo” duty consists on implementing all the processes and getting a reference of a process given its type and its associated LLID.
It installs itself in shared memory, so each process gets a reference to it by calling “op_pro_modmem_access()”. Each process also gets its associated LLID through “op_pro_parmem_access()”, so it is able to get its own process reference (which is required, since each process must send messages to other processes using its own process).
The implementation of “process_kubo” and “pm_olt_kubo” (which is the concrete implementation of “process_manager_kubo” used by OLT) can be found in the external source file “ak_epon_procman_kubo”.

A.3.3.4 - Association of streams to processes

All the MPCP instances are contained in a single module, which implies that it must be connected to all the sources, sinks and transmitter used by OLT.
This design needs to correctly associate stream indexes to processes that must send and receive packets from the bus (RS process) or processes that must forward packets to the upper layer (CP process) and processes that must receive packets from the upper layer and forward to the right ONU through RS (CM process).
For this, OPNET topology discovery functions are used to detect stream indexes associated to transmitter/receiver and to the i-th process, in a given direction. The informations are taken by the function “get_stream_kubo(dir, llid)”, which can be found in the external source code “ak_epon_topo_kubo”. A process can obtain the number of all the input and output streams, the kind of module connected to it (processor/queue or transmitter/receiver) by each stream and the index of the stream associated to the respective LLID.

In the following section each dynamic process implementation of the EPON OLT is described.

A.4 – Description of the processes

A.4.1 - OLT RS

The main objective of the RS process is to:

- dispatch an incoming frame to the CP associated with the ONU which has sent the packet;
- send a frame on the bus by putting in the preamble the LLID of the ONU associated with the index of the CM which has delivered the packet.

The event-response table is:

STATE	EVENT	ACTION	ARRIVING STATE
ready	Packet from ONU i	Dispatch to CP-i	ready
ready	Packet from CM i	Write LLID i in the preamble, and send	ready

		to the network	

Table A1. OLT RS event-response table

The process is implemented in “ak_epon_rs_kubo” (Fig. A6).

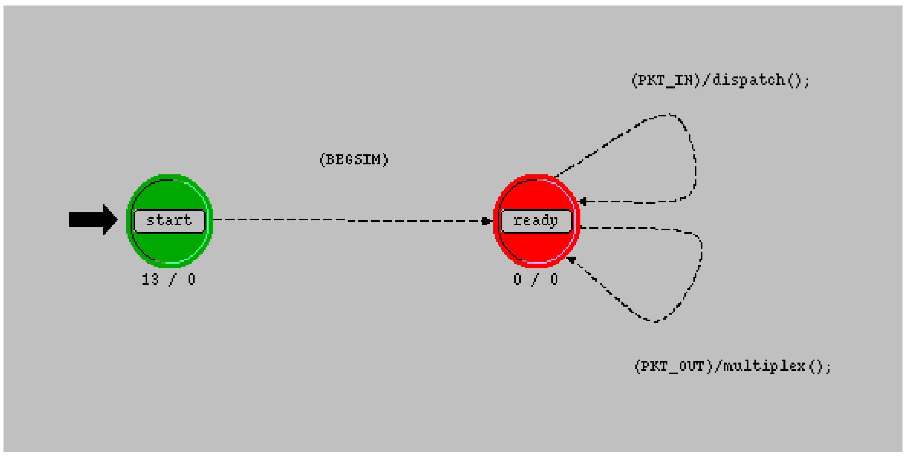


Fig.A6. “ak_epon_rs_kubo”

A.4.2 - OLT and ONU CP

The Control Parser is responsible for parsing the received frame and dispatching it to the Gate or the Report process or to the upper layer.

Since we assumed that the propagation delay between OLT and ONU is fixed, there is no need to implement timestamp-related functions, so in our model the CP of OLT and ONU is the same.

The event-response table is the following:

STATE	EVENT	ACTION	ARRIVING STATE
ready	Arrival of a GATE or a REQUEST	Parse the message and send to GP	ready
ready	Arrival of a REPORT or an	Parse the message and send	ready

	ACK or a CONFIRMATION	to RP	
ready	Arrival of a DATA packet	Pass the data packet to the upper layer	ready

Table A2. CP event-response table

The first event can occur only at the ONUs, the second one can occur only at the OLT and the third at both.

In our implementation, we have grouped the three events into a generic event "PKT ARRIVED", so the action dispatch_cp() first determines the kind of frame received and then continues with the parsing / delivering of the payload.

The process is implemented in "ak_epon_cp_kubo" (Fig. A7).

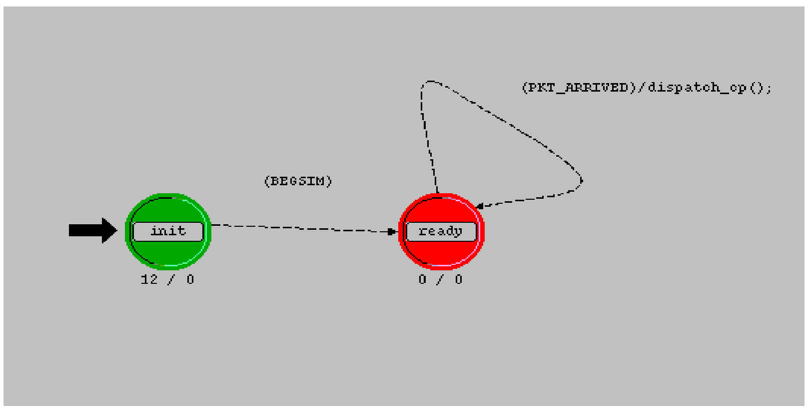


Fig.A7. "ak_epon_cp_kubo"

A.4.3 - OLT RP

The Report Process is in charge of parsing the Ethernet MAC control frame containing the REPORT and sending the report field to the DBA_US process or parsing the messages of the SPW technique forwarding them to the DBA_US process.

The event-response table is:

STATE	EVENT	ACTION	ARRIVING STATE
init	Simulation start	Generation of a Report fake and forward it to DBA_US	ready

ready	Arrival of a REPORT or an ACK or a CONFIRMATION from CP	Parse the message and forward it to DBA _US	ready

Table A3. OLT RP event-response table

In the initialization state a first “fake” report is generated from the associated ONU, in order to start the polling cycle.

The process is implemented in “ak_epon_olt_rp_kubo” (Fig. A8).

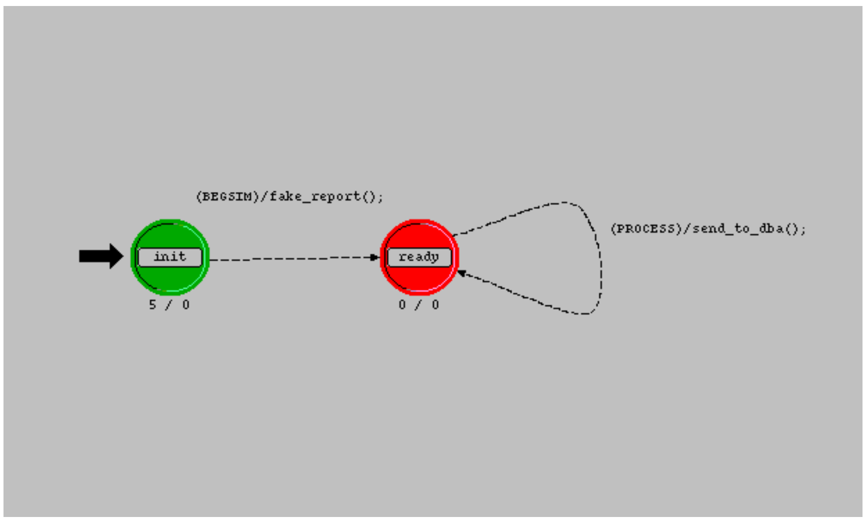


Fig.A8. “ak_epon_olt_rp_kubo”

A.4.4 - OLT DBA_US

The DBA_US process is in charge of allocating the bandwidth for upstream traffic using the information that each ONU sends to the OLT in a REPORT message.

The algorithm used in our case is IPACT with GATED mechanism, in which at every REPORT received from ONU i with a queue length n bits, the OLT sends a GATE to ONU i with length time(n + REPORTsize), where time(k) is the window length to transmit k bits, and a start time in which the ONU can start to send traffic.

The value of the constants defined by EPON (T_{on} , T_{off} , $T_{process}$, sync_length) where taken from [28].

Our implementation defines also a local statistic, “Polling cycle”, which reports the detected polling cycle length and has been defined mainly to perform model verification.

The event-response table is:

STATE	EVENT	ACTION	ARRIVING STATE
ready	Arrival of an ACK or CONFIRMATION from RP	Parse the message and forward it to DBA_DS	ready
ready	Arrival of a REQUEST from DBA_DS	Parse the message and forward it to GP	ready
ready	Arrival of a REPORT from RP	Allocate the bandwidth and generate the GATE packet	ready

Table A4. OLT DBA_US event-response table

The process implementation is given in “ak_epon_olt_dba_us_kubo” (Fig. A9).

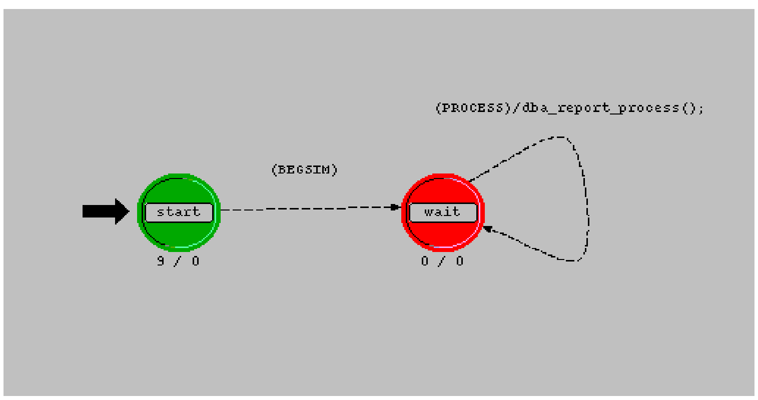


Fig.A9. “ak_epon_olt_dba_us_kubo”

A.4.5 - OLT DBA_DS

The energy saving technique SPW is implemented in the DBA_DS process. Every time a packet arrives, this process informs the CM process and at the same time calculates the average inter-arrival time by using an exponential smoothing calculation. Comparing the average inter-arrival time with a threshold, the OLT can predict the behavior of the downstream traffic. The SPW is activated if the average inter-arrival time is more than the threshold, setting T equal to T_s,

otherwise the technique is not activated, T is set to zero and the traffic is sent.
In this process a REQUEST message is created in two cases: the OLT receives a CONFIRMATION message, sends all the packets arrived during the previous T_s and at the end sends a REQUEST message, or when, in case of SPW not enabled, the transmission of all the packets is done and the comparison of the $i_{a,k}$ (the average inter-arrival time), is greater than the threshold.
The description of the DBA_DS process is detailed in the table:

STATE	**EVENT**	**CONDITION**	**ACTION**	**ARRIVING STATE**
(1) IDLE (2) WAIT_DATA	Arrival of a DATA packet	True	Inform the CM to queue the packet, update the *iak* and forward a request to the MPCP-DU to send traffic	(1) SEND_DATA (2) SEND_DATA
SEND_DATA	Arrival of a DATA packet	True	Inform the CM to queue the packet, update the *iak* and forwards a request to the MPCP-DU to send the new traffic if *iak* is lower than the threshold	SEND_DATA
SEND_DATA	Packets in queue transmitted	iak lower than the threshold	NO ACTION	WAIT_DATA
SEND_DATA	Packets in queue transmitted	iak upper than the threshold	Compare iak with the threshold, set T value, generate a REQUEST frame and forward it to DBA_US	WAIT_ACK
WAIT_ACK	Arrival of an ACK	ACK to a REQUEST (T=0)	Forwards a request to the MPCP-DU to send traffic	SEND_DATA
(1) WAIT_ACK (2) WAIT_ CONFIRMATION (3) SEND_OLD_DATA	Arrival of a DATA packet	True	Inform the CM to queue the packet, update the *iak*	(1) WAIT_ACK (2) WAIT_ CONFIRMATION (3) SEND_OLD_DATA
WAIT_ACK	Arrival of an ACK	ACK to a REQUEST (T=Ts)	NO ACTION	WAIT_ CONFIRMATION
WAIT_ CONFIRMATION	Arrival of a CONFIRMATION message	Queue of DATA packets not empty	Forwards a request to the MPCP-DU to send traffic	SEND_OLD_DATA
WAIT_ CONFIRMATION	Arrival of a CONFIRMATION	Queue of DATA packets empty	Compare iak with the threshold, set T value,	WAIT_ACK

	message		generate a REQUEST packet and forward it to DBA_US	
SEND_OLD_DATA	Packets in queue transmitted	True	Compare iak with the threshold, set T value, generate a REQUEST packet and forward it to DBA_US	WAIT_ACK

Table A5. OLT DBA_DS event-response table

The process implementation is given in “ak_epon_olt_dba_ds_kubo” (Fig. A10).

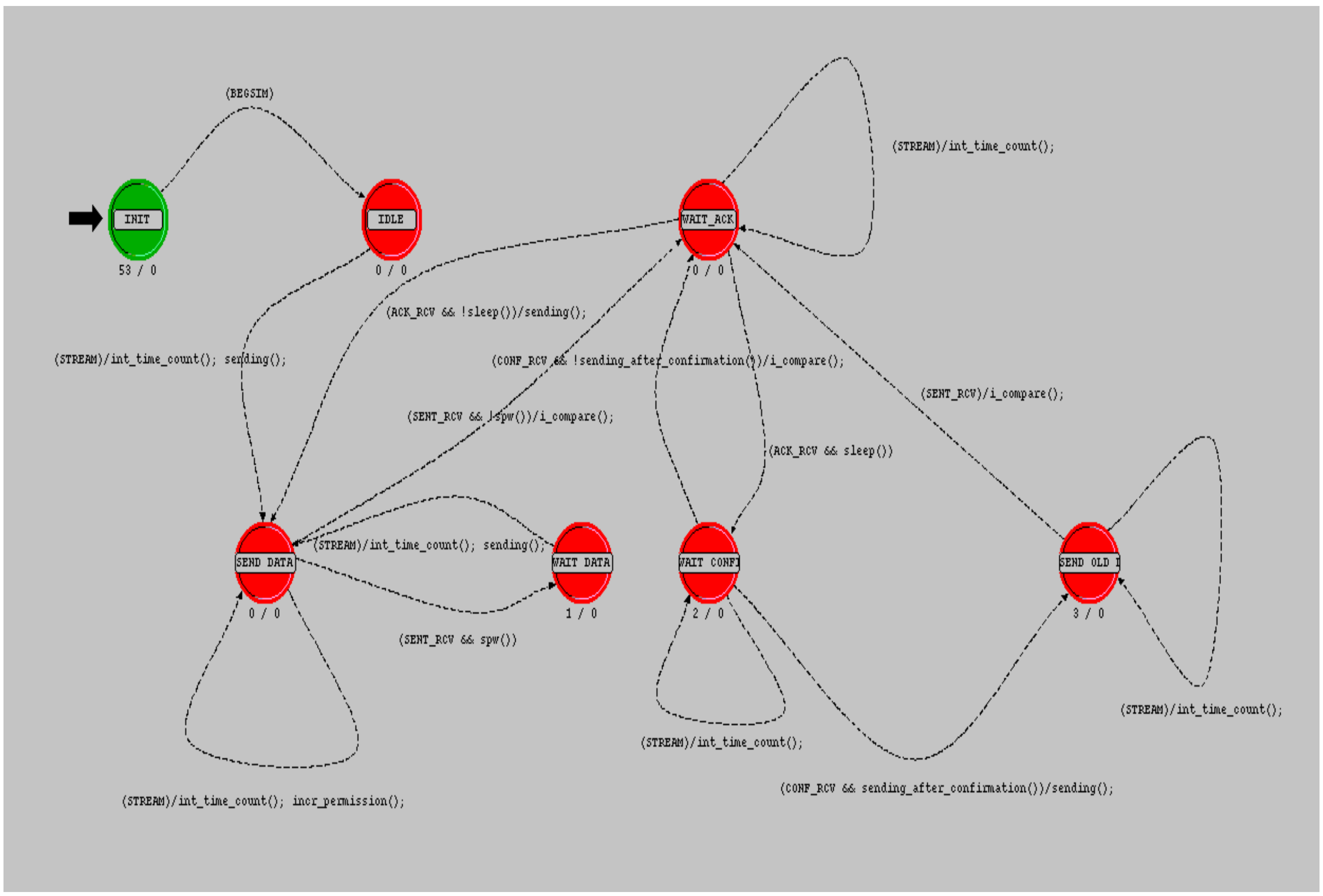


Fig.A10. “ak_epon_olt_dba_ds_kubo”

A.4.6 - OLT GP

The duty of the Gate Process is to receive a couple (start time, length) from DBA_US and to forward to CM the corresponding GATE frame.

Since the number of ONUs is fixed, the periodic transmission is not supported, which is a keep-alive mechanism.
The event-response table is:

STATE	EVENT	ACTION	ARRIVING STATE
ready	Message from DBA_US	build the corresponding GATE frame and forward it to CM (if a REPORT frame was received) or forward a REQUEST frame to CM if it is received	ready

Table A6. OLT GP event-response table

The implementation is given in “ak_epon_olt_gp_kubo” (Fig. A11).

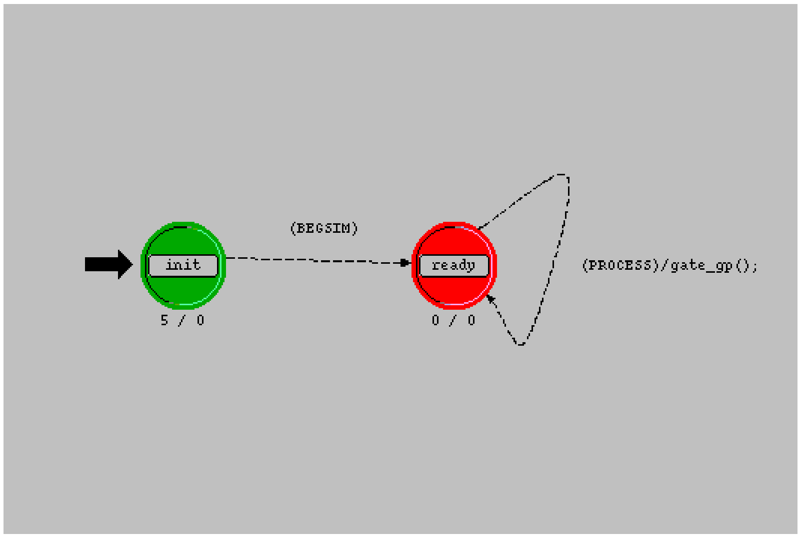


Fig.A11. “ak_epon_olt_gp_kubo”

A.4.7 - OLT CM

Control Multiplexing process multiplexes regular data, controls messages to a

specific ONU and encapsulates incoming frames into Ethernet frames. Different Cms are controlled MPCP-DU, in order to get a grant and send data.
The time stamping functions are not supported in this process because the transmission delays between OLT-ONU is known and fixed in our model.
In the CM process the arrival DATA packets generated in the OLT are queued in a priority queue, instead control frames are managed by using regular lists. The control frames get an high priority over data frames.
The event-response table is the following:

STATE	EVENT	CONDITION	ACTION	ARRIVING STATE
(1) ready (2) wait_grant_ds (3) send_pkt_ds	Arrival of a GATE message or a DATA packet	True	Queue the DATA packet, insert the control messages (GATE and REQUEST) in the specific list	(1) ready (2) wait_grant_ds (3) send_pkt_ds
ready	Arrival of a permission from the MPCP-DU to send traffic	True	Request a grant to MPCP-DU to send a GATE message or a DATA packet into the queue	wait_grant_ds
(1) wait_grant_ds (2) send_pkt_ds	Arrival of a permission from the MPCP-DU to send traffic	True	Request more grants to MPCP-DU to send the new DATA packets into the queue (*)	(1) wait_grant_ds (2) send_pkt_ds
wait_grant_ds	Arrival of a grant	True	Send the GATE message or the packet in priority queue	send_pkt_ds
send_pkt_ds	End of the packet transmission	More DATA packets in the queue requested from DBA_DS to send or a GATE message in the list	Release the grant to MPCP-DU and request a new grant to MPCP-DU to send a GATE or a DATA packet into the queue	wait_grant_ds
send_pkt_ds	End of the packet transmission	No more DATA packets in the queue requested from DBA_DS to send and no GATE message in the list	Release the grant to MPCP-DU and communicate the end of the transmission to DBA_DS	ready
ready	Arrival of a REQUEST message	True	Insert the message in the list and request a grant to MPCP- DU	wait_grant

(1) wait_grant (2) send_pkt	Arrival of a packet	True	Queue the DATA packet	(1) wait_grant (2) send_pkt
wait_grant	Arrival of a grant	True	Send the REQUEST message	send_pkt
send_pkt	End of the REQUEST transmission	More REQUEST messages in the list	Error	ready
send_pkt	End of the REQUEST transmission	No more REQUEST messages in the list	Release the grant to MPCP-DU	ready

Table A7. OLT CM event-response table

(*) The DBA_DS manages the packets to be sent by comparing the average inter-arrival time with the threshold. When the SPW is not enabled, the transmission starts with the n packets that are in the queue. The next packets that arrived during this transmission are sent if the average inter-arrival time is less than the threshold. In this case the DBA_DS requires the CM to transmit more packets.
The implementation is in “ak_epon_olt_cm_kubo” (Fig. A12).

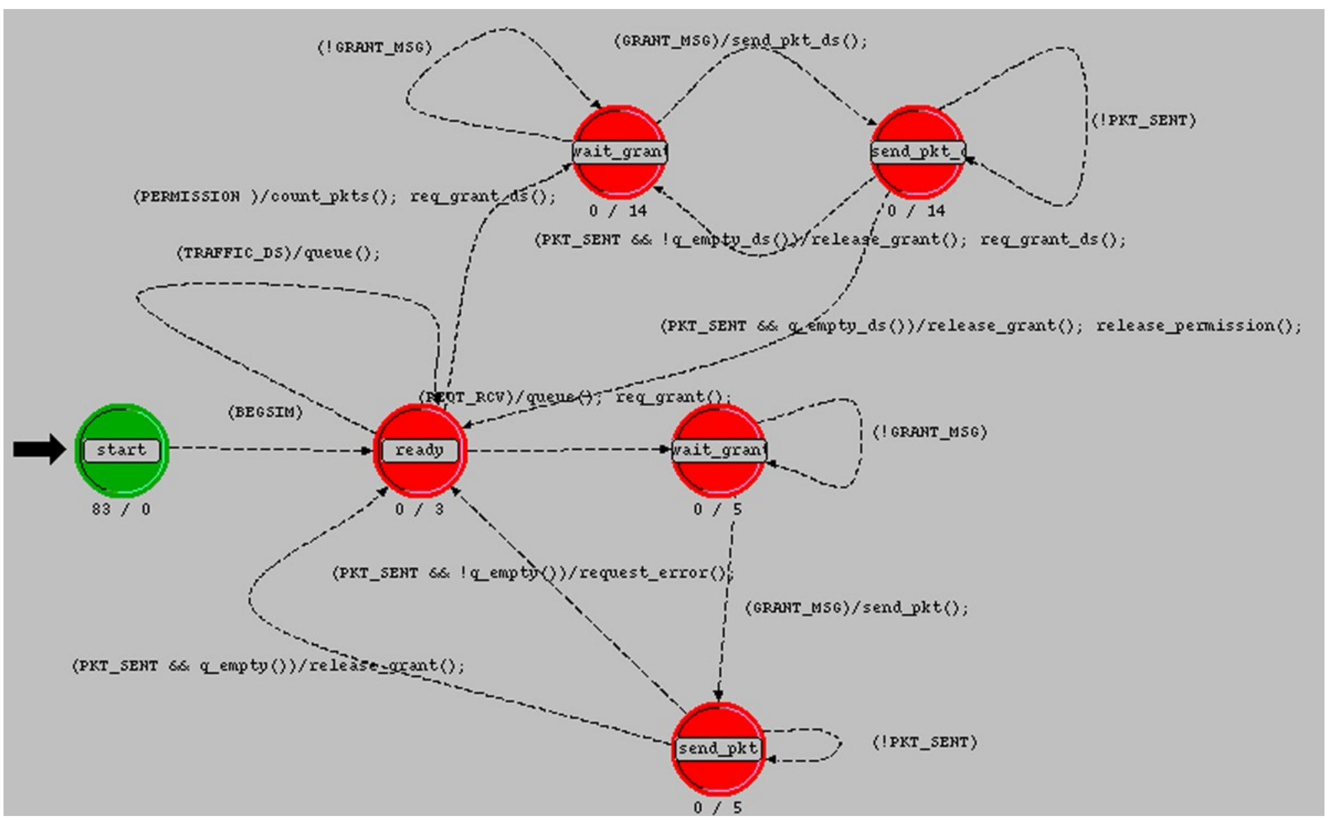


Fig.A12. “ak_epon_olt_cm_kubo”

A.4.8 - MPCP-DU

The MPCP-DU process manages the access to the downstream direction to all the CM instances present in the module.

We have implemented a mechanism where a CM which needs to send control frames, gets higher priority over CM which doesn't.

It is important to underline that this mechanism, coupled with the priority queue used in CM, guarantees the lowest delay in the control frames.

The event-response table is detailed:

STATE	EVENT	CONDITION	ACTION	ARRIVING STATE
wait	Grant request	True	Release a grant to requesting CM	busy
(1) wait (2) busy	Arrival a permission from DBA_DS to send traffic	True	Forward it to CM	(1) wait (2) busy
busy	Grant request	True	Queue the request	busy
busy	Grant release	Queue not empty	Release a grant to the CM at queue's head	busy
busy	Grant release	Queue empty	NO ACTION	wait

Table A8. MPCP-DU event-response table

The implementation can be found in "ak_epon_olt_du_kubo" (Fig. A13).

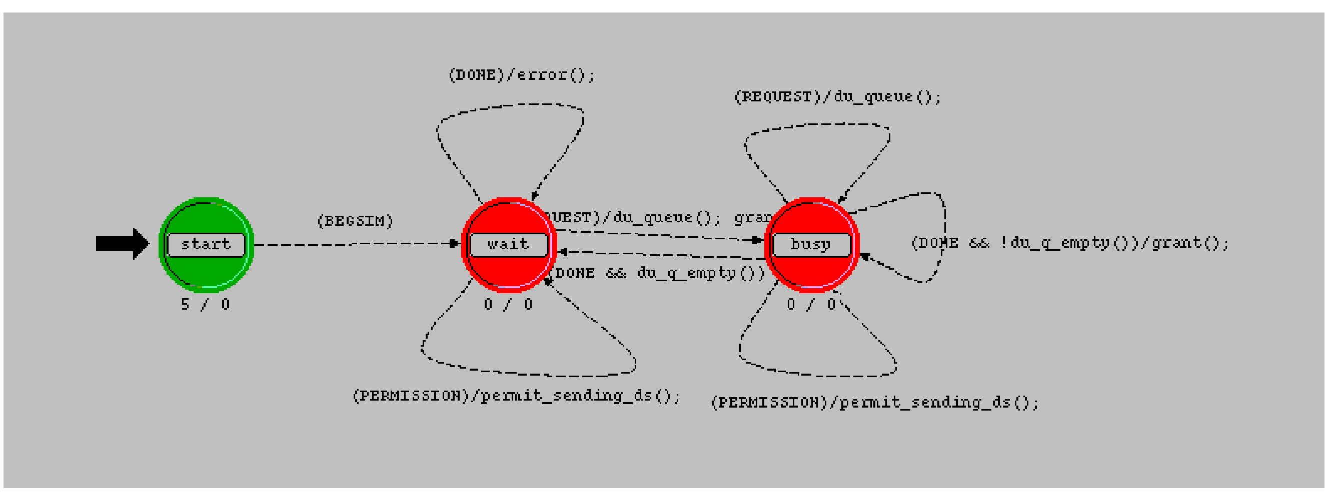


Fig.A13. “ak_epon_olt_du_kubo”

In the following section every dynamic process implementation of the EPON ONU is described.

A.4.9 - ONU RS

The duty of the RS process is to:

- parse an incoming frame and check the LLID in the preamble. If the LLID is the same of the ONU which is controlling it, the packet is accepted and forwarded to the CP, otherwise the packet is destroyed;
- send a frame on the bus by putting in the preamble the LLID of the ONU which is sending it;

The event-response table is:

STATE	EVENT	ACTION	ARRIVING STATE
dispatch	Arrival of a DATA packet from the bus	Parse the packet, check the LLID and forward it to CP or destroy it	dispatch
dispatch	Arrival of a packet from CM	Write the own LLID and send the packet in the bus	dispatch

Table A9. ONU RS event-response table

The process is implemented in “ak_epon_onu_rs_kubo” (Fig. A14).

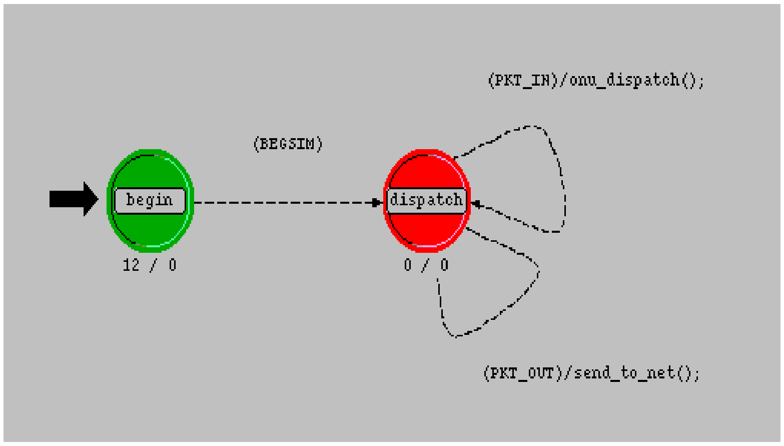


Fig.A14. “ak_epon_onu_rs_kubo”

A.4.10 - ONU GP

If a report has to be produced, the Gate Process would parse the GATE frame received from the downstream direction, obtain the couple of values <start_time,length> fixed in the bandwidth allocation in the OLT and pass the length value to CM at the time “start_time” and to the DBA_US agent in an infinitesimal time before “start_time”.

Upon the GATE reception, the delay d between the “start time” and the local time is calculated to determine the moment of forwarding the relevant data for CM and DBA_US.

If a REPORT has to be produced (always in the case of IPACT), the GP would pass the “length” value to the DBA _US at time $d - \varepsilon$ (with $\varepsilon > 0$), to ensure the generation of the REPORT before the “start time”.

In the implementation, a processing time, T_PROCESS, is considered before the forwarding and the local time is determined by getting the synchronization between the OLT and the ONU by the <onu_local_time_kubo(llid)> function in the “ak_epon_onu_all_kubo” file .

This is a consequence of the assumption that propagation delays are fixed and known and because the timestamping-related functions are not implemented in this project.

The event-response table is:

STATE	EVENT	ACTION	ARRIVING STATE
ready	Arrival of a GATE frame	Obtain the <start_time,length> values, pass the length value to DBA_US and CM at start_time or forward the REQUEST message if it is received	ready

Table A10. ONU GP event-response table

The process is implemented in “ak_epon_onu_gp” (Fig. A15).

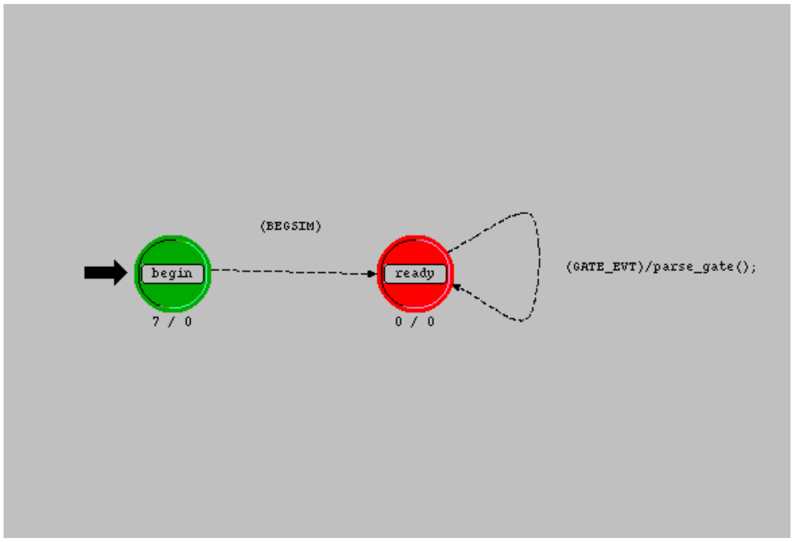


Fig.A15. "ak_epon_onu_gp"

A.4.11 - ONU DBA_US

The DBA_US process emulates the IPACT gated conduct in an ONU. In specific, it calculates how many packets from the queue, are to be reported.

The process receives the length value of the bandwidth allocated in terms of seconds, translates it in bits and makes a difference between the value obtained and the bits into the queue. This value is the size to be reported for the next polling cycle and is passed to the RP.

The periodic reports are not included since the number of ONUs is fixed over time and since IPACT continuously polls all the ONUs.

In this process also a local statistic is defined, "Packets reported", which is the size to be reported If the process receives messages different than a GATE message, it forwards the messages to DBA_DS or RP processes.

The event-response table is:

STATE	EVENT	ACTION	ARRIVING STATE
ready	Arrival of a GATE message	Calculation of the traffic to be reported and inform it to RP	ready
ready	Arrival of a REQUEST message	Forward the message to DBA_DS	ready
ready	Arrival of an ACK or a CONFIRMATION message	Forward the message to RP	ready

Table A11. ONU DBA_US event-response table

The process is implemented in “ak_epon_onu_dba_us_kubo” (Fig. A16).

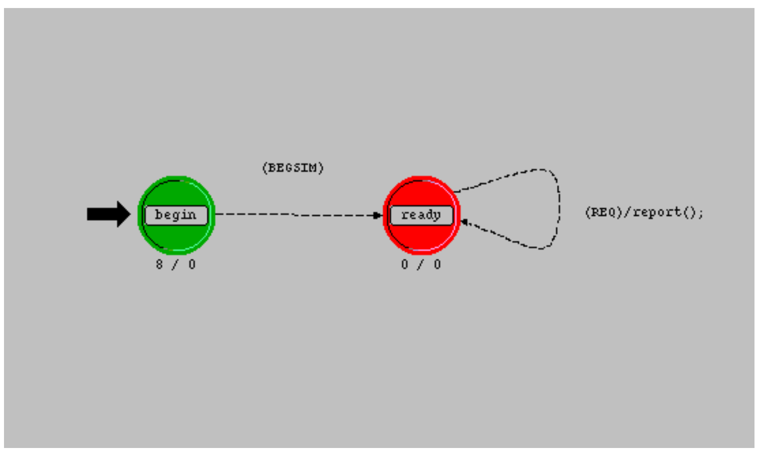


Fig.A16. “ak_epon_onu_dba_us_kubo”

A.4.12 - ONU DBA_DS

The functions of an ONU to implement the energy saving technique SPW are defined in the DBA_DS process. This process manages the consumption of energy of an ONU, it decides if the terminal has to switch in sleep mode or continue to use the whole power. When a REQUEST message is received, the process obtains the value of the sleeping time T and sends an ACK to the OLT. If T is different than zero, the process waits that time and a transition time that emulate the passage from the asleep state to the awake state. Afterwards the DBA_DS builds a CONFIRMATION frame to send to DBA_US.

Therefore the time in which the ONU is awake and asleep is exported in an external file "sleep_time_kubo_$seed.txt" useful for post-processing.

Details in the table:

STATE	EVENT	ACTION	ARRIVING STATE
WAIT_REQUEST	Arrival of a REQUEST message	Parse the message, obtain the T value, build an ACK frame, send it to DBA_US and schedule the generation of a CONFIRMATION frame	SLEEP_PROCESSING
SLEEP_PROCESSING	Sleeping time expired	Generate a CONFIRMATION frame and send it to DBA_US	WAIT_REQUEST

Table A12. ONU DBA_DS event-response table

The process is implemented in "ak_epon_onu_dba_ds_kubo" (Fig. A17).

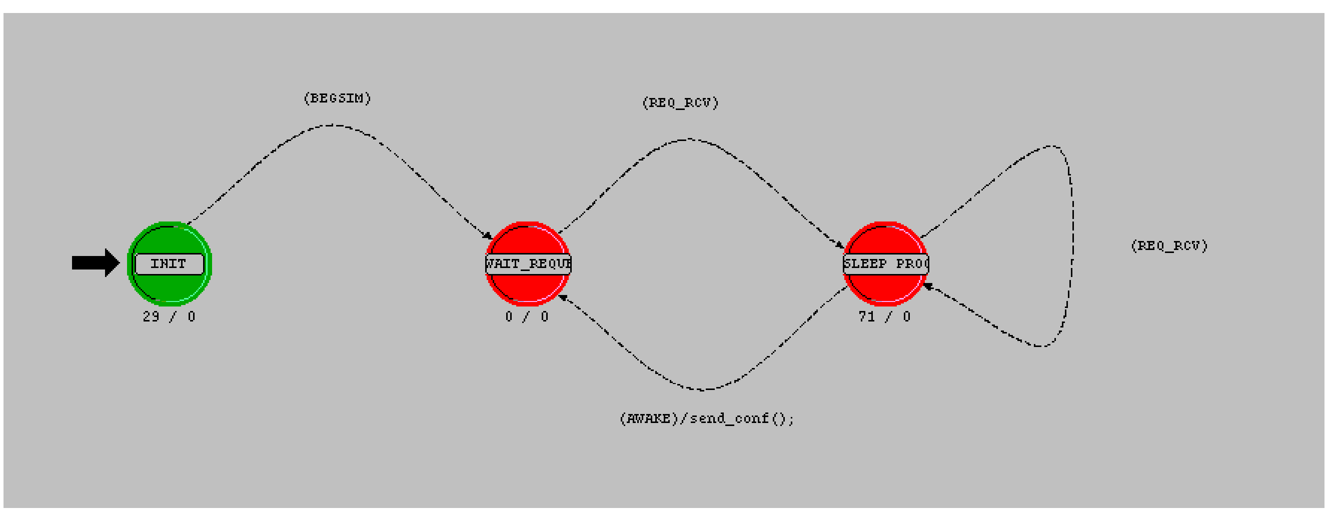


Fig.A17. "ak_epon_onu_dba_ds_kubo"

A.4.13 - ONU RP

The task of the RP process is to build a REPORT frame in which the size reported from the DBA_US is incapsulated. If an ACK or a CONFIRMATION frame is received, it is forwarded to CM.

The table describes the behaviour:

STATE	EVENT	ACTION	ARRIVING STATE
ready	Arrival of the traffic value to be reported	Build a REPORT frame and send it to CM	ready
ready	Arrival of an ACK or a CONFIRMATION frame	Forward it to CM	ready

Table A13. ONU RP event-response table

The process is implemented in “ak_epon_onu_rp_kubo” (Fig. A18).

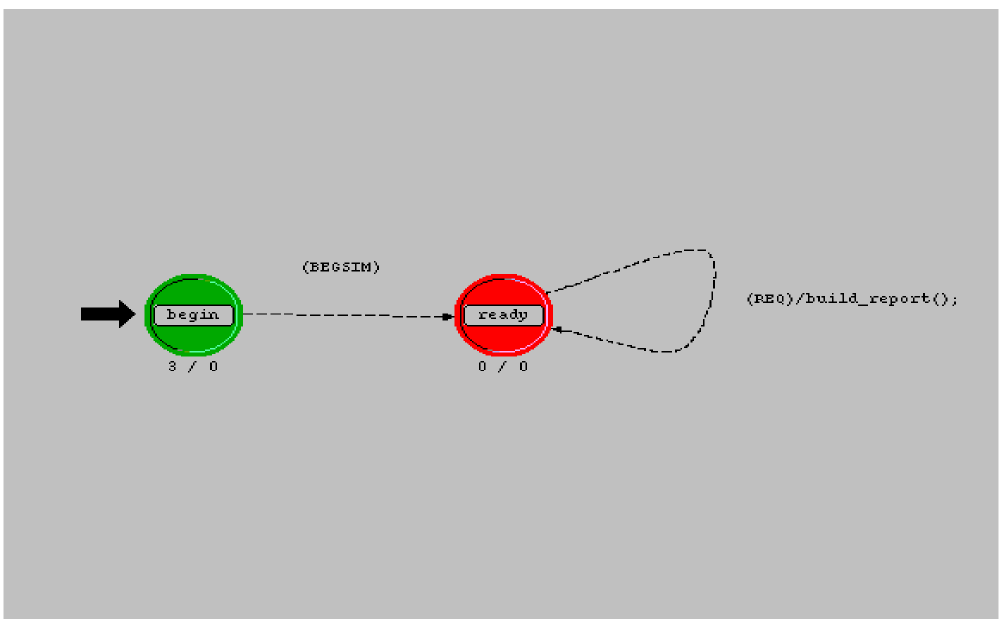


Fig.A18. “ak_epon_onu_rp_kubo”

A.4.14 - ONU CM

In the ONU CM process, the incoming data and control traffic upstream are multiplexed, similarly to OLT CM. The main difference is that CM is arbitrated by GP, which sends a message containing the grant length scheduled when transmission must start.
CM puts all incoming data packets in queue, properly framed, the REPORT message in a list and waits for the reception of the grant length.
If the CM receives a SPW control packet the, an ACK or a CONFIRMATION is sent immediately to the RS, without external controls.
Detailed description here

STATE	EVENT	ACTION	ARRIVING STATE
(1) wait (2) send (3) send kubo_pkts	Arrival of a DATA packet or a REPORT	Frame the packet or put the REPORT in the list	(1) wait (2) send (3) wait
wait	Arrival of a GATE message	Send a report and an amount of packets into the queue according to the grant length	send
send	Arrival of a GATE message	Discard the GATE message	send
send	Packets sent	NO ACTION	wait
(1) wait (2) send kubo_pkts	Arrival of an ACK or a CONFIRMATION packet	Frame the packet and send it to RS	(1) send kubo_pkts (2) send kubo_pkts

Table A14. ONU CM event-response table

The process is implemented in “ak_epon_onu_cm_kubo” (Fig. A19).

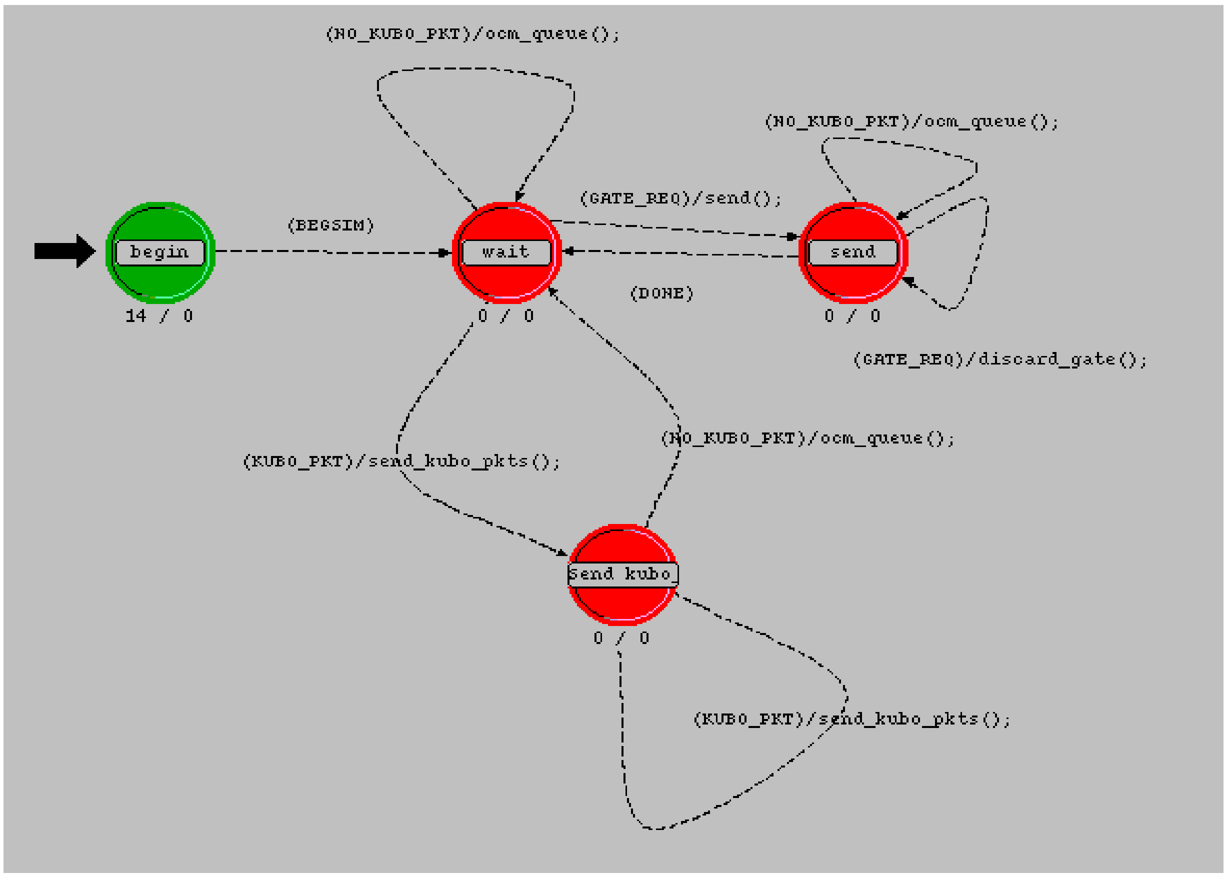


Fig.A19. "ak_epon_onu_cm_kubo"

A.5 – Shell script

In this section, it's explained a way to run the simulations in a simple way by launching a shell script on the terminal.

There are two ways to run simulations: launching the simulation executable program generated by op_mksim or using op_runsim from either the GUI (using DES menu commands) or from the command line. The basic command to run a simulation from the terminal is:

op_runsim -net_name Kubo-KUBO -exec_id 1 -opnet_port 52591

The command to execute a simulation is *op_runsim*, the name of the network must be specified by the *net_name* preference. The *exec_id* option identifies the execution of a simulation. It has to be the same for simultaneous simulations that belong to the same network model. The *opnet_port* specifies the Inter-process communication (IPC) port used by the opnet program to communicate with spawned simulations.

Some parameters can be set by specific static preferences, in particular the simulation time and the seed are considered in the simulations of this work making them modifiable for each execution by *-duration* and *-seed* options.

Some other parameters are created by the network developer in the project and it's not possible to use specific options to call them. The way to set these parameters is to call them by the name of the attribute promoted to the higher level specifying the name of the network and the objects of the previous hierarchical levels that includes it. For example, if the Packet Inter-arrival time attribute in a processor module is promoted, it's possible to set it adding this command to the basic one:

"-.OLT.source.Packet Interarrival Time" exponential $inttime*

The * is for the network that can be omitted because it is equal for all the attributes, *OLT* is the name of the node, *source* is the name of the module and *Packet Interarrival Time* is the name of the attribute that is set with the value that follows the command.

Also the global attributes are called on the command line in the same way without the specification of the objects. For example the "Ts" attribute is called:

"-Ts" $Ts

In the shell script, the command line to run the simulations and the execution of programming scripts for post-processing issues are set.

In particular the script "packet_delay.cpp" accepts as input the file "data_delay_collec.txt" in which the time of generating and sending of the packets with their own ID is included. In output, it releases a file with the average delay of the packets. The file in input is filled with the Opnet process OLT CM which states the time and ID of the packet every time one is generated and leaves the queue.

Another file generated in Opnet is "sleep_time_kubo.txt" by ONU DBA_DS in which the values of sleep time and active time are written every time they are updated during the simulation. The script "power_processing.cpp" takes it in input and releases in output a file with the average power value.

The bash script is as follows:

```
#!/bin/csh
```

```
set data = 11_11_2011
set time = 10
set change_Ts = 0
set ith = 0
set Ts = 0.01

cd /home/pippo/path
g++ packet_delay.cpp -o compiling_delay
mkdir delay
mkdir Risultati/simulazioni_$data
foreach seed (100 111 234 7 70 666 347 500 4 510)

mkdir seed_$seed
foreach inttime (0.8e-4 0.4e-4 0.2e-4 9.77e-6 4.88e-6 2.44e-6 1.3e-6)

op_runsim -net_name Kubo-KUBO -exec_id 1 -opnet_port 52591 -seed $seed -duration $time  "-*.OLT.source.Packet
Interarrival Time" exponential $inttime "-*.*.*.*.ith" $ith "-change_Ts" $change_Ts "-Ts" $Ts
mv /home/micchi/Documenti/TESI_SPECIALISTICA/project_KUBO/file_tesi/data_delay_collec_$seed.txt
/home/micchi/Documenti/TESI_SPECIALISTICA/project_KUBO/simulazioni_18-
Ottobre/seed_$seed/data_delay_collec_inttime_$inttime.txt

./compiling_delay seed_$seed/data_delay_collec_inttime_$inttime.txt /delay/Average_delay_diff_seed_inttime_$inttime.txt
packets_info_$seed-inttime-$inttime.txt

end
end

foreach seed (100 111 234 7 70 666 347 500 4 510)
mv seed_$seed Risultati/simulazioni_$data
end
```

REFERENCES

[1] C. Lam, Ed., “Passive Optical Networks: Principles and Practice”, Academic Press, 2007.

[2] G. Kramer, Ed., “Ethernet Passive Optical Network”, McGraw-Hill, 2005.

[3] J. Baliga et al., “Energy Consumption in Optical IP Networks”, IEEE/OSA JLT, vol. 27, no 13, 2009

[4] C. Lange et al., “Energy Consumption of Telecommunication Networks and Related Improvement Options,” Selected Topics in Quantum Electonics, IEEE Journal of, vol. 17, no. 2, pp. 285-295, March-April 2011.

[5] R. Tucker, “Green Optical Communications – Part II: Energy limitations in networks,” Selected Topics in Quantum Electronics, IEEE Journal of, vol. 17, no. 2, pp. 261-274, March-April 2011.

[6] M. Gupta and S. Singh, “Greening of the Internet,” in Proc. ACM SIGCOMM, 2003, pp. 19-26.

[7] “Code of Conduct on Energy Consumption of Broadband Equipment,” Version 4, European Commission Directorate General Joint Research Center (JRC) Institute for Energy, Renewable Energy Unit, Feb. 10, 2011.

[8] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi and J. Maestro, “IEEE 802.3az: The Road to Energy Efficient Ethernet,” Communications Magazine, IEEE, vol. 48, no. 11, pp. 50-56, November 2010.

[9] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasami, and D. Wetherall, “Reducing Network Energy Consumption via Sleeping and Rate-adaptation,“ in Proceedings of the 5th USENIX Symposium of Networked Systems Design and Implementation (NSDI), 2008, pp 1-14.

[10] J. D. Vereen, R. H. Diego, and B. A. Farber, “Optical Network Terminal with Low-power Sleep Logic that Substantially Extends the Life of the Battery after the AC main Power Supply has been Lost,” Tellams Petaluma, Inc., US007287175B2, October 2007.

[11] S.-H. Yen, S.-W. Wong,S. Das, N. Cheng, J. Cho, S. Yamashita, O. Solgaard, and L. Kazovsky, “Photonic Components for Future Fiber Access Networks,” Selected Areas in Communications, IEEE Journal on, vol. 28, no. 6, pp. 928-935, aug. 2010.

[12] L. Valcarenghi, M. Presi, G. Contestabile, P. Castoldi and E. Ciaramella, “Impact of Modulation Formats on ONU Energy Saving,” in Optical Communication (ECOC), 2010 36th European Conference and Exhibition on, Sept. 2010, pp. 1-3.

[13] W. Freude, M. Roeger, G. Boettger, M. Dreschmann, M. Huebner, C. Klamouris, A. Bett, J. Becker and J. Leuthold, “Optically Powered Fiber Networks,” Optics Express, vol. 16, no. 26, Aug. 2008.

[14] “GPON power conservation,” ITU-T G-series Recommendations - Supplement45 (G.sup45), 2009.

[15] E. Trojer and P. Eriksson, "Power Saving Modes for GPON and VDSL2," in *in Proc. 13th Euro. Conf. Netw. & Optical Commun. (NOC)*, Austria, Jun. 30-Jul. 3 2008.

[16] J. Mandin and L. Khermosh, "Power Saving in IEEE 802-style Networks," US Patent Application, May 13 2010, uS201000118753A1.

[17] http://www.pmc-sierra.com/products/details/pas7401.

[18] J. Zhang and N. Ansari, "Toward Energy-efficient 1G-EPON and 10GEPON with Sleep-aware MAC Control and Scheduling," *Communications Magazine, IEEE*, vol. 49, no. 2, pp. s33 –s38, february 2011.

[19] B. Skubic and D. Hood, "Evaluation of ONU Power Saving Modes for Gigabit-capable Passive Optical Networks," *Network, IEEE*, vol. 25, no. 2, pp. 20–24, March-April 2011.

[20] R. Kubo, J.-i. Kani, H. Ujikawa, T. Sakamoto, Y. Fujimoto, N. Yoshimoto, and H. Hadama, "Study and Demonstration of Sleep and Adaptive Link Rate Control Mechanisms for Energy Efficient 10G-EPON," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 2, no. 9, pp. 716 –729, september 2010.

[21] 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 *GLOBECOM Workshops, 2009 IEEE*, 2009, pp. 1–6.

[22] R. Kubo *et al.*, "Adaptive Power Saving Mechanism for 10 Gigabit Class PON Systems," IEICE E93-B, 280-287 (2010).

[23] M. Fiammengo *et. al.,*"Experimental Evaluation of Cyclic Sleep with Adaptable Sleep Period Length for PON", ECOC 2011, We.8.C.3.

[24] ITU-T Recommendation G.1010: "End-user Multimedia QoS Categories", URL: http://www.itu-t.org.

[25] ITU-T Recommendation Y.1541: "Network Performance Objectives for IP-based Services", URL: http://www.itu-t.org.

[26] Hideaki Takagi, "Analyisis of Polling Systems", The MIT Press, 1986, pp. 105-117.

[27] A G. Bolch *et al.,* "Queueing Networks and Markov Chains", Wiley, May 2006, pp. 148-151.

[28] http://www.ieee802.org/3/av/public/2007_11/3av_0711_suzuki_1.pdf.