Contributions towards softwarization and energy saving in passive optical networks

Khalili Hamzeh

CONTRIBUTIONS TOWARDS SOFTWARIZATION AND ENERGY SAVING IN PASSIVE OPTICAL NETWORKS

By

Hamzeh Khalili

A thesis submitted in partial fulfilment of the requirements for the

degree of Doctor of Philosophy in

NETWORK ENGINEERING

by the

UNIVERSITAT POLITECNICA DE CATALUNYA

Advisors:

Dr. David Rincon Rivera

Prof. Sebastia Sallent Ribes

June 2018

Dedicated to my wonderful parents

Don't wait until everything is just right. It will never be perfect. There will always be challenges, obstacles and less than perfect conditions. So what. Get started now. With each step you take, you will grow stronger and stronger, more and more skilled, more and more self-confident and more and more successful.

– Mark Victor Hansen

ABSTRACT

This thesis is a result of contributions to optimize and improve the network management system and power consumption in Passive Optical Networks (PON). Passive Optical Network elements such as Optical Line Terminal (OLT) and Optical Network Units (ONUs) are currently managed by inflexible legacy network management systems. Software-Defined Networking (SDN) is a new networking paradigm that improves the operation and management of networks by decoupling control plane from data plane.

In this thesis, we propose a novel architecture, based on the SDN concept, for Ethernet Passive Optical Networks (EPON) that includes the Service Interoperability standard (SIEPON). In our proposed architecture, the OLT is partially virtualized and some of its functionalities are allocated to the core network management system, while the OLT itself is replaced by an OpenFlow switch. A new MultiPoint MAC Control (MPMC) sublayer extension based on the OpenFlow protocol is presented. The OpenFlow switch is extended with synchronous ports to retain the time-critical nature of the EPON network. Our simulation based results demonstrate the effectiveness of the new architecture, while retraining a similar (or improved) performance in terms of delay and throughput when compared to legacy PONs.

In addition, we introduce an energy-efficient Dynamic Bandwidth Allocation (DBA) algorithm for both the upstream and downstream channels of EPON. The proposed algorithm analyzes the queue status of the ONUs in order to turn ONUs to doze/sleep mode when there is no upstream/downstream traffic in the network, respectively.

ACKNOWLEDGEMENTS

"Education is the most powerful weapon, which you can use to change the work." – Nelson Mandela

First and foremost, I would like to express my gratitude to all people who supported me throughout the years of my PhD studies and beyond. I would like to express my deepest appreciation to my PhD supervisors Prof. Sebastià Sallent and Dr. David Rincón for their efficient supervision, guidance, constant help and support throughout my PhD. They were always ready and available to guide me with their deep knowledge and board experience. They showed me the path of a better life. It is my fortune and honor to have had advisers like them. I would also like to extend my gratitude to Dr. Jòse Ramòn Piney for his valuable technical suggestions, feedback and his continued assistance throughout my candidature.

I would like to thank all my friends and colleagues in the UPC and Barcelona, for their friendship and support during my PhD research. They made my time as a researcher more enjoyable and helped me during my day-to-day research student life over the past few years.

Finally, a special thanks to my parents and sister, who have dedicated their whole life to make my life a better one. Their trust in my abilities, freedom given to me to pursue my dreams, and their unconditional love has made me who I am today. Words cannot express how grateful I am to my father and my mother for all of the sacrifices that you have made on my life. I could have never achieved my goal without their encouragement, understanding, supports, and true love.

CONTENTS

List of Tables

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

1G-EPON	1 Gbit/s EPON
10G-EPON	10 Gbit/s EPON
APON	ATM PON
ATM	Asynchronous Transfer Mode
BNG	Broadband Network Gateway
BPON	Broadband PON
CAPEX	Capital Expenditure
CBR	Constant Bit Rate
CDF	Cumulative Distribution Function
CORD	Central Office Re-architectured as a Datacenter
DBA	Dynamic Bandwidth Allocation
DDSPON	Dynamic Distributed Scheduling for EPON
DiffServ	Differentiated Services
DSL	Digital Subscriber Line
DTE	Data Terminal Equipment
EA	Ethernet Aggregation
EPON	Ethernet Passive Optical Network
ESP	EPON Service Path
FTTH	Fiber to the Home
FSAN	Full Service Access Network
GEM	GPON Encapsulation Method
G-EPON	Gigabit Ethernet Passive Optical Network
GPON	Gigabit PON
IEEE	Institute of Electrical and Electronics Engineers
IPACT	Interleaved Polling with Adaptive Cycle Time

IPTV	Internet Protocol Television
ITU	International Telecommunication Union
JIT	Just-In-Time
LLID	Logical Link Identifier
MAC	Medium Access Control
MDI	Medium-Dependent Interface
MPCP	Multi-Point Control Protocol
MPMC	MultiPoint MAC Control
MPLS	Multiprotocol Label Switching
MPTC	MultiPoint Transmission Control
NAT	Network Address Translation
NFV	Network Functions Virtualization
NG-ONUs	Next Generation ONUs
NNI	Network-Network Interface
OAM	Operation, Administration and Maintenance
OF	OpenFlow
ONOS	Open Network Operating System
ONU	Optical Network Unit
OLT	Optical Line Terminal
OLT_MDI	OLT Medium Dependent Interface
OPEX	Operational Expenditure
OSI	Open System Interconnection
P2MP	Point-to-Multi-Point
P2P	Point-to-Point
PBB	Provider Backbone Bridge
PCE	Path Computation Element
PCS	Physical Coding Sublayer

PDU	Protocol Data Unit
PMA	Physical Medium Attachment
PMD	Physical Medium Dependent
PON	Passive Optical Network
PHY	Physical
QAP	Queue Admission Policies
QoS	Quality of Service
RAN	Radio Access Network
REGISTER_ACK	Register Acknowledgment
REGISTER_REQ	Register Request
RRU	Remote Radio Unit
RS	Reconciliation Sublayer
RTT	Round Trip Time
SA-FS	Service-aware Flow Scheduling
SDN	Software Defined Networking
SDOAN	Software-defined Optical Access Network
SIEPON	Service Interoperability in Ethernet Passive Optical Networks
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
TDM	Time Division Multiplexing
UNI	User-Network Interface
VCPE	Virtual-CPE
VCSEL	Vertical-cavity Surface-emitting
VLAN	Virtual LAN
VOIP	Voice over Internet Protocol

vOLT	Virtual OLT
vRouter	Virtual Router
vSG	Virtual Subscriber Gateway
WDM	Wavelength Division Multiplexing
XaaS	Everything-as-a-Service

1 INTRODUCTION

This chapter has the following organization. The motivation for the PhD is presented in section 1.1. Section 1.2 enumerates the results and contributions of the research work presented in this document. Finally, Section 1.3 describes the structure of the document.

1.1 Motivation

In recent decades, the demand for high data rates in communication networks has increased rapidly. This increase in demand led to the development of reliable and high-speed data transmission in the broadband access networks. Optical fiber-based technologies, such as PON (Passive Optical Network) provide the only solution for existing and future access network requirements. PONs can supply high bandwidth demands and a wide range of services and applications to the users. One of its variants, EPON (Ethernet PON) combines the high bandwidth of optical networks with the well-known Ethernet architecture and frame format.

Nowadays, Software Defined Networking (SDN) is an emerging paradigm that has been successfully adopted by the networking community. It defines three architectural principles: 1) separates the control plane from the data plane in the SDN switches; 2) a logical centralization of the control plane in the so-called controller; and 3) programmability of network functions. The idea is that networks can be made easier to manage (control and monitor, to name a few possibilities) with a centralized management. OpenFlow is currently the de-facto standard interface between the SDN controller and the network devices.

Currently, network management in PON networks is not always automated nor normalized. One goal of the researchers in optical networking is to improve the programmability, efficiency, and global optimization of network operations, in order to minimize both Capital Expenditure (CAPEX) and Operational Expenses (OPEX) by reducing the complexity of devices and its operation. Therefore, it makes sense to use an SDN approach in order to manage the passive optical network functionalities and migrating must of the upper layer functions to the SDN controller. Many approaches have already addressed the topic of applying the SDN architecture in PON networks; however, the focus was usually on facilitating the deployment of SDN-based services and so service interoperability remains unexplored in detail. The main challenge towards this goal is how to make compatible the synchronous nature of the EPON media access control protocols with the asynchronous architecture of SDN, and in particular, OpenFlow.

The main goal of our research has been be to create a new EPON architecture based on the SDN paradigm. Optical access networks present several characteristics (tree-like topology, distributed shared access to the upstream channel, centralization of the intelligence of the network in the OLT device) that make them appealing for the virtualization of some of their functionalities. Such architecture has several advantages from the network management point of view. The control plane of the OLTs of a large operator can be (partially) virtualized and centrally controlled via OpenFlow/SDN, thus opening the way to coordinated backbone/access network optimization, dynamically adjusting the configuration to the traffic pattern, end-to-end flow prioritization and QoS enforcement, to name a few possibilities.

Nowadays, many researchers are working simultaneously to develop power saving techniques and improves energy efficiency in the PON networks, and since the contribution of access networks to the total energy consumption of global networks is large, energy efficiency has become an increasingly important requirement in designing access networks. Therefore, energy-saving approaches are being investigated to provide high performance, which consumes less energy.

Several technique have been proposed to increase energy efficiency in PON networks. Such techniques are related to OLT or ONU and regarded as either hardware or software approaches. The hardware approach introduces new physical optical architecture element, while software approach requires extension of resource management algorithms to include energy awareness. Many approaches already have been discussed about centralized DBA but the advantage of power saving in a distributed DBA remains untouched.

Our second goal has been to introduce a power-saving algorithm for a distributed DBA (such as DDSPON, Dynamic Distributed Scheduling for EPON) to improve energy efficiency in EPON networks. The presence of a power-saving mechanism should not affect the user's quality of experience. The quality of service must remain unchanged when switching from active mode to the sleep/doze mode and vice versa. The introduction of sleep and doze mode operations in the DBA algorithm can save a lot of energy for the PON networks without affecting QoS parameters (such as average delay).

1.2 Thesis Contributions

This thesis aims to provide an architectural contribution in Ethernet-based PON (EPON) network by designing a new SDN-based EPON architecture to separate and virtualize the control and management functions of the OLT. In addition, we provided a novel scheme for the Multipoint MAC Control (MPMC) sublayer of the OLT. Finally, we designed a virtualized internal OpenFlow switch in the OLT to emulate the forwarding functions of the MPMC sublayer, which is able to keep the synchronous nature of the optical network. These results have been published in a conference [1], a journal [2], and a book chapter [3].

During this research, we have involved mainly with manageability and synchronization issues in EPON networks, but we have also addressed the energy-saving issues in WDM/TDM-PON. We presented and evaluated a new DBA algorithm able to include sleep/doze mode operations in EPON networks. The results have been published in [4].

The thesis is based on a research methodology that comprises simulations, analysis and dissemination of results. To evaluate the proposed solutions, a simulation network model has been developed. The simulation environment is based on the OPNET Modeler tool [5] and the code developed during this thesis is another of our contributions.

1.3 Thesis Organization

This thesis is organized in five chapters and an appendix. The background materials of the fields related to our research are provided in Chapter 1 and Chapter 2. Our contributions are presented in Chapter 3 and Chapter 4. The appendix provides details about the network simulator. Specifically, the contents of each chapter are:

- Chapter 1 is the current introduction.
- Chapter 2 provides an overview on optical access networks and particularly on EPON networks and their architecture. We also review resource management techniques (such as bandwidth allocation, QoS, and energy-saving techniques). Next, we consider the Service Interoperability in Ethernet Passive Optical Networks (SIEPON) standard and its architectural model. Finally, we discuss the SDN paradigm and recent works on SDN-based optical networking.
- Chapter 3 defines our novel SDN-based SIEPON architecture and a new design for the MPMC sublayer based on an OpenFlow switch. We also introduce the new SDN-OLT functionalities, as well as implementation and operational details. The last section of this chapter is devoted to the description of the results obtained from a reference implementation based on simulation, and the evaluation of the proposed architecture.
- Chapter 4 presents an algorithm-based approach for saving energy in EPON networks. In this chapter, we evaluate sleep/doze operations embedded in a DBA algorithm to reduce energy consumption in the ONUs. Finally, simulation results are also presented to evaluate the performance of the algorithm.
- Chapter 5 summarizes the main conclusions from this PhD and points out future work directions.
- Appendix A describes the SDN-EPON network model, and the SDN-OLT and SDN-EPON controller node models developed in OPNET Modeler.

2 OPTICAL ACCESS NETWORKS

2.1 Introduction

The development of bandwidth-intensive applications has resulted in a rapid growth of the Internet during last decade. The growth of demand for high bandwidth and communication services cannot be meet with copper-based, conventional wired broadband access networks (Digital Subscriber Line (DSL)), due to the bandwidth limitations of the copper medium. To overcome this bottleneck and deliver high-speed content with more bandwidth to the customer, copper is being replaced with optical access networks. Optical networks are the most efficient solution to provide broadband access.

Nowadays, optical networks are not only used as access networks for fixed wire line customers, but are also used as backhaul for mobile networks, and even in the fronthaul (for example, in the cloud RAN architecture). Optical networks are more energy-efficient than its copper-based predecessor as a result of low transmission losses, and high bandwidth. Due to their ability to deliver high bandwidth in a cost effective manner, a large number of them have already been deployed globally.

This chapter presents the background of the research topics treated in this dissertation. We review the PON (specifically EPON) technology, and the main functionalities and constraints of this architecture. Then, we discuss the scenario and specifications provided by the Service Interoperability in Ethernet Passive Optical Network (SIEPON) standard for the EPON network. Next, we study the state of the art of Software Defined Networking (SDN) and OpenFlow regarding the integration of SDN with optical networks, and specifically with access networks.

The chapter is organized as follows: Section 2 introduces the PON networks, EPON networks, and its architecture. Section 3 describes the SIEPON standard and its architecture model. Section 4 discusses the SDN paradigm, the OpenFlow protocol and the most recent works on optical network based on the SDN paradigm. Section 5 summarizes the chapter.

2.2 Passive Optical Networks

A Passive Optical Network (PON) is a fiber access network that only uses fiber and passive components such as splitters and combiners, rather than active components such as repeaters and amplifiers. The PON splitter / combiner splits a beam of light into several bundles and distributes it to several optical fibers in one direction (downstream), and combines light signals from various optical fibers to a single optical fiber output in the other direction (upstream), respectively. A PON is typically capable to support and delivering integrated voice, data and video services at distance beyond 20 km. Passive optical networks are also known sometimes as Fiber to the Home (FTTH) networks.

Over the years, various PON standards have been developed. The first PONs, named Full Service Access Network (FSAN), were introduced in late 80s to overcome the last mile problem. In the late 90s, the International Telecommunication Union (ITU) [6] created the ATM PON (APON) standard as data link protocol, which used the Asynchronous Transfer Mode (ATM) for long-haul packet transmission. When ATM started its decline, a newer version was created, called Broadband PON (BPON) (ITU-T G.983). BPON provided 622 Mbits/s transmission speed in downstream and 155 Mbits/s in the upstream channel. The Gigabit PON (GPON) standard (ITU-T G.984) was released shortly after the BPON standard. GPON provides further bandwidth improvement due to new techniques allowing for dynamic bandwidth allocation. It delivers 2.488 Gbits/s downstream and 1.244 Gbits/s upstream. GPON has been widely deployed its and deployment continues as of today. In 2010, ITU introduced a higher data rate X-GPON (ITU G.987) [7] as a next generation PON standard with support for 10 Gbits/s downstream and 1 Gbits/s upstream. X-GPON uses wavelengths in the bands 1575-1580 nm in downstream direction, and the 1260-1280 nm band in the upstream direction.

Meanwhile, the IEEE standardized the Ethernet PON (EPON) based on the IEEE Ethernet standard 802.3 [8]. The transmission speed is 1.25Gbits/s in both downstream and upstream directions with a 20 km network span. Later in 2006, the 10 Gigabit Ethernet PON (10G-EPON) [9] was presented. The 10G-EPON supports two network configurations: symmetric and asymmetric EPON. The symmetric case operates at 10 Gbits/s in both downstream and upstream directions, while the asymmetric case supports 10 Gbits/s in the downstream direction and 1 Gbits/s in the upstream direction [10]. In the downstream direction, EPON and 10G-EPON use the 1480-1500 nm and 1575-1580 nm bands, respectively, while in the upstream direction use the 1260-1360 nm and 1260-1280 nm bands, respectively.

2.3 Ethernet Passive Optical Networks

The concept of Ethernet Passive Optical Network (EPON) was introduced by IEEE Std. 802.3ah in 2004. IEEE Std. 802.3ah proposed EPONs to expand the application of Ethernet to include subscriber access networks, which can provide a significant increase in performance, while minimizing equipment, operation and maintenance costs. EPON follows the original architecture of PON, where the Data Terminal Equipment (DTE) is connected to the tree. It keeps all the advantages of a Passive Optical Network physical layer (high data rates, cheap passive optical components) in the signal distribution network, while using easy-to-integrate, well-known Ethernet technology as the data link layer. The active components in EPONs are 1) the OLT, located at the service provider's premises, at the top of the topology tree, and 2) the ONUs, located at the customer's premises (FTTH) or at the curb (FTTC) [11], as illustrated in Figure 2-1. The signals transmitted by the OLT pass through a passive splitter in order to reach the ONU and vice versa.

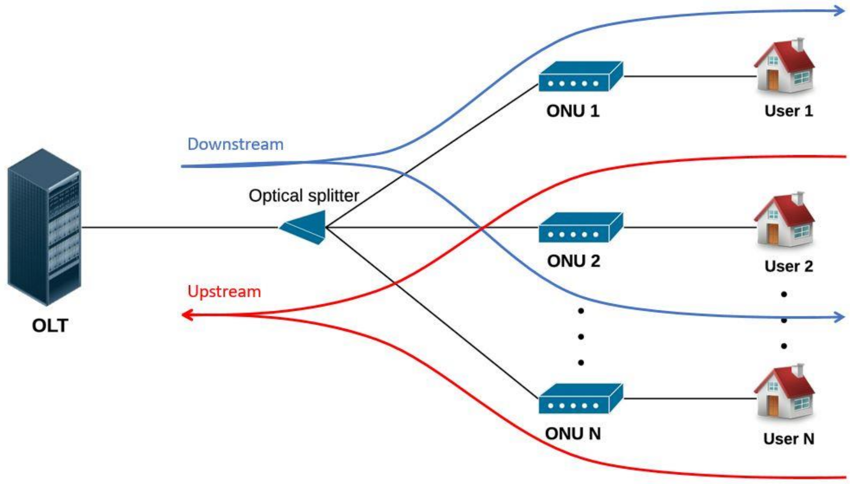


Figure 2-1 PON topology.

IEEE Std. 802.ah focuses on two layers of the OSI model, namely the physical layer (PHY) and data link layer [8]. Figure 2-2 shows the relationship between the Open System Interconnection (OSI) reference model and the EPON elements (OLT, and ONU). EPON elements combines a set of extensions such as Media Access Control (MAC), Multi-Point MAC control sublayers and a mechanism for Operations, Administration, and Maintenance (OAM) in the data link layer with a number of physical layers such as Reconciliation sublayer (RS), Physical Coding sublayer (PCS), Physical Medium Attachment (PMA) and Physical Medium dependent (PMD).

The physical layer includes the optical fiber and Physical Medium Dependent (PMD) sublayers for point-to-point (P2P) connections in the subscriber access network. The P2MP network topology is implemented with passive optical splitters, along with extensions to the MAC Control sublayer and Reconciliation sublayer, as well as optical fiber PMDs to support this topology. The reconciliation sublayer (RS) interconnects the MAC sublayer and the PHY sublayers. The RS sublayer includes a point-to-point emulation that allows an underlying P2MP network to appear as a collection of point-to-point links to the higher protocol layers (above the MAC client). The emulation creates virtual private paths to each ONU. The Physical Coding Sublayer (PCS) is used to encode data and transmit them over the physical medium, and turn the laser on and off at the right times. The Physical Medium Attachment (PMA) specifies a time interval required as a clock and data recovery time by the receiver to acquire phase and frequency lock on the incoming data stream. This specification requires the PMA sublayer to be instantiated in the OLT and ONUs to keep synchronization.

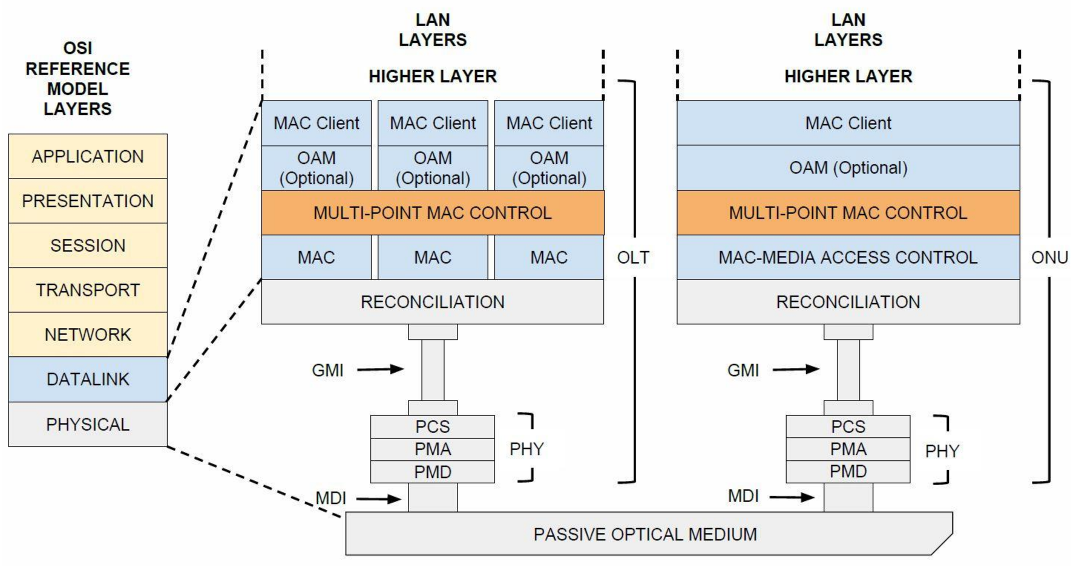


Figure 2-2 Protocol architecture of OLT and ONU of the EPON, [8].

The Point to Multi-Point MAC control (MPMC) is defined in clause 64 of the IEEE 802.3ah standard. As shown in Figure 2-2, the Multi-Point MAC control is composed of one or more MAC instances, where the number of MAC instances depends on the number of ONUs connected to the OLT. On the contrary, on the ONU side there is only one MAC instance. The MAC control sublayer has extensive functionality designed to manage the real time control and manipulation of the MAC sublayer operation. A functionality of the Multi-Point MAC control sublayer is the Multi-Point Control Protocol (MPCP), which is in charge of managing the exchange of messages between the OLT and ONUs during the initial registration phase, and also during normal operations such as synchronization and ranging (Round Trip Time computation), bandwidth arbitration, timeslot assignment to ONUs, and discovery functions. MPCP defines a state machine, messages and several timers to control access to the upstream channel, one ONU at a time.

In order to avoid collisions, the system has a strict synchronization, which ensures that in the upstream channel each ONUs occupies only the timeslot assigned by the OLT. Every ONU has a different distance from the OLT, and thus a different one-way delay and round trip time (RTT). The delay between ONU and OLT is measured during the initial phase and is equalized in order to make the logical distances between the OLT and the ONUs the same; i.e., the absolute time of each transmission is advanced or delayed in order to compensate the propagation delay.

The main challenge for EPONs is how to organize the medium access control (MAC) to the shared optical transmission medium. In the downstream channel Ethernet frames transmitted by the OLT pass through a set of passive splitters and a copy reaches every one of the ONUs, thus behaving as a point-to-multipoint architecture. The upstream channel is arbitrated in order to avoid collisions between the transmission attempts of the ONUs (thus behaving like a point-to-point architecture). The Point-to-point emulation achieved by adding a Logical Link Identification (LLID) ahead of every frame. The LLID replaces two bytes in the preamble of the frame. The OLT can distinguish frames of different ONUs by the LLIDs.

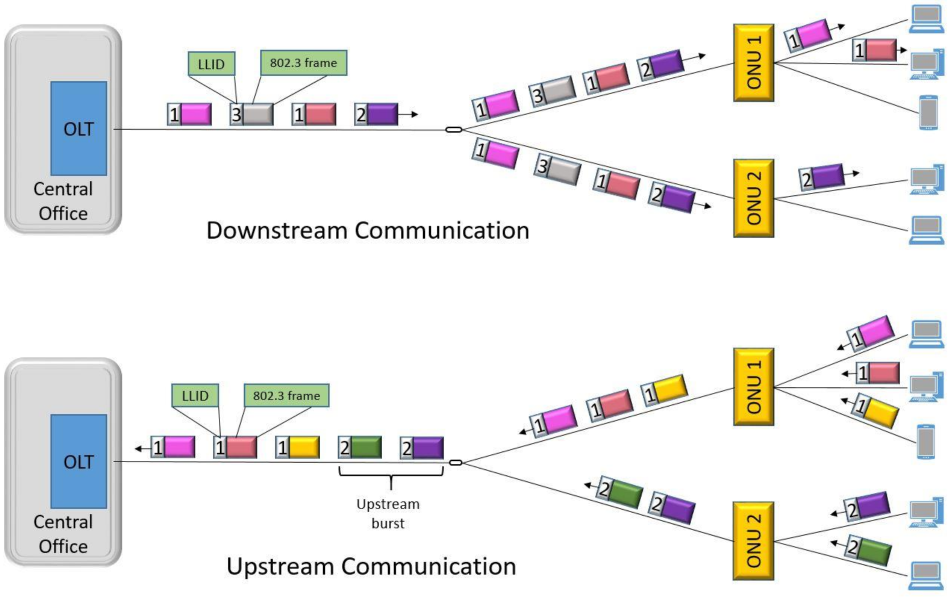


Figure 2-3 EPON operation

The standards defines the messages, leaving undefined the details of the specific medium access control. In the EPON case, a hybrid Time Division Multiplexing (TDM) and Wavelength Division Multiplexing (WDM) technique is used [12]. The combination of these mechanisms are used as a channel separation technique in order to avoid collision of optical signals that could occur in data transmission in upstream direction (from ONUs to OLT).

MPCP introduces GATE and REPORT messages in order to allocate the required bandwidth to each ONU, request bandwidth and transfer data at the corresponding time from ONUs to OLT, respectively. The GATE message performs two roles: a discovery GATE message for the registration phase, and a normal GATE message for assigning the transmission time and bandwidth to each ONU. The discovery GATE message is used as the main operation for initial registration phase. The purpose of the initial registration is to complete the discovery and register of newly connected ONUs.

For the initial registration phase, the following messages are defined:

- REGISTER_REQ: unregistered ONUs use this message to respond to discovery GATE messages. In the downstream channel, the OLT sends GATE messages, which all ONUs receive (broadcast mode).
- REGISTER: the OLT uses this message to assign an unique identifier to a newly registered ONU.
- REGISTER_ACK: sent by the ONUs to the OLT as a final registration acknowledgment.

The OLT opens periodically a discovery time window in order to allow unregistered ONUs to attempt register at the beginning of the window, the OLT broadcasts a discovery message, which includes the starting time and length of the discovery window. ONUs, upon receiving this message, wait for a random time (in order to avoid collision, in case all receivers answered at the same time) and then transmit a register message to the OLT. The off-line ONUs, after receiving the discovery message, will be registered during the previously established window. This window is unique because this is the only time that ONUs can communicate with the OLT without a specific grant window.

The way the OLT distributes the bandwidth depends on a medium access mechanism implemented by a Dynamic Bandwidth Allocation (DBA), which is outside the scope of the standard and is left open to be implemented by the equipment vendors. Apart from bandwidth request, the REPORT message is used to report local queue status to the OLT. It is used to carry data (in the form of several Ethernet frames piggybacked to the REPORT messages header), which were assigned previously by the OLT via a GATE message. The REPORT messages have to be generated periodically even when no request for the bandwidth is being made. The OLT must grant the access to the ONUs periodically to keep a watchdog timer in the OLT to prevent it from deregistering the ONU.

The OAM (Operation, Administration, and Maintenance) is a sublayer of EPON, which is indicated as an optional part for both OLT and ONU (Figure 2-2). OAM allows network operators to monitor the network, detect fault conditions, and determine locations of failure. OAM functions provided for EPON include remote failure indication, remote loopback control and link monitoring, among others. Remote failure indicates that the reception path of the local device is not operational. Remote loopback control provides support for level frame loopback and a data link layer ping. Link monitoring provides event notification with covering of diagnostic of the data and polling of variables in IEEE 802.3 management information base. The OAM of 802.3ah provides the possibility of adding flexible, extended functions that can achieve the functions that are not supported in the standard, through the use of extended instance reports or information frames. The OAM protocol is capable of negotiating the set of OAM functions that can operate on a given link interconnecting Ethernet devices.

2.3.1 Bandwidth Allocation

In the EPON architecture, resource allocation is defined by the Dynamic Bandwidth Allocation mechanism (DBA). DBA is a medium access control, which is used to achieve the best utilization of the available bandwidth in a shared medium network. Although the details of DBA and scheduling are not specified by the EPON standard, vendors and researchers have developed many interesting schemes to facilitate high bandwidth utilization and robust SLA provisioning.

The problem of how to optimally allocate bandwidth in PON has been discussed in [13] [14] [15] [16] [17] and [18], to name a few examples. Bandwidth allocation can be categorized into two categories: static and dynamic. The static allocation dedicates a fixed amount of bandwidth to each ONU (fixed time slot) regardless of the variation of the traffic profile of the sources. This is a simple algorithm to implement but it does not perform optimally. On the contrary, dynamic bandwidth allocates bandwidth to each

ONU based on their demands. Obviously, a dynamic algorithm provides more fair and efficient bandwidth allocation. The distributed bandwidth allocation (DBA) can be performed by OLT (centralized) or by the cooperation of the OLT and the ONUs (distributed), as we will detail in the following subsections.

The MPMC sublayer performs the dynamic bandwidth allocation in the upstream channel of EPON. Each ONU calculates its own required bandwidth according to the buffer occupation, and generates a REPORT request message that is transmitted to the OLT at specified time slots. The OLT receives the request message and allocates the required bandwidth to each ONUs based on its defined algorithm. The OLT sends a GATE grant message to each ONU through the downstream channel, indicating the granted bandwidth and the timeslot. Eventually, ONUs sends their frames in the time slot specified by the OLT.

2.3.1.1 Example of a Centralized DBA

Interleaved Polling with Adaptive Cycle Time (IPACT) [19] is an algorithm that was proposed to reduce the wait times (time that each ONU must wait for the next transmission time) and cycle time (time taken to poll all ONUs on the network). It is the best example of a centralized DBA scheme. An interleaved polling scheme is used to schedule data transmission of ONUs, so the other ONUs are able to transmit their data between two adjacent transmissions of an ONU. A GATE message is transmitted to an ONU, and then the OLT schedules the GATE message of the next ONU. The next ONU is pulled before transmission of the previous one is finished in order to utilize the channel efficiency. ONUs send REPORT messages to the OLT for requesting bandwidth, and the OLT allocates the required transmission slot time by sending GATE messages in an interleaved fashion to the ONUs for covering the round trip delay.

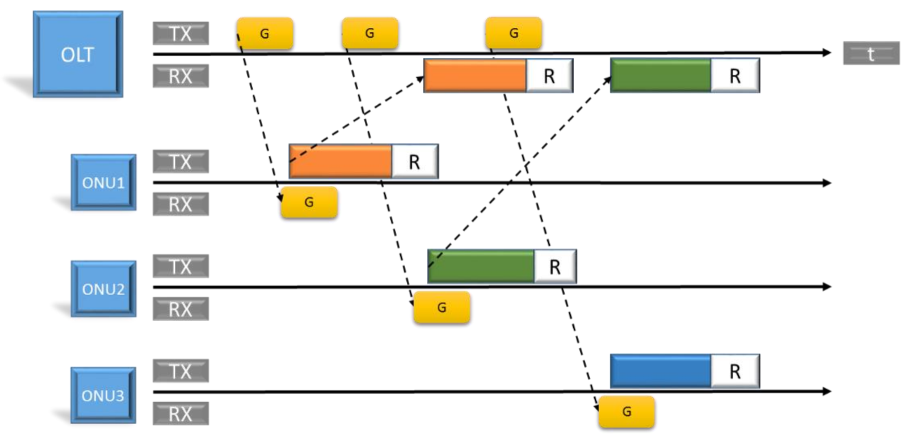


Figure 2-4 IPACT Polling mechanism, from [19].

The IPACT granted window size is one of the most important challenges for distributed bandwidth allocation (DBA). There exist six different schemes:

- Fixed: the OLT ignores the request window size and always grants the maximum window. The cycle time is constant.
- Gated: the OLT authorizes the ONU to transmit as much as data as it has requested. The cycle time is limited by the maximum window.
- Limited: the OLT grants the requested number of bytes, but not more than a predefined maximum value. The cycle time can be variable.
- Constant-credit: the OLT adds a constant credit to the requested window size. The constant-credit reduces delay for packets arrived between the time when a REPORT message was sent and the grant was received.
- Linear-credit: it uses an approach that is similar to the constant credit case. The OLT calculates the window size of credit which is proportional to the requested window size.
- Elastic: the OLT controls a fixed maximum window limit based on the accumulated size of the last N grants requested by the N ONUs.

The authors of [20] proposed an active intra-ONU scheduling to make the result from the inter-ONU scheduling predictable, thus preventing the USR (Unused Slot Remainder)[1] problem. Each ONU receives a predictable bandwidth allocation from inter-ONU scheduling.

There are three steps for the DBA algorithm based on the active intra-ONU scheduling:

- Record_Report: every ONU has to calculate several levels of bandwidth requirement instead of only the maximum bandwidth requirement as in the passive intra-ONU scheduling, in order to prevent USR occurrence. So each ONU calculates its bandwidth packet by packet from the first packet in the highest priority queue to the last packet in the lowest priority queue.
- Bandwidth_Assignment: the OLT starts inter-ONU scheduling after receiving one or more or all REPORT messages.

[1] USR refers to the case where a particular allocated slot is never used, causing waste of bandwidth.

- Data_Transmission: the ONU receives a bandwidth grant in the GATE message of the OLT and checks which level is equal to the bandwidth granted. Then when transmission start ONU transmits packets in each queue to the OLT according to level information's in ONU memory.

In [21], the authors proposed an algorithm based on IPACT, where OLT employ a control based traffic prediction to increase the grants in order to allow additional traffic arriving at the ONUs. The prediction calculates the difference between the current reported queue size at the ONU and the previously reported queue size and increases the size of grants accordingly. A control parameter is also used to prevent system instability. The traffic model used assumed piecewise-constant with fast settling and hence did not exhibit a high degree of burstiness.

2.3.1.2 Example of a Distributed DBA

Dynamic Distributed Scheduling for EPON (DDSPON) is proposed in [22]. Its main contribution is to distribute the computation of the DBA among the ONUs, which can update their bandwidth requests directly without intervention of the OLT, thus allowing a much faster response time than a centralized scheme. It also introduces an important novelty: fairness in the bandwidth allocation. DDSPON extends the simple algorithm of IPACT by adding extra information in the GATE messages. This extra information includes a weight vector that allows ONUs to schedule its transmission window size. The bandwidth allocation is mainly performed by each active ONU, so the ONUs are able to proportionally schedule its transmission window size by themselves, based on its current queue requirements and the rest of ONUs. A key different aspect of DDSPON with respect to other algorithms is the distributed calculation of the bandwidth performed by ONUs.

A weight vector $\vec{\Phi}$, which represents the fraction of bandwidth assigned to each ONU, is carried in the control messages of the MPCP extension of the DBA algorithm. This weight vector allows ONUs to calculate its transmission window size. The ONU calculates the required bandwidth and the current weight and reports the calculated value through REPORT messages to the OLT. As illustrated in Figure 2-5, the size of transmission window for the ONU is calculated by each respective ONU.

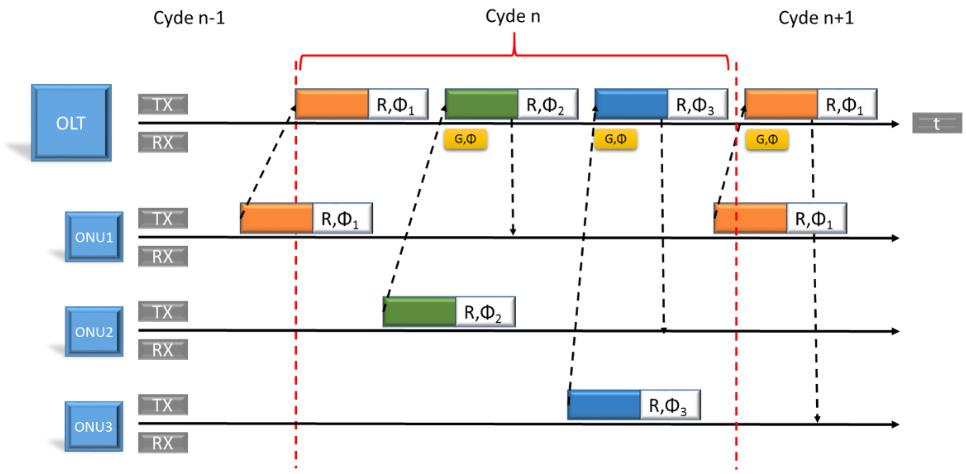


Figure 2-5 DDSPON Polling Mechanism, from [22].

At the beginning each ONU has a fixed weight based on its guaranteed bandwidth agreement, so every final user is assigned a specific bandwidth during each upstream transmission cycle. The average transmission window size (weight assignment) of each ONU is shown as follows:

$$W_i = \frac{\Phi_{\mathrm{i}}}{\sum_{j=1}^{n=N} \Phi_{\mathrm{i}}} W_{max}$$

The W_{max} corresponding to the maximum cycle time (T_{max}) calculated as follows:

$$W_{max} = Upstreamrate * T_{max}$$

Bandwidth allocation in DDSPON is as follow: The $ONUi$ sends a REPORT message to the OLT that contain window size (R_i^n) and the weight (Φ_i^n). The OLT updates the weight vector and sends a GATE message to the ONU_i in the next cycle. The ONU_i transmits data once it receives a GATE message from the OLT, and then ONU_i calculates the new value for the maximum window size and required transmission window size of the ONU_i for the next cycle (n+1) respectively as follows,

$$W_i^{n+1} = \frac{\Phi_i^{conf}}{\Phi_i^{conf} + \sum_{j=1;j\neq i}^{N} \Phi_j^n} W_{max}$$

$$R_i^{n+1} = min(W_i^{n+1}, Q_i)$$

where Q_i is the size of queue. Finally the new weight (Φ_i^{n+1}) for the next cycle (n+1) is calculated based on the request value for cycle (n+1) as follows:

$$\Phi_i^{n+1} = \frac{\mathrm{R_i^{n+1}}(\Phi_i^{conf} + \sum_{j=1;j\neq i}^{N} \Phi_j^n}{Wmax}$$

Table 1 Notation used in the DDSPON bandwidth allocation mechanism.

Notation	Value
N	*Number of ONUs*
ϕ_i	*Set up weight value of the* ONU_i
ϕ_i^{conf}	*Initially configured weight value of the* ONU_i
ϕ_i^n	*Weight of the* ONU_i *computed in cycle n*
W_i	*Transmission window size of the* ONU_i *computed in cycle n*
W_{max}	*Maximum transmission window size (bits)*
T_{max}	*Maximum cycle time*
R_i^n	*Requested transmission length of* ONU_i *in cycle n (bits)*
Q_i	*Queue size of* ONU_i *(bits)*

Every ONU performs a similar process, and dynamically schedules the size of its transmission window. The scheduling process in DDSPON is executed without needing to wait for the reception all the REPORTs from the ONUs to the OLT. Moreover, thanks to the weight vector, each ONU can get an overview of the load of the rest of ONUs.

In [23], the authors proposed a scheme based on the Deficient Weighted Round Robin (DWRR) scheduling to achieve adaptive fairness among different classes of services. They showed that every class of traffic gets a fair sharing of assigned bandwidth at the ONU by forcing the scheduler to visit every Priority queue for a specific period of time that is determined by the weight allocated to the corresponding Priority queue. In general the advantage of such system is that each ONU can adaptively (depending on the traffic demand and SLA) set its own weights in both phases (i.e. initially or after computing remaining bandwidth).

Efficient Distributed DBA (EDDBA) is Proposed in [24] to support DiffServ and quality of service for both inter-ONU and intra-ONU allocations algorithm independently at each ONU. Based on this framework each ONU decides the size of time slot (the authorization length is the time slot requested by ONU). The OLT adds some extra information in the GATE frame, with which the ONU is able to compute the size of the uplink transmission window.

2.3.2 Quality of Service in EPONs

EPON is designed to handle multiple traffic, services and applications, such as voice communications (VoIP), video conferencing and data traffic. These different services have strict bandwidth and delay requirements which if not meet, the service quality can degrade or risk disconnection. Therefore, the quality of service (QoS) mechanisms in EPON are responsible of assigning different priority to different users and applications, and guaranteeing a certain level of performance data flows [25].

EPON is able to allocate different bandwidth and quality of services to each user by performing inter-ONU and intra-ONU scheduling. Inter-ONU scheduling is performed by the DBA process within the OLT, by arbitrating the transmission of different ONUs and is defined by IEEE standard 802.3ah. Intra-ONU scheduling is concerned with arbitrating transmission of different priority queues and traffic classification within each ONU [26] [27] and is defined by IEEE standard 802.1D. The intra-ONU DBA algorithm can be managed either in the OLT or in the ONU. Inter and intra-ONU scheduling is illustrated in Figure 2-6.

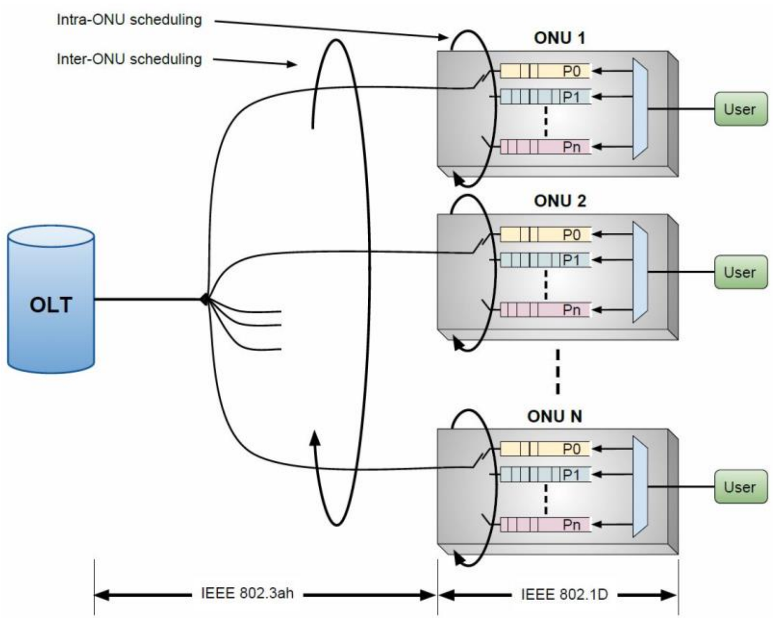


Figure 2-6 Inter-ONU and intra-ONU scheduling in EPON, from [26].

There are two different QoS families of algorithms for scheduling in EPON: online and offline scheduling. In online scheduling, upon receiving a REPORT message from each ONU, the OLT decides the required bandwidth allocation for the ONU and sends a GATE message. IPACT and DDSPON are examples of online scheduling. They reach a high throughput but have a variable cycle time. The variable cycle time causes high jitter and variable packet delays, which are not appropriate for applications that are jitter sensitive, such as IP telephony, IPTV or, in general, any real-time transmission [28] [29]. In offline scheduling, the OLT waits until all the REPORTs from every ONU are received, then calculates and performs the best bandwidth assignment by sending GATE messages to the corresponding ONU. The offline scheduling improves the delay performance at low load, but at high load the delay increases excessively. Also, the idle time increases dramatically due to the time to collect all the REPORTs from every ONU, causing the offline scheduling to obtain low throughput [30] [31].

2.3.3 Energy Saving

Since the energy consumption and environmental protection concerns are increasing in recent years, energy saving techniques are becoming more and more important in the research literature [32]. In the past few years, optical networks have rapidly gained large popularity in the broadband access network; hence, energy consumption of network devices has increased as well [33]. Therefore, many researchers are developing mechanisms to reduce the energy consumption and cost.

Energy saving becomes even more important as the access network technologies evolve into the next generation with larger amount of users and higher data rates. Energy saving in the access network can be achieved at both the physical and medium access control (MAC) layers [34] [35]. In the physical layer, approaches introduce new energy efficient optical device architectures, while the MAC layer controls the device energy consumption by disabling or enabling a particular function module based on the network operation.

Energy saving mechanisms in PON networks can be classified into two categories: OLT-side energy saving and ONU-side energy saving. The OLT-side energy saving has received less attention, but due to the increased provisioning data rate in PONs, which implies more energy consumption, it is desirable to focus on OLT structural designs with lower power consumption.

We now list some of the most relevant contributions in the OLT-side energy saving technique. A sleep control functions for power saving in EPON networks is proposed in [36] to employing duplicated layer-2 switches. The idea is to put one of the duplicated layer-2 in sleep mode in the OLT. The proposed sleep control function calculates the traffic in advance from the DBA information, and switches one of the duplicated L2SW between active and sleep mode based on the accurate traffic predictions.

In [37] the energy consumed by the OLT is reduced by using fewer wavelengths and putting idle transmitters to sleep. Authors proposed to use thresholds to determine, based on downstream network traffic, when to use one wavelength less and when to add one more.

The authors of [38] proposed a new OLT hardware configuration by placing an optical switch in the OLT. The OLT comprises multiple OLT line cards for adapting to the traffic load, each of which serves a number of ONUs. The optical switch in the OLT can configure the connection between the OLT line cards and ONUs, so in the light loaded network traffic case, the optical switch forces ONUs to communicate with the minimal number of OLT line cards and the other OLT line cards can be powered off.

Unlike OLT-side, energy saving mechanism in PON are designed preliminary for ONU-side and consist on exploiting the variation of upstream and downstream traffic load by switch between different energy saving modes. The Telecommunication Standardization sector of the International Telecommunication Union (ITU-T) has proposed appropriate methods for minimizing the energy consumption of PON networks. As a result, energy saving in the ONU were standardized by the ITU-T series G.45 for PON [39]. When improving the energy saving of ONU, two main approaches are taken; hardware-based and software-based (algorithm-based). These two techniques can work independently or jointly.

Hardware techniques can improve energy consumption of an ONU by focusing on the physical modification of modules and changing the architectural. First generation of ONUs did not support sleep mode, and therefore always consumed maximum energy. Later on, some new architectures proposed support of clock recovery time and synchronization periods in order to enable sleep mode functionality with doze and sleep procedures. The amount of energy saving in the hardware technique depends on how modules can quickly handle clock recovery, synchronization period, and turning on/off the elements [40].

Software-based technique perform in a different way; they introduce changes on how the devices element should be implemented logically, and provide flexibility through an algorithm to manage elements of the network. Software-based techniques can be classified in three modes based on the ONU transmitter and receiver behavior: power shedding, dozing and sleeping modes [41]. These modes involve powering-off individual components of the ONUs based on the network traffic and the idle periods. Table 2-2 summarizes the energy saving modes for the ONU.

Table 2-2 Power saving modes in ONU

Mode	**Description**
Power shedding	*An ONU maintains operational the transmission and receiving physical elements ON while powering-off or reducing power to non-essential functions and services in upper layers.*
Dozing	*An ONU powers-off its transmitter while its receiver remains ON during the power saving state. Doze mode allows ONU to remain synchronous with the OLT.*
Sleeping	*Both transmitter and receiver of the ONU are powered-off in absence of traffic.*

The main advantage of doze mode is that ONU can always receive control frame traffic. However, due to the always-on receiver, the doze mode achieves only small amount of power saving. On the other hand, in the sleep mode, because of powering-off both transmitter and receiver, the power saving increases, while in this operation mode, there is possibility of losing control messages during power saving mode. Therefore, ONU can be deregistered from the network, which imply a temporal loss of connection.

The IEEE 802.3ah standard declares that the maximum time for receiving a REPORT message in the OLT from ONUs must be 50 *ms*, otherwise ONU will be deregistered from the network and thus deregistered ONU should wait for a period of auto-discovery and registration in order to establish connection again. Deregistration of the ONU introduces incremental delay and degrades the quality of service of the end users.

Among the power saving modes, many research efforts have been devoted to enabling the sleep mode in ONU. The main challenges of this approach are the slow transition from active mode (ONU transceiver and the necessary supporting functions are powered-on) to sleep mode (ONU transmitter and receiver are powered-off), and the time needed during the wake-up process. Depending on the traffic load, an ONU sleep mechanism reduces energy consumption by keeping transmitters/receivers inactive if the traffic arrival is low (according to a defined threshold). The definition of the threshold is a key issue, as an appropriate definition can lead to relevant performance improvements in terms of delay.

A new type of energy-efficient mechanism based on EPON is proposed in [42] by taking into consideration traffic in the OLT and the ONUs. The algorithm is based on the ONU sleep control and on balancing the downstream traffic of OLT and upstream traffic of ONU at the same time, in order to reduce energy consumption and improve the quality of service of the end users. The OLT distributes the upstream time slot to the ONUs with the standard GATE messages. Then each ONU sends upstream data during its corresponding time slot and changes its state to sleep during the time slot that is assigned to other ONUs.

In [43] the authors propose a downstream traffic scheduling rules at the OLT and a sleep control scheme at the ONUs. The scheme allows ONUs to be aware of the downstream traffic scheduling rules, and ONUs can infer their downstream queue status instead of being explicitly notified by the OLT. When an ONU does not receive downstream traffic for a certain scheduling cycle, the ONU can switch into sleep state and sleep for a preconfigured time. This scheme can achieve up to 50% energy saving when the network is lightly loaded.

In [44] the authors proposed an energy efficient DBA framework to reduce energy consumption. The DBA framework includes an energy efficient MAC control scheme, a grant sizing policy and two grant scheduling algorithms. The proposed MAC control scheme is based on the three operating modes of ONU (sleep, active and doze). Since the MAC control scheme is offline-based, the OLT periodically allocates transmission windows to all ONUs and puts them into sleep/doze mode in each polling cycle to reduce energy consumption. The results show that the DBA framework can achieve significant energy saving in light load traffic.

A dynamic energy-efficient bandwidth allocation (DBA) algorithm for EPONs is presented in [45]. The DBA algorithm is based on the sleep mode functionality. It is applicable to legacy ONUs (L-ONUs) with large sleep overhead and next generation ONUs (NG-ONUs) with small sleep overhead. The proposed scheme for L-ONUs can quit the sleep mode on the arrival of high priority traffic and can thus meet strict QoS requirements. L-ONUs shows can achieve average energy savings between 30-70% of the time depending on the network load. On the other hand, for NG-ONUs, the authors introduced a sleep-period prediction scheme and the minimum sleep time to maintain the delay bound and simultaneously reducing frequent mode switching. In this case, the ONUs remain in sleep state up to 70% of the time.

[46] Presents a parametric extension for unifying power saving in PON networks. In this scheme the ONUs periodically turn off their receiver and transmitter based on a cyclic sleep mode, and perform infrequent bidirectional handshakes in the doze mode. Authors simplify the ONU implementation by eliminating the redundant state of the controlling state machine. The doze-aware and sleep-aware states are merged into the single-aware state, and the asleep and doze states are merged into the single low-power state. The low-power state makes ONU to alternate between an intermediate power level with the receiver on and a low power level with the receiver off. The unified power saving mode can be applicable to G-PON, XG-PON1 or any related PON system.

In [47], authors proposed a downstream simple-sleep control scheme and a sleep-aware traffic scheduling scheme to avoid missing packets during the sleep state. The handshake process is eliminated in the sleep control scheme. It avoids frequent changes in energy modes of ONU by putting ONU in listening mode for a period before going into sleep mode or changing to active mode. The active mode is used to deliver packets and save energy in sleep mode. The listening mode is used to monitor the traffic state to reduce the frequency of mode change.

The author of [48] proposed two just-in-time dynamic bandwidth allocation algorithms for 10 Gbps Ethernet passive optical networks. One algorithm is based on varying polling cycle time (JIT) and the second algorithm is based on fixed polling cycle time (J-FIT). The energy saving can be achieved when the idle time of an ONU is less than the sleep to active transition time. This is only feasible by a vertical-cavity surface-emitting laser (VCSEL)-based ONU, which allow ONU to change its mode to sleep or doze modes during the idle time. Result shows that just-in-time with fixed polling cycle time performs better and reduce energy consumption at low network loads, while the average delay of the network is increased.

The following table summarizes the main proposals:

Table 2-3 Energy saving techniques

	Approach		Energy Saving	Technology
OLT	*[36] Proposes a sleep control function by using duplicated layer-2 switches to put one of the L2SWs in sleep mode in the OLT.*		*Medium*	*EPON*
	[38] Introduces an optical switch with multiple OLT line cards for several ONUs.		*Medium*	*EPON/GPON*
ONU	*Doze mode*	*[44] Proposed a DBA with an energy efficient MAC control scheme to put ONU into doze mode at the beginning of the polling cycle to receive the GATE message.*	*Low*	*EPON*
		[48] Achieves saving energy when the idle time of an ONU is less than sleep to active transition time.	*Medium*	*G-EPON*
	Sleep mode	*[42] Assigns uplink timeslot to ONU based on the threshold and the average frame length of ONUs.*	*Low*	*EPON*
		[43] ONUs aware of downstream traffic scheduling rules and queue status, to switch into sleep state in zero downstream data traffic for a certain scheduling cycle.	*Good*	*EPON*
		[44] Proposes a DBA with an energy efficient MAC control scheme to put ONU into sleep mode based on the queued data through a REPORT message, when upstream and downstream data transmissions finish.	*Low*	*EPON*
		[45] Presents a DBA algorithm for L-ONUs and NG-ONUs, which quit the sleep mode on the arrival of high priority traffic, and maintain the delay bound and simultaneously reducing frequent mode switching to keep ONU into the sleep state, respectively, at low traffic load conditions.	*Medium*	*EPON*
		[47] Increases the number of sleep cycles to improve energy efficiency at slower rate. It avoids frequent changes in energy modes of ONU by putting ONU in listening mode for a period. The delay performance worsens linearly.	*Good*	*EPON*
		[48] Achieves saving energy when the idle time of an ONU is less than sleep to active transition time.	*Medium*	*G-EPON*
	Unified mode	*[46] The ONUs periodically turn-off their receiver and transmitter as in the cyclic sleep mode and perform infrequent bidirectional handshakes as in the doze mode.*	*Good*	*GPON*

2.4 Service Interoperability in Ethernet Passive Optical Networks

The IEEE 1904.1 Service Interoperability in Ethernet Passive Optical Networks (SIEPON) standard [49] [50] creates an open system-level specification to provide plug and play interoperability of service, transport and control planes in a multi-vendor EPON and GPON [51] environment. It intended to realize service interoperability among 1G-EPON or 10G-EPON equipment, and focus on the development of a unified EPON data path architecture and interoperability of the OLT and the subtended ONUs to ensure that data services can be managed, provisioned and monitored across the EPON. The scope of SIEPON standardization includes physical (PHY), MAC layers and upper layers in order to support all functions of data path such as multicast, tunneling, VLAN, and quality-of-service (QoS). The scope of IEEE 802.3 and IEEE 1904.1 standardizations are shown in the Figure 2-7.

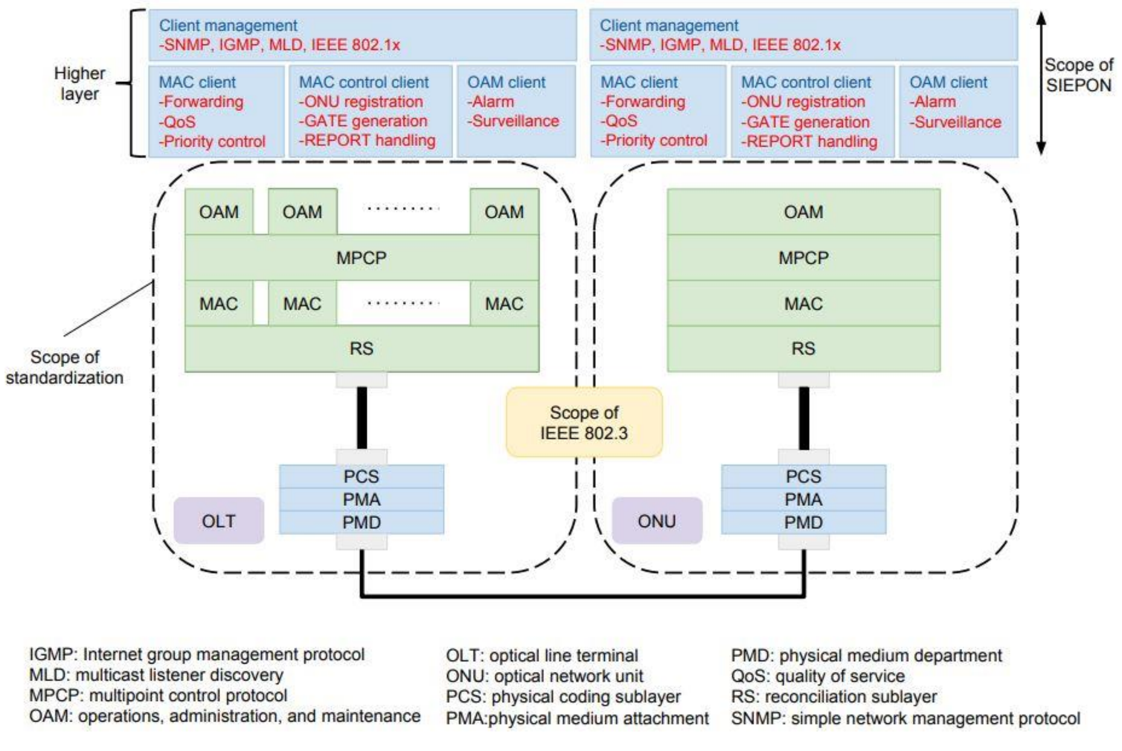


Figure 2-7 Scope of IEEE 802.3 and IEEE 1904.1 standardization, from [52].

Out-of-scope functions of the IEEE 802.3 standard include dynamic bandwidth allocation (DBA) mechanism, packet classification rules, software/firmware download, service protection and restoration mechanism and power saving modes of operation. In traditional switched Ethernet devices, these exclusions are understandable because they are implementation choices and ordinarily do not affect service interfaces or interoperability. However, at the system level, EPON devices (OLTs and ONUs) differ from typical layer 2 switches or bridges in some important aspects. EPON can be viewed as a distributed switch with various features such as path protection, power saving, service quality, and management require coordination between the OLT and the ONUs [52].

The main aspects that SIEPON addresses for the EPON networks are:

- Service configuration and provisioning: features affecting connectivity, frame classification and manipulation, and forwarding rules, to name a few examples.
- Performance requirements and service quality: features that affect service performance (real-time control of delay, jitter, packet loss and bandwidth guarantees).
- Service Survivability: feature that affect service availability (monitoring mechanisms, system alarms, path protection, and power saving methods).
- System/device maintenance and management: features required to operate EPON.

Regarding the aspect of service survivability and service quality, [53] and [54] proposed a scheme for the SIEPON standard, which includes new components in the OLT and ONU. A green DBA, an ONU sleep manager, and a transmitter controller are added to guarantee the QoS and have more control over the energy saving mechanisms. In this scheme, each ONU calculates the transmitter sleep time based on its current queue state and the maximum boundary delay requirement. The OLT is responsible for calculating the receiver sleep duration and determines whether the ONU's sleep mode is transmitter or transmitter/receiver mode.

2.4.1 Architecture Model of SIEPON

SIEPON provides a flexible architecture for EPON and GPON networks and defines the functions for MAC control, OAM and MAC clients. As shown in Figure 2-8, the architecture model of SIEPON for OLT and ONU are separated into Line OLT/Line ONU having functions covered by the IEEE 802.3 standards, Client OLT/Client ONU having functions covered by SIEPON standards, and Service OLT/Service ONU having additional functions for specific services provided by the communication carrier or system vendors.

The L-OLT and L-ONU are capable of sending and receiving various types of Ethernet frames including data, OAM and MPCP frames (REPORTs and GATEs). However, they cannot initiate the MPCP discovery and registration processes and they do not contain the data path components such as frame buffers, shapers, policers and schedulers.

A C-OLT combines at least one L-OLT with higher-layer functions, including the OAM client (statistics, alarms, SNMP, IGMP/MLD protocol), MAC control client (discovery and registration, GATE generation and REPORT processing), and MAC client (VLAN modes, tunneling, multicast, QoS features, buffering and scheduling). These clients reside above the L-OLT and interface to it via an OLT-LI interface. Similarly, C-ONU combines at least one L-ONU with the OAM, MAC control and MAC clients via an ONU-LI interface.

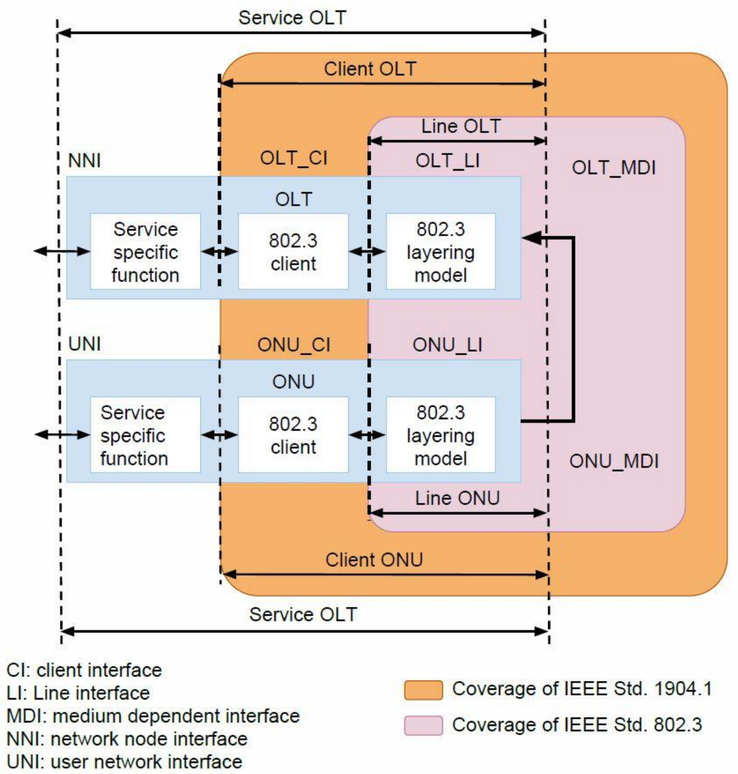


Figure 2-8 Architecture model of SIEPON, from [52].

An S-OLT combines a C-OLT with any service-specific functions present in the physical device. Examples of these functions include SIP, NAT, L2/L3 switching, carrier Ethernet services (CES) and VoIP, which are outside the scope of SIEPON. Similarly, an S-ONU combines a C-ONU with any service-specific functions present in the physical device. Figure 2-9 shows a detailed view of the OLT and ONU internal architectures.

The C-ONU/OLT have MAC Client functions not defined within the L-ONU/OLT and thus not covered in IEEE 802.3 the MAC Client reference model exits between the ONU_LI and ONU_CI of the Client ONU and between the OLT_LI and OLT_CI of the

Client OLT. The reference model provides a framework to achieve interoperability between different compliant ONUs and OLTs. This interoperability allows for distinct ONUs from different vendors to work with the same OLT as well as distinct OLTs from different vendors to work with the same ONU. The nature of this interoperability is to control the specification for information that passes between the ONU and OLT, and the corresponding actions of both, thereby allowing the network to achieve the same behavior among compliant ONUs and OLTs.

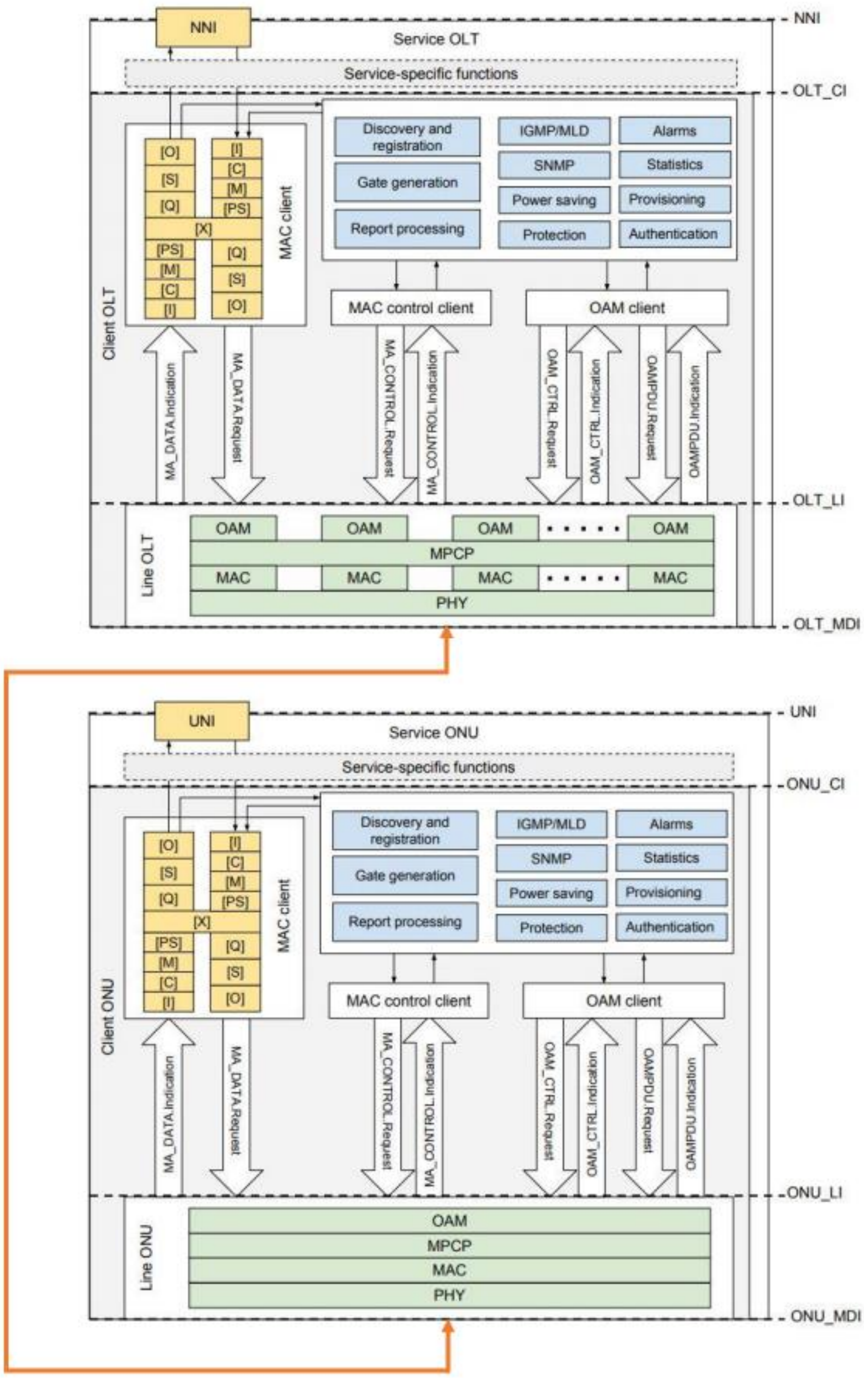


Figure 2-9 Internal architecture of OLT and ONU in SIEPON, from [49].

2.4.2 EPON Service Path

The MAC client reference model describes the connectivity and how performance may be ensured for frames traversing the client OLT or client ONU. SIEPON defines ESPs (EPON service paths) which provide basic functional blocks of the MAC client model to achieve provisioning, connection and QoS in services. The ESP is the conceptual path of a frame through the functional blocks of the MAC client. The basic functional blocks in ESPs that provide data forwarding in OLTs and ONUs are:

- ***Input block [I]*** is the ingress port that receives frames from the User-Network Interface (UNI), Network-Network Interface (NNI), MAC service interface or any service-specific function that may be present in the S-ONU or S-OLT.
- ***Classifier block [C]*** classifies incoming frames and includes a set of (provisionable) rules that compare incoming frame-header fields to predefined values. The Classifier can match L2/L3/L4 headers and a single rule may include multiple conditions that evaluate multiple fields in the field. A matched rule determines whether frame modification (for example, VLAN translation) is required, what rate control policy must be applied and to which queue the frame is sent.
- ***Modifier block [M]*** performs modifications to selected frame fields according to the provisioned operations. There are three basic operations: add, replace and remove field.
- ***Policer/shaper block [PS]*** enforces conformance to the provisioned service contract. As a basic provisioning element, the policer/shaper uses a token bucket (TB).
- ***Cross-connect block [X]*** directs individual frames to the proper queue(s). For unicast flow, the constituent packets are mapped to a single queue. For multicast or broadcast flow, the constituent packets may be replicated and mapped to several queues. For example, in the OLT, that simultaneously serves both 1 Gb/s and 10 Gb/s ONUs, each broadcast frame is duplicated and placed into the two separate queues serving the 1 Gb/s and 10 Gb/s broadcast channels.
- ***Queues block [Q]*** holds frames until they are selected by the scheduler block for transmission. This block may implement various queue admission policies (QAP).

To support various QAPs, the queues block tracks the fill levels and threshold-crossing levels and discards frames if needed.

- ***Scheduler block [S]*** multiplexes the frames stored in the queues block into the output block using a predefined scheduler algorithm. There is exactly one scheduler instance associated with each output block. The ONU's upstream scheduler is controlled by the OLT's DBA engine via GATE messages.
- ***Output block [O]*** is the egress port that receives frames from a scheduler and forwards them to the UNI, NNI, MAC service interface or any of the service-specific functions present in an S-ONU or S-OLT.

Each of the blocks may include multiple independently instances that act on different traffic flows. The example of the Figure 2-10 shows an ESP configuration connectivity of an OLT/ONU. The ESP concept is intended to be of help in understanding standard specifications because it simplifies and conceptualizes behaviors of MAC client functions, thus absorbing implementation differences such as single LLID (logical link ID) and multiple LLID configurations.

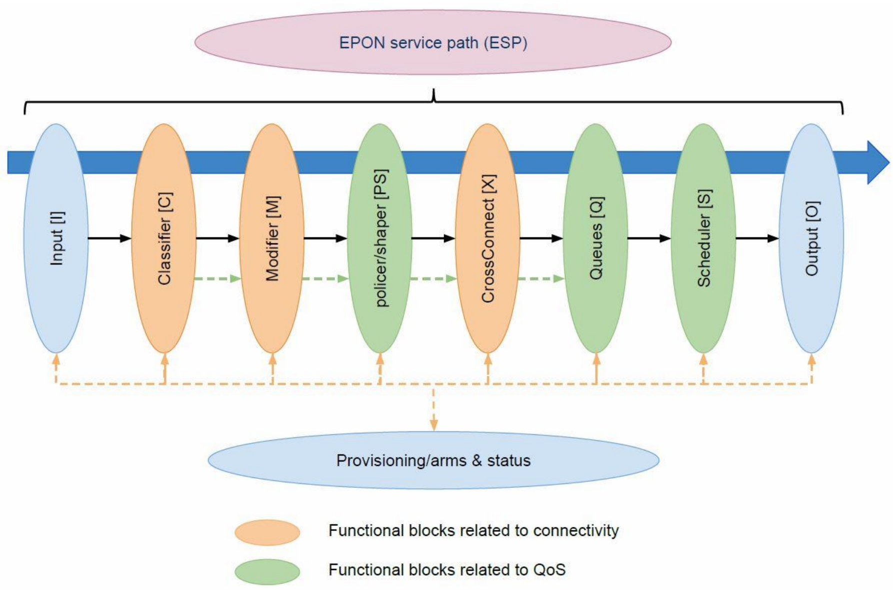


Figure 2-10 Logical ESP model of SIEPON, from [52].

The ESP traversed by a frame fully determines the connectivity-related and QoS-related treatment of the frame by the OLT or ONU. Connectivity is controlled by classification based on header fields (MAC address, IP address, VLAN tag, etc.) and by modifying the values of some fields. Performance guarantees are ensured by enforcing the provisionable QoS parameters, which control the way frames are queued, prioritized and scheduled.

2.5 Software Defined Networking

SDN is a novel paradigm that opens the way for a more efficient operation and management of the networks, allowing centralization of some functions and mechanisms that are distributed in the current network architectures [55]. Network administrators can manage network services through abstraction of lower level functionality, by decoupling the system that makes decisions about where traffic is sent (the control plane) from the underlying systems, that forwards traffic to the selected destination (the data plane). Moving the control plane out of the data plane elements enable both planes to function independently, thus introducing many advantages such as programmability, global optimization of network operations (load balancing, provisioning of services energy consumption reduction, and etc…), higher flexibility and possibility of realizing a centralized network view.

Logically centralized network control makes it possible for operators to specify more complex tasks that involve integrating many disjoint network functions (e.g. security, resource control and prioritization) into a single control framework, allowing network operators to create more sophisticated policies, and making configurations easier to configure, manage, troubleshoot and debug.

The Software Defined Network architecture is composed of three main layers: Application, Control, and Infrastructure (data path).

- Application Layer: it plays an important role from the innovation point of view; it provides great value to the network application such as network virtualization, load balancing, access control, network energy efficiency, and security.
- Control Layer: responsible of managing and controlling the behavior of network elements centrally. The main function of the SDN controller is to set appropriate connections to transmit flows between devices and control the behavior of network elements.
- Infrastructure layer: acts as a foundation for SDN and is responsible for packet forwarding. The infrastructure layer consists of physical and virtual devices or network elements such as switches.

The logical view of the SDN architecture, with the three layers, is illustrated in Figure 2-11. As shown, there is a communication interface between control layer and infrastructure layer. Currently, the most popular interface is OpenFlow, to be detailed in the next subsection.

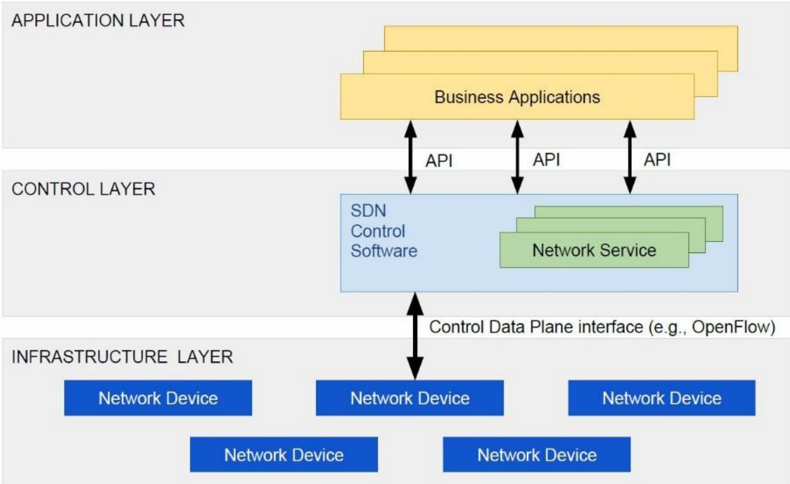


Figure 2-11 SDN architecture, based on [56].

2.5.1 OpenFlow

OpenFlow is the original southbound interface for communication between the control layer and the infrastructure layer (SDN network devices). The OpenFlow protocol [57] uses a standardized instruction set that allows the controller to send a common set of instructions to any OpenFlow-enabled switches. It is an open standards approach to virtualizing the network. It allows a remote controller to specify the path of network packets through the network of switches. The use of OpenFlow centralizes the control of network devices in a virtualized manner and mitigates network management complexity. Any devices (switches, routers) that want to participate in this network must be able to support OpenFlow with standardized interfaces.

OpenFlow uses the concept of flow to identify network traffic based on pre-defined match rules that can be statically or dynamically programmed by the SDN control software. Because OpenFlow works on a per-flow basis, it provides extremely granular control, enabling the network to respond to real-time changes at the user, application and session levels. As illustrated in Figure 2-12 OpenFlow removes the control plane responsibilities from the switch. When controlled via OpenFlow, the switch itself no longer makes forwarding decisions. Hence, the switch no longer needs to build and maintain information required to facilitate decisions.

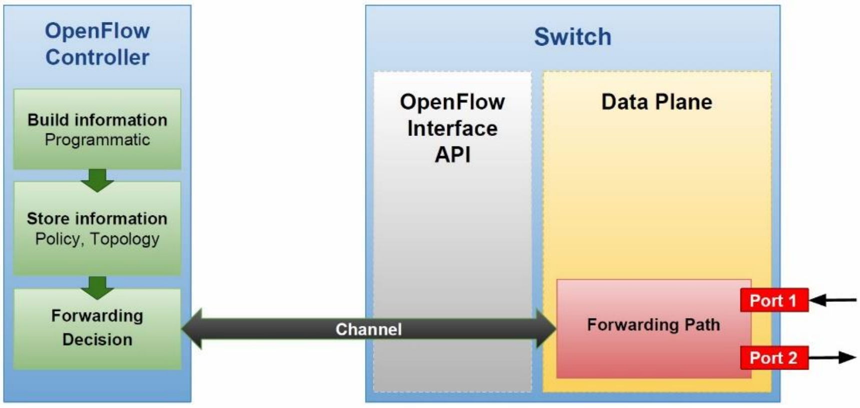


Figure 2-12 SDN/OpenFlow setup operation, based on [58].

The SDN/OpenFlow architecture defines a set of flow tables. When a packet arrives to an OpenFlow switch, a table lookup is performed in the first flow table, as illustrated in Figure 2-13. Packet match fields are read from the packet header and will be used to match against flow entries. Then, based on the instructions set in the tables, various operations can be applied to the packet. For example, the instruction can send the packet to the next flow table to match against different entries while writing actions to the action set and updating the metadata. The process continues until the action set contains an output action, and forwards the packet to the corresponding output port destination.

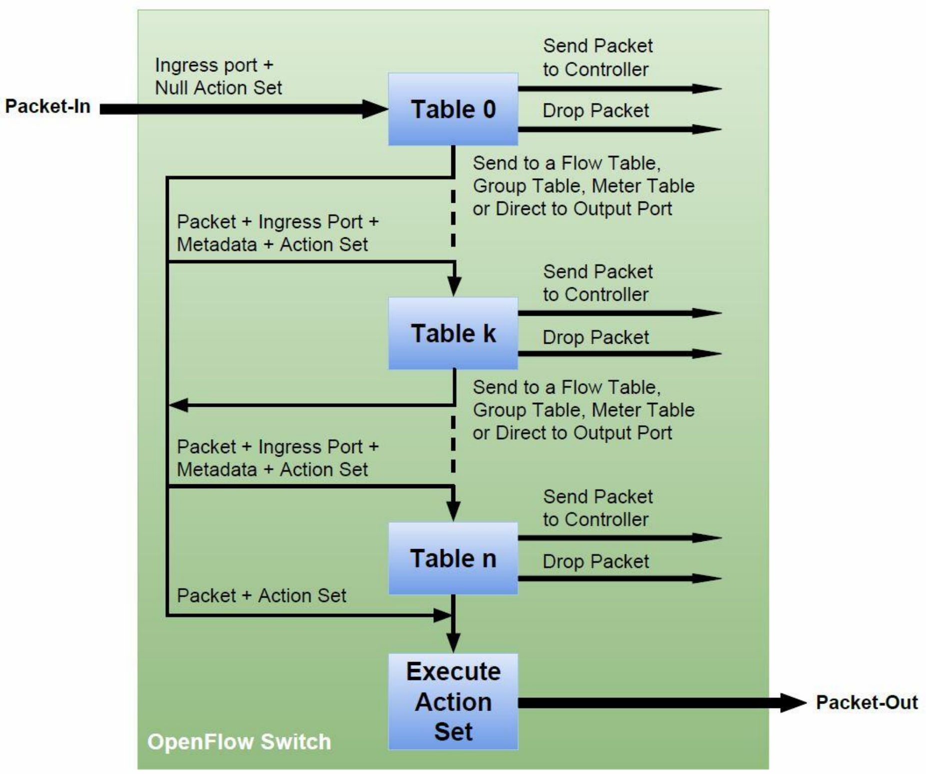


Figure 2-13 Packet flow through an OpenFlow compliant switch, based on [59]

The flow tables contain flow entries (Table 2-4), and each flow entries composed of set of fields such as:

- Match fields: to match against packets, such as packet headers fields and ingress port (in_port, eth_src, eth_dst, etc).
- Priority: match preference of flow entries. The flow entries with the highest priority will be selected.
- Counters: numbers of bytes or flows transmitted, for example the counters associated with the selected flow entry must be updated each time a new packet belonging to that flow arrives to the switch.
- Instructions: set of actions to apply to packets. Each time a flow is matched to a flow entry in the table, the instructions that were defined for that flow entry will be executed.
- Timeouts: define the maximum amount of time or idle time to expire flow entries.
- Cookies: opaque data value chosen by the controller. Controller can use it to filter flow statistics, flow modification and flow deletion.

Table 2-4 Flow entry in flow table

Match Fields	*Priority*	*Counters*	*Instructions*	*Timeouts*	*Cookies*

A flow entry to the flow table is identified by match field and priority, where the match field and priority taken together in order to identify a unique flow entry in the flow table.

The first widely implemented of OpenFlow was version 1.0, released on 2009. It was based on a single flow table and fixed 12 tuples for matching flows. OpenFlow version 1.1, released in 2011, supported multi flow table, VLAN and MPLS protocol. The disadvantage of this version is the incompatibility with version 1.0. Version 1.2 was released in late 2011, and included and supported IPv6 protocol, flexible flow match and flexible rewriting of packet header. Version 1.3 was released on 2012. It supports flexible measurement capabilities, Provider Backbone Bridge (PBB), event filters and meters. Version 1.4 provides some new advantage including bundle message, optical ports support, flow monitoring and eviction. The latest version is OpenFlow 1.5 (2014), which improved the extensibility of the protocol.

There are a couple new features of version 1.5 that provide a significant advantage over the past versions, for the purpose of our research. Namely, packet registers and port recirculation [60]. Packet registers use set-field and copy-field actions to store temporary values and information alongside the packet through pipeline processing. Switches can support as many registers as required. This feature is important because it can be critical for the tight timing characteristics (synchronization) of any reconfiguration via the DBA algorithm. Regarding port recirculation, it is used to allow a logical port to be associated with another logical port to process packets (packets are recirculated between logical ports). This function can be used to interact with the DBA for calculation operations.

2.6 SDN in Optical Access Networks

SDN is possibly the biggest shift in the networking field in the last 30 years. The application of SDN in optical access is being now intensively investigated by researchers, in order to control and manage network precisely and dynamically. Consequently, to effectively extend the SDN paradigm to optical access networks, the dynamic resource management and virtualization of SDN need to be preserved, while taking into account the physical connectivity restrictions, fundamental optical channel impairments and legacy technology limitation that arise. Applying the SDN paradigm to multi-service optical access must focus on the edge point of the network such as centralized control protocols at OLT, ONU and reconfigurable network applications for both OLT and ONU.

Several works have already addressed the topic of how to apply the SDN architecture in PON networks, but the idea to make easier the deployment of a SDN-based services and the interoperability of different protocols in optical networks remains not explored in detail. [61] is a good survey of the most recent efforts in the field of SDN applied to optical access networks, classifying them into the categories of control layer, infrastructure layer, application layer, and orchestration of multi-layer and multi-domain networking. [62] and [63] provide a single unified control plane over packet and optical networks. The proposed techniques provide a simplified control model, but requires optical switching devices to be upgraded to support control plane extension. In [64] and [65], authors propose to use a virtual switch on each optical switching node to obtain an unified control plane. Each physical interface of an optical switching node is mapped to a virtual interface. Messages between the controller and the virtual switch are converted to commands acceptable by the optical switching devices. During deployment, there will be an extra layer to bridge controllers and legacy switches [66]. These approaches can reuse legacy network equipment, through this would cause extra communication latency by using message proxying.

The discussion of how SDN principles might be applied to optical access and mobile backhaul networks is the focus of [67]. Regarding PONs, Software-Defined OLTs and ONUs are proposed, though few details are given. A "meta-MAC" algorithm (able to coordinate the medium access control algorithms of different PON technologies and/or physical layers) is presented as one of the applications of a centralized, SDN-based management of the network.

GPON (Gigabit Passive Optical Networks) are the focus of [68], where an extension of OpenFlow is proposed in order to provide traffic mapping and forwarding capabilities, while keeping its original functionality, architecture and capabilities. Accordingly, each OLT and Optical Network Terminal (ONT) in the GPON contains flow tables and communicates over a secure channel with the remote controller via the OpenFlowPLUS protocol. Three new actions are proposed, all of them related to the mapping of Ethernet frames to GPON logical ports, containers, and framing. The considered actions are illustrated in Table 2-5. Other functionalities of the OLT, such as ONT discovery, synchronization, or bandwidth assignment are not tackled.

Table 2-5 Main GPON forwarding functions provided by OpenFlowPLUS

GPON unit	**Action type/Action**	**Remarks**
ONT, OLT	*Gpon: Map to GEM port*	*Introduction of a new action to the original OpenFlow action set* *Function: mapping Ethernet frames to particular GEM port instance*
ONT	*Gpon: Map to T-CONT*	*Introduction of a new action to the original OpenFlow action set* *Function: Mapping GEPON Encapsulation Method (GEM) port to particular T-CONT instance*
ONT, OLT	*Output*	*Action modification when executed for GPON interface* *New function: GEPON Transmission Convergence (GTC) framing before forwarding the packet on the GPON port*

An integrated management architecture for SDN-based access, metro and core networks is proposed in [69]. The access segment is enhanced with OpenFlow-enabled OLTs that pass bandwidth assignment information to the Metro segment (an optical ring of OpenFlow devices) and the Core segment (an IP/MPLS network with path computation network element (PCE)). A so-called Generalized SDN controller is responsible for the control planes of the global network and is potentially able to coordinate and optimize the end-to-end performance.

The authors of [70] propose a software-defined flexible and efficient passive optical network for intra-datacenter communication. They apply SDN to PON to provide high flexibility during the data transmission process. Software-defined technology enables network to dynamically configure the connections between OLT and ONU. A scheme based on Network Coding (NC) is presented to flexibly implement the coding operation on local peer traffic and keep the compatibility with current Multipoint MAC control sublayer in a single OLT. The NC technique increases the total throughput and efficiency of the system, boosting downstream bandwidth up to 50%. Their DBA scheme realizes an efficient scheduling and resource allocation in time domain, together with an ONU grouping algorithm to reach to best resource assignment in the wavelength domain. It allows the network to configure the connection and assign policy based on traffic status. The centralized controller is responsible of the dynamic configuration.

[71] Presents a control plane architecture for SDN-driven converged metro-access networks. The controller is based on the SDN targeted fast feeder fiber protection over dual-homed passive optical network system, and dynamic service provisioning over a multi-wavelength PON. The main task of the SDN-based controller in the access network is to maintain communication with the network orchestrator, export an abstracted network view to the controllers, ensure the QoS of individual or aggregate flows, handle network capacity management within the access aggregation network and translate the abstract link parameters into appropriate commands for the southbound interface.

Yang [72] proposed a new software-defined optical access network (SDOAN) architecture of remote unified control for OpenFlow-based software-defined PON. A service-aware flow scheduling strategy (SA-FS) is introduced to efficiently and flexibly allocate bandwidth resources of the network, and detect the network flows status in real time. This novel architecture improves the QoS guarantee and utilizes the optical network resource of each user through the controller minimizing the operational expenses via remote interaction and operation.

[73] Proposes a software-defined edge network architecture (SDEN) to operate on top of PON networks, able to integrate with the core SDN network to enable flow control functions and end-to-end manageability via a SDN controller. An agent of SDEN is

designed in the edge network to translate high-level flow control request of the controller to a set of PON configuration commands and send them to the OLT via the command line interface. [1] presents a novel EPON architecture based on the SDN and NFV (Network Functions Virtualization) paradigms, where OLT and ONUs are partially virtualized and migrated to the centralized controller to reduce OPEX, CAPEX and complexity of EPON operations. The authors split some functionalities of MultiPoint MAC control sublayer between centralized controller and a modified OLT (SDN-OLT).

The authors of [74] propose a software-defined EPON architecture to replace the hardware-based DBA with a re-programmable DBA and a bandwidth allocation scheme to manage the DBA module through the controller, although low-level details are not provided. Compared to traditional EPON networks, the scheme shows less traffic delay and better support to traffic differentiation. However, the improvement is not significant.

A new architecture known as Central Office Re-architected as a Datacenter (CORD) has been introduced [75] to replace the hardware found in telephone exchanges into software-based equipment, effectively turning central offices into datacenters. CORD opens a way to unify the SDN, NFV, and cloud concepts to support cloud-based and virtualization techniques in the data and control planes of the central offices and enable fast and flexible deployments of new services. CORD follows the Everything-as-a-Service (XaaS) paradigm by decoupling the control and data planes and running Virtual Machines (VMs) on top of the Open Network Operating System (ONOS) [76]. ONOS is an SDN operating system that manages the (softwarized) physical switching fabric and storage resources to support CORD infrastructure and network virtualization. As of 2017, major service providers (including AT&T, NTT Communications and Telefonica) support CORD.

Finally [77] is, to the best of our knowledge, the only work that tackled the application of the SDN paradigm to the SIEPON standard, before this thesis was written. The author presented an invention that provide efficient ways to operate network nodes in a heterogeneous management environment. It permits nodes to interoperate with an existing management protocol and with other protocols, including a control plane and data plane interface for supporting SDN. Our contributions described in [2] and this thesis are aligned, and we go farther by providing a complete architectural solution, low-level implementation details, a functional validation and performance measurements.

The following table summarizes the main proposals:

	Approach
SDN Control layer	*[62] [63] Provides a single unified control plane over packet and optical networks.*
	[68] Defines OpenFlowPLUS to extend OpenFlow southbound interface for GPON.
	[69] Presents an integrated management architecture for SDN-based access, metro and core networks.
	[71] Develops a control plane architecture for SDN-driven converged metro-access networks.
	[72] Proposes a SDOAN architecture of remote unified control based on OpenFlow for PON networks.
	[1] [75] An SDN controlled PON is created to control OLT via OpenFlow Southbound Interface.
SDN Infrastructure layer	*[64] [65] Presents a virtual switch for the optical switching nodes.*
	[67] Proposes a hierarchical flexi-grid infrastructure for multi-service broadband optical access.
	[70] Develops components for SDN control of a PON in a data center.
	[73] Develops a module for mapping OpenFlow flow control requests into PON configuration commands.
	[1] [75] Upgrade OLT to SDN-OLT by proposing a virtual OpenFlow switch as the base for building the OLT.
	[77] Provides SDN-based management system to operate existing network nodes with the heterogeneous management system.
Applications	*[69] Presents an SDN based routing application within I2RS framework.*
	[77] Presents an application of the SDN paradigm to the SIEPON standard.
Virtualization	*[1] Presents a scheme to partially virtualized MPMC sublayer of the OLT into the SDN-controller.*
	[74] Replaces the hardware-based DBA with a re-programmable DBA to manage it through the controller.

2.7 Summary

This chapter has presented the fundamentals of the architecture of EPON broadband access networks. This architecture uses TDM as medium access control MAC to deliver data encapsulated in Ethernet packets from the ONUs to the OLT. We addressed the MPMC sublayer functionality in terms of controlling the data transmission in the upstream channel, which is performed by GATE and REPORT messages. We have reviewed several proposals on resource management and energy saving, and summarized the most relevant characteristics.

We also presented the SIEPON standard, which provides standardization for MAC layers and upper layers of the EPON network in order to support functions related to the data path, such as QoS. It introduces ESPs as a conceptual path of a frame through the functional blocks of the MAC client to achieve provisioning, connectivity and QoS.

Finally, we studied the state of the art of Software Defined Networking (SDN) and OpenFlow protocol and the advantage of introducing them in optical access networks. Most proposals have targeted either the simplifying of the PON network or increasing the manageability and QoS of the network.

3 SDN-BASED SIEPON ARCHITECTURE

3.1 Introduction

The main challenges faced by current EPON networks are how to optimize the performance, and how to improve the flexibility and efficiency of network operations. Given the centralized nature of PONs, the OLT is a key element in order to simplify the management and reducing the complexity of the optical network. The SIEPON standard was created to provide a flexible architecture for PON networks by defining functions for MAC control, and OAM. It also intended to realize service interoperability among 1G-EPON and 10G-EPON equipment, and finally it also focused on the development of unified PON data path architecture and interoperability of the OLT and the subtended ONUs.

Software Defined Networking (SDN) has gained a lot of attention in recent years, because it addresses the lack of programmability in legacy networking architectures and enables easier and faster network innovation. Network administrators can manage services through the abstraction of lower level functionalities, by decoupling the system that makes decisions about where traffic is sent (the control plane) from the underlying systems that forward traffic to the selected destination (the data plane). Moving the control plane out of the data plane elements enables both planes to function independently, thus introducing many advantages such as: programmability; global optimization of network operations (such as load balancing, for example); high flexibility; and the possibility of realizing a centralized network view (and thus opening the way to the optimization of network operations).

In this chapter, we present a novel architecture based on SDN concept for EPON networks that includes the Service Interoperability in Ethernet Passive Optical Networks (SIEPON) standard. This method is intended to minimize the operational and capital expenses, while optimizing dynamically the use of resources. To do so, we have defined a new Multipoint MAC Control (MPMC) sublayer extension based on the OpenFlow protocol with splitting some of its functionalities between SDN-EPON controller and SDN-OLT. We have virtualized the main functions of the MAC client of SIEPON's OLT by migrating to the SDN-EPON controller, and keeping its functional block in the SDN-OLT.

The simulation results demonstrate the effectiveness of the new architecture, while keeping similar or improved performance compared to legacy PONs, in term of delay and throughput.

The chapter is organized as follows. Section 2 defines the novel SDN-based SIEPON architecture, SDN-OLT functionalities, and implementation. Section 3 presents the principle operation and operation details. Section 4 discusses the results of the proposed scheme. Section 5 concludes the chapter.

3.2 SDN Architecture for SIEPON

This section presents our novel architecture based on the SDN-OpenFlow principle for EPON networks. To implement our architecture an SDN controller is designed to effectively manage and enhance the resource utilization, flow monitoring, bandwidth assignment, quality-of-service (QoS) guarantees, and energy management of optical network access for each ONU and end user in EPON networks. In addition, we defined an OpenFlow switch (OF-switch) with the capability to manage dialogues between SDN-OLT elements to emulate the forwarding functions and execute numerical calculations by interacting with the DBA. The OpenFlow switch is extended with synchronous ports to retain the time-critical nature of the EPON network. OpenFlow messages are also extended with new functionalities to implement the concept of EPON Service Paths (ESPs).

The main concept behind our approach is to build an OLT around an OpenFlow switch. This novel architecture is based on five principles, which are developed in the following subsections:

- Split EPON functionality into virtualizable and non-virtualizable parts.
- Extend OF-switch with synchronous ports.
- Extend OF messages with new functionalities.
- Implement ESP functional block via OF messages.
- Open EPON management to multi-tenant and multi-provide scenarios.

3.2.1 Split EPON Functionality into Virtualizable and non-Virtualizable Parts

From the experience gained in [1], one of the problems that must be resolved is how to split into two parts the functionality of the Multipoint MAC Control (MPMC) sublayer of the OLT, which defines the MAC control operation for optical point-to-multipoint networks. One part must be migrated to the centralized SDN-EPON controller, and the other part remains at the SDN-OLT. As illustrated in Figure 3-1, the SDN-based SIEPON architecture for EPON networks is divided into three main parts: SDN-OLT; SDN-EPON controller; and ONUs. In our proposal, ONUs are not modified by the SDN approach.

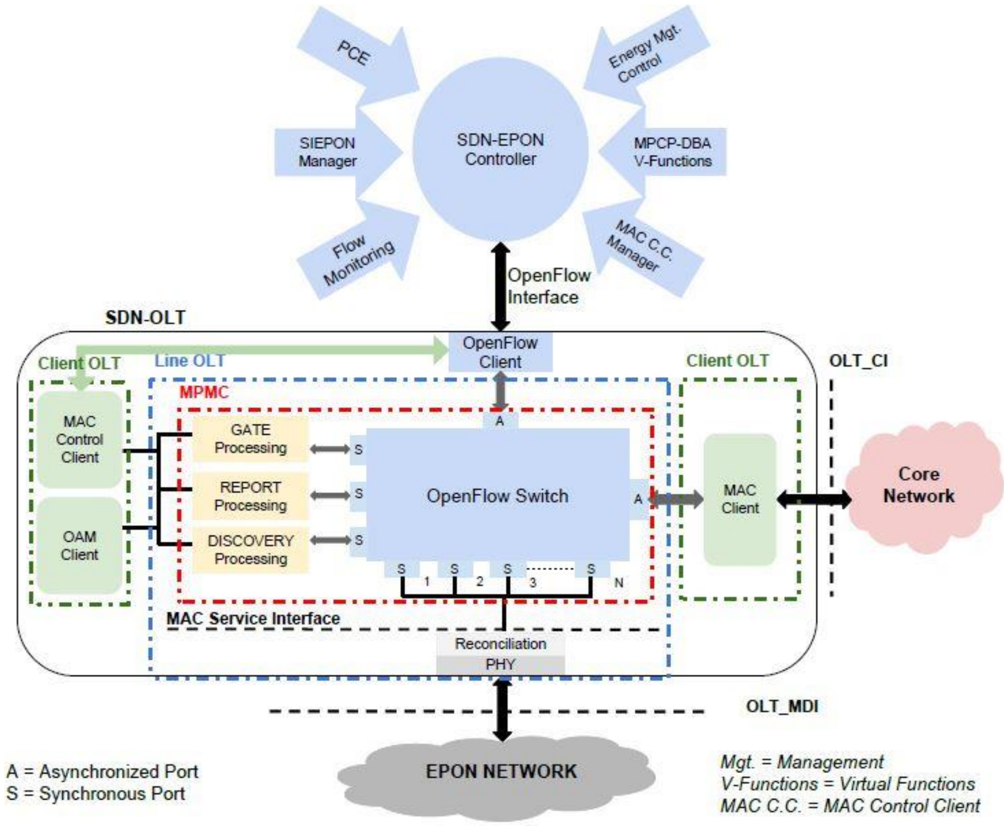


Figure 3-1 SDN-enabled SIEPON architecture

3.2.1.1 SDN-OLT

The SDN-OLT is the main element for handling and forwarding packets. It is located between the SDN-EPON controller, core network and the EPON network (ONUs). In this architecture, we designed the SDN-OLT by implementing a virtualized internal switch (OF-switch) that emulates the forwarding functions of the MPMC sublayer and the functionalities and the operations of Control Multiplexer, Control Parser, and Multipoint Transmission Control elements. This emulation is performed with a set of synchronous and asynchronous ports of the (virtualized) internal switch. The synchronous logical ports are designed to connect with the ONUs and internal processes of the SDN-OLT (such as gate, report, discovery processing and DBA). The asynchronous logical ports are designed to connect with the controller via the OF-client to receive OF-messages, and with a MAC client connected with the core network. The number of synchronous logical ports associated with ONUs vary from 1 to 64 + 1, where the 64 ports[2] are unicast and dedicated to each of 64 possible connected ONUs, and the extra port is a broadcast port associated with all available ONUs. All 64 logical ports and their corresponding 64 MPMC instances are created in the SDN-OLT and set to sleep mode. For newly connected ONUs, the controller is responsible for activating a new instance with three associated logical ports to connect to the respective discovery, gate, and report processing entities. Simultaneously, the controller activates a unicast MAC service logical port and associates it with the newly activated instance. Figure 3-1 illustrates the SDN-OLT architecture with the OF-switch embedded in the MPMC functional block, and more generally embedded in L-OLT - with interfaces to the C-OLT (MAC control client, MAC client and OAM client), and OLT_MDI (Medium dependent Interface).

Due to manageability property of the MPMC, SDN-OLT requires a proper processing mechanism to efficiently coordinate the data transmission among the other entities, and regulate the transmitting of MPCP instances. Instead of defining several distributed OF-switches for several MPMC instances, we have defined a centralized intelligence OF-switch for several instances to simplify the operation, functionalities and timing of the OLT. All instances are coordinated via this centralized OF-switch and it is the only element in SDN-OLT that dialogues with the DBA for coordinating the forwarding functions.

We have defined a set of instance_ID (up to 64) in the OF-switch to allow it to coordinate and communicate with different instances at a specified time. The OF-switch changes its own instance_ID when it receives a signal from an active instance, and then playing as a part of that instance. This architecture allows OF-switch to process all operation of active instances in coordination with DBA in lower time interval. Migration from multiple complex instances with several coordinators in legacy OLT to multiple instances with a single simplified coordinator in SDN-OLT[3] are illustrated in Figure 3-2 (Figure's elements will be discussed later in Appendix A).

---

[2] The maximum number of ONUs supported in an EPON is 64.

[3] A single OF-SW is substituted with a MPTC and several CPs, CMs of multiple instances.

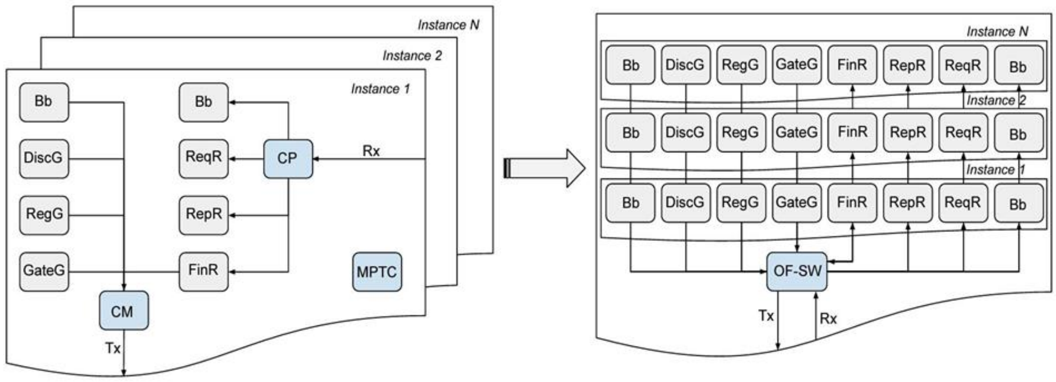


Figure 3-2 Left architecture of a legacy OLT with several complex MPMC instances. Right: the simplified SDN-OLT architecture. Acronyms of the processing entities: OF-SW (OpenFlow switch); Bb (backbone), DiscG (discovery gate generation); RegG (register generation); GateG (gate generation); FinR (Final Registration); RepR (report reception); and ReqR (request reception).

3.2.1.2 SDN-EPON Controller

The SDN-EPON controller is the intelligent part of the system that dialogues with the SDN-OLT and can be run either in a datacenter of the access provider, or in a dedicated server located within the network backbone. The SDN-EPON controller has a centralized view of the network and is responsible for configuring: MAC control client parameters (for example, the Logical Link Identifier (LLID)); MAC client policies; SIEPON control management; energy management control; QoS; linking the high-level QoS of SIEPON with the low-level MAC QoS management; Path Computation Element (PCE); and monitoring the flow statistics.

The ESP management module of the MAC client is virtualized and migrated to the SDN-EPON controller as a module of the SIEPON manager. This module provides the basic functional blocks of the MAC client model and offers the services of provisioning, QoS, flow management, bandwidth assignment policy, timing decision, adding unregistered ONUs to the network, and data forwarding services in the SDN-OLT. But the main functional block of the MAC client remains in the SDN-OLT and is managed by the flow tables of the OF-switch.

The MAC control client manager of the controller is responsible for controlling and configuring the initial parameters of the MAC control client; as well as gate, report, and discovery processing. Thus, upon connecting to the controller through the OF-client, the SDN-OLT receives the initial values for each entity from the virtual MAC control client of the controller. It also provides a database for the OF-switch to keep information about LLIDs, logical port status (including its speed, 1Gbps/10Gbps), and MAC address of the ONU to configure the switch flow tables. The virtual MPCP-DBA module manages the general policy of the network and enables the controller to dialog with the SDN-OLT to change the DBA parameters and policies, change the DBA priority for each ONU, or switch between multiple DBA algorithms, to name a few possibilities.

In summary, our proposal separates the functions that can be virtualized at the controller (those that work at long timescales) from those that must remain at the switch because their operation is synchronous (and they work at short timescales).

3.2.2 Extend OF-switch with Synchronous Ports

Being EPON a synchronous network, the SDN-OLT must manage time-critical messages delivered to synchronous ports. In each of these ports, the transmitted or received packets are processed in specific times determined by the synchronous flow entries. To replace the Control Multiplexer, Control Parser, and Multipoint Transmission Control elements of the OLT with the OF-switch, we extended the MPMC sublayer of the OF-switch that maintains the synchronous nature of the EPON network. Frame forwarding from the MAC control and MAC client to the MAC service interface in the downstream direction must be synchronous.

The OF-switch synchronizes a newly added ONU with the network by:

- Setting an instance_ID in the OF-switch for the respective MPMC instance
- Assigning three synchronous logical ports for each connected instance of a gate, report, and discovery process.
- Assigning an input/output synchronous logical port with its respective LLID to link with the MAC address of the connected ONU.
- Timestamping out-going packets sent to the ONUs.
- Calculating the round-trip-time of incoming packets from the ONUs.
- Allocating bandwidth and transmission time slots to allow ONU to transmit data within a specified time slot in the upstream direction (packet forwarding strategy) with the cooperation of the SIEPON manager (according to its QoS policies).

- Creating and destroying flow entries in the OF-switch, thus assigning bandwidth on demand.

The OF-switch uses its local clock to write the timestamp for every out-going packet at the logical synchronous port. Synchronization between SDN-OLT and ONUs is kept using a register and port recirculation (a functionality of OpenFlow v1.5) which interact with the DBA for calculating the RTT of incoming packets.

Additionally, the OF-switch synchronizes the MPMC instances with the DBA to enable them to transmit packets on the downstream channel. For that, we have defined three registers P (transmitPending), I (transmitInProgress) and E (transmitEnable). Based on these, the OF-switch can change the transmission state of the packets and enable only one of the available packets to be transmitted through the associated MAC service interface to the Reconciliation Sublayer (RS) by enabling one transmitEnable signal at a time. Figure 3-3 illustrates the extension of the MPMC sublayer of the SDN-OLT.

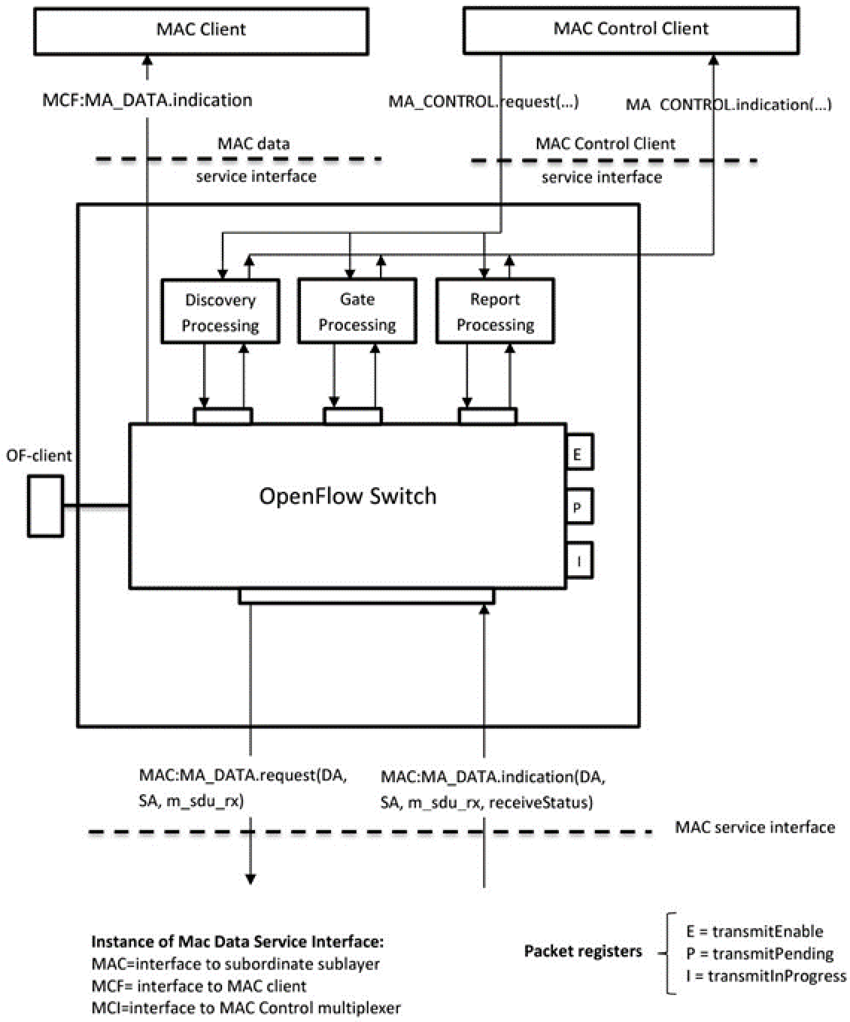


Figure 3-3 Standardized Multipoint MAC Control functional block

3.2.3 Extend OF Messages with New Functionalities

We now describe how the concept of ESP can be implemented in SDN. We extended OpenFlow 1.5 with a new set of messages:

- The ***OFPT_FEATURES_REPLY*** message is extended to respond to the query of controller whether the switch is an SDN-OLT switch or not. Table 3-1 shows the extension structure of OFPT_FEATURES_REPLY of OF-message.
- The ***OFPT_PORT_STATUS*** message is extended to carry the assigned LLID of the port with the respective MAC address of the ONU to the controller.
- The ***OFPT_SET_CONFIG*** message is extended to configure the initial parameters of the MAC control client and MAC client at the beginning of the dialogue between the controller and the SDN-OLT. For instance, it can assign a set of 64 LLID to the MAC control client and configure the range of LLIDs.
- Additionally, we propose an extension of the ***OFPT_FLOW_MOD*** message with a new set of synchronous rules, matching fields, and actions to perform the main functionalities and operations of the legacy EPON - and to implement the ESP functionalities of SIEPON (explained in detail in Section 3.2.4).

Table 3-1 OpenFlow features reply structure extension

```
/*Switch features.*/
struct ofp_switch_features {
  struct ofp_header header;
  uint64_t datapath_id;        /*Datapath unique ID. The lower 48-bits
                                 are for a MAC address, while the upper
                                 16-bits are implementer-defined.*/
  uint32_t n_buffers;          /*Max packets buffered at once.*/
  uint8_t n_tables;            /*Number of tables supported by datapath.*/
  uint8_t auxiliary_id;        /*Identify auxiliary connections.*/
  uint8_t pad[2];              /* Align to 64-bits.*/

  /*Features.*/
  uint32_t capabilities;       /*Bitmap of support "ofp_capabilities".*/
  uint8_t sdn-olt              /*Switch definitions.*/
  uint24_t reserved;
};
OFP_ASSERT(sizeof(struct ofp_switch_features) == 32);
```

We have also extended the OpenFlow match fields from 45 to 49 as shown in Table 3-2:

- ***PDU type (45)*** indicates the type of incoming packet (whether it is a MAC control frame or a data frame).
- ***Opcode (46)*** indicates the type of incoming packet (whether it is a DISCOVERY GATE, REGISTER_REQ, REGISTER, REGISTER_ACK, GATE or REPORT).

- ***Flag (47)*** represents the type of GATE packet (whether it is a DISCOVERY GATE (flag = 1) or a normal GATE (flag = 0)).
- ***LLID (48)*** holds the LLID number of the port and is defined by the controller and assigned by the MAC Control Client.
- ***Grants number (49)*** represents the number of grants issued by the GATE packet. The number of grants varies from 0 to 4, where 0 is a periodic GATE packet and 1 to 4 are related to the bandwidth assignment to a normal GATE.

Table 3-2 Match fields extension for OpenFlow 1.5

```
/*OXM FLOW match field types for OpenFlow basic class*/
enum oxm_ofb_match_fields {
  OFPXMT_OFB_EPON_PDU_TYPE      = 45, /*MPCPDU packet type
                                      (MAC Control type)*/
  OFPXMT_OFB_EPON_OPCODE        = 46, /*Packet type*/
  OFPXMT_OFB_EPON_FLAG          = 47, /*GATE/DISCOVERY GATE flag*/
  OFPXMT_OFB_EPON_LLID          = 48, /*Port LLID*/
  OFPXMT_OFB_EPON_GRANTS_NUMBER = 49, /*GATE grants number*/
};
```

To carry out the calculations and functionalities performed by legacy OLTs and those operations that need to interact with the DBA, we use two new features of OpenFlow version 1.5: packet registers[4] and port recirculation[5]. In this architecture, we have implemented five packet registers for RTT calculation and writing timestamps; and the last three packet registers (P, I, and E) are used to emulate the Multipoint Transmission Control. Table 3-3 shows the five extended action sets of OpenFlow 1.5 (30-34). These extended action sets for EPON packets are described as follows:

- ***TransmitPending (30):*** modifies the value of the header field of the packet (from 0 to 1) interacting with the packet register, changing the transmission state of the packet to TransmitPending.
- ***TransmitInProgress (31):*** Same as the previous case, changing the transmission state of the packet to TransmitInProgress.
- ***TransmitEnable (32):*** same as above, changing the transmission state of the packet to TransmitEnable.

[4] Packet registers are used to store temporary value and information with set-field and copy-field actions.

[5] Port recirculation allows a logical port to be associated with another logical port to process packets (packets are recirculated between logical ports).

- ***RTT calculation (33):*** calculates the round-trip time of incoming packets via port recirculation and packet registers, interacting with the DBA.
- ***Timestamp (34):*** writes the timestamp through the packet register, with the local time of the SDN-OLT once the packet transmission is enabled.

Table 3-3 Actions extension for OpenFlow 1.5

```
/*Action type for OpenFlow basic class*/
enum ofp_action_type {
  OFPAT_SET_TRANSMIT_PENDING     = 30, /*Set transmission to pending*/
  OFPAT_SET_TRANSMIT_IN_PROGRESS = 31, /*Set transmission to in progress*/
  OFPAT_SET_TRANSMIT_ENABLE      = 32, /*Set transmission to enable*/
  OFPAT_SET_RTT                  = 33, /*Calculate RTT of packet
                                       (Port recirculation)*/
  OFPAT_COPY_TIMESTAMP           = 34, /*Copy local time of SDN-OLT*/
};
```

3.2.4 Implement ESP Functional block via OF messages

As we mentioned in subsection 4.2.3, our extension of the Flow-Mod message is also used to implement the ESP functional blocks of SIEPON in the SDN-OLT. Figure 3-4 shows an example of the OF flow entry with the extended and existing matching fields[6]. The main actions in our proposal that correspond to the ESP functional blocks are:

- ***Set-Field:*** corresponds to the *Modifier block [M]* and applies all set-field actions to the packet. It modifies the value of header fields in the packet (example, set Ethernet/IPV4/IPV6 source address, destination address, VLAN ID and VLAN priority).
- ***Apply meter:*** corresponds to the *Policer/Shaper block [PS]* and enables the measurement and control of the rate of packets (rate-limiter).
- ***Set-Queue:*** corresponds to the *Queues block [Q]* that sets the queue_id for a packet when sent to the port. It is used for scheduling and forwarding the packet.
- ***Output port:*** corresponds to the *Output block [O]*. Packets are assigned to a physical port (controller) or a logical port (unicast port/broadcast port/an entity port of instances).

---

[6] The existing match fields together with the extensions and priority are used to classify the incoming packets and determine which action must be executed.

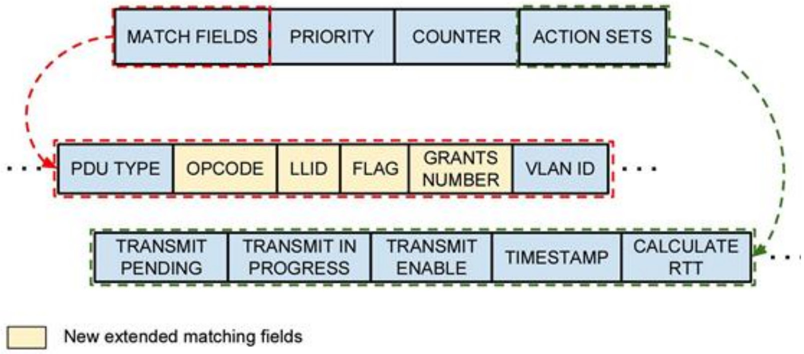


Figure 3-4 The example of some extended matching fields.

Note that in this scheme the *Scheduler block [S]* is executing three actions (transmitPending, transmitInProgress, and transmitEnable) and changing their associated registers (P, I and E) before executing the output action. These three actions must interact with a predefined scheduler algorithm through their registers in order to schedule and forward synchronous packets to the downstream channel at a precise time.

Operations performed by the *Classifier [C]* and *Cross-connect [X]* blocks are executed via rule matching and actions in the flow table. Incoming packets are compared and classified based on their matching fields with the existing rules, and later packets are sent to the destination based on the associated action. The ESP functional blocks of the OF-switch are shown in Figure 3-5.

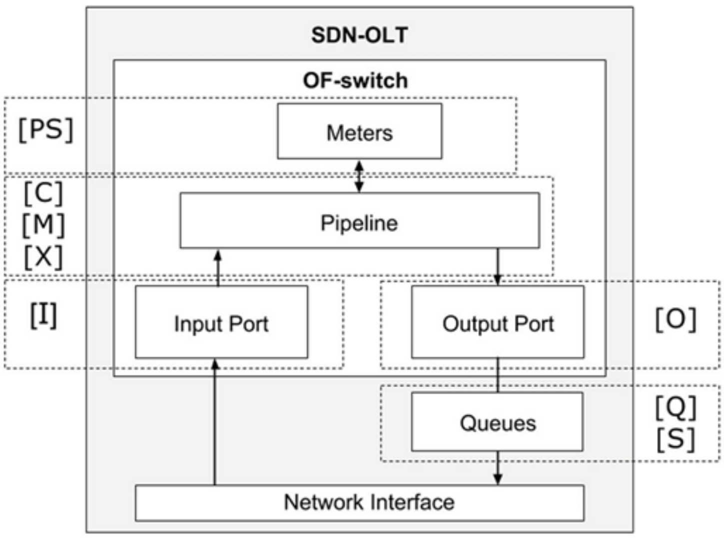


Figure 3-5 ESP functional block in the OF-switch.

3.2.5 Open EPON Management to Multi-tenant and Multi-Provider Scenarios

The proposed SDN-based SIEPON architecture allows EPON networks to be shared by several infrastructure and service providers and to virtualize and centralize the control functions. This virtualization of the underlying optical network enables operators to provide Service Level Agreement (SLA) and QoS requirements. In this scenario, the SDN-EPON controller manages the allocated virtual network resources. Figure 3-6 illustrates a scenario with multiple service and infrastructure providers with different SDN-EPON networks, which are managed by different controllers. The controllers of each service provider offer various on-demand services to the users over an infrastructure that does not belong to a single infrastructure provider. For instance, service provider 1 (SP 1)may offer an IPTV service over the SP 1 controller while service provider 2 (SP 2) may offer a VoIP service over the SP 2 controller.

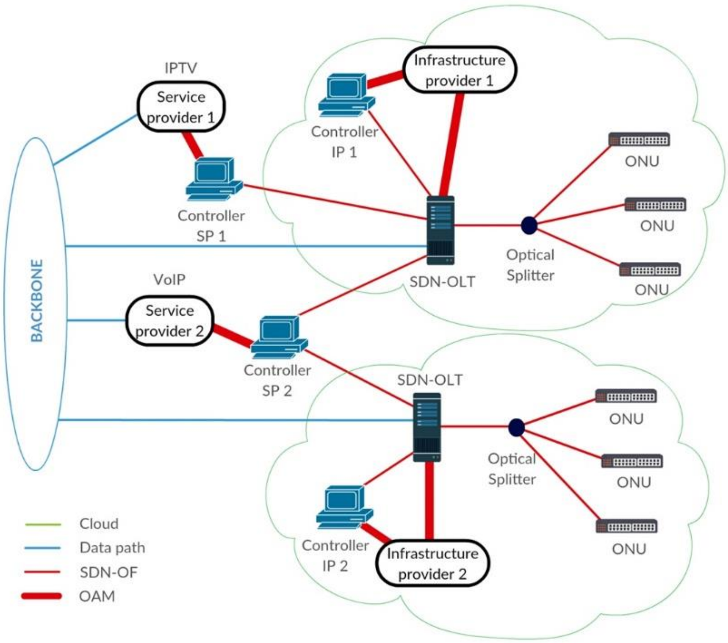


Figure 3-6 SDN-EPON scenario with multiple service providers.

The SDN-EPON controller can be used to configure virtual networks according to the user demands and deliver virtual network services. Based on existing service providers in the network, controllers can define new rules to optimize resource allocation, service availability, and enhance QoS guarantees for SDN-OLT devices and each end user/ONU. To implement a level of QoS mechanism for each user, we follow a multi-tenant approach because the system is dynamic and the infrastructure providers can offer the services to multiple users. We implement those services using the OpenFlow meters to share the access to the common transmission channel with specific privileges depending on the specific software instance. OpenFlow enables meters to measure the rate of packets assigned to each flow and apply QoS policies (such as rate limiting) and DiffServ-like policing. Additionally, meters can assist per-port queues to schedule packets on an output port according to their priority, thus ensuring QoS. Therefore, with OpenFlow meters we can share the infrastructure among multiple tenants by defining different access and control privileges.

3.3 Operation Principles

This section provides more details about the operation of the Multipoint MAC Control (MPMC), and an example to illustrate the procedure for generating a new flow entry in response to a GATE message in the SDN-OLT.

3.3.1 Implementation of the MPMC

In our proposal, the MPMC sublayer is simpler than that found in the legacy OLT, where an MPTC module coordinates several MPMC instances (one per ONU, each one including a Control Multiplexer (CM) and a Control Parser (CP) entity). We simplified the design by defining three single MPTC, CM, and CP entities that are implemented as a set of rules and flow entries in the OF switch. The switch is responsible for communicating with the multiple MAC instances in the OLT to handle their packet processing (send/receive) requests.

As shown in Figure 3-7, the OF-switch consist of three main parts: 1) OpenFlow agent; 2) control path; and 3) data path. The OpenFlow agent is responsible for communicating with the SDN controller in order to manage the control and data paths. The control path coordinates the operations (mimicking the functionality of the legacy OLT) through the flow entries in the flow table of the OpenFlow switch. The data path is responsible for matching the incoming packets according to the matching rules, and executing the associated actions (such as forwarding the frame and scheduling transmission).

The OF-switch in the MPMC sublayer performs the following functionalities: when a frame is received from the underlying MAC, it is forwarded to the flow table of the OF-switch, where the frame is parsed according to its opcode (see Section 3.2.3 for more detail). The REPORT and data frames that come from the ONUs are processed by the CP-related functionalities, implemented in the control plane and their associated actions in the flow entries of the flow table (shown in orange in Figure 3-7). In addition, the new MPMC is responsible for arbitrating the processing of the GATE, DISCOVERY GATE, and data frames generated by each MAC instance and forwarding them to the RS layer (shown in blue in Figure 3-7). Finally, the MPMC coordinates the transmission of frames via the three registries I, P, and E defined in the control path of the OF-switch via a dialogue with the DBA agent (shown in green in Figure 3-7, see section 3.1.3 for more detail).

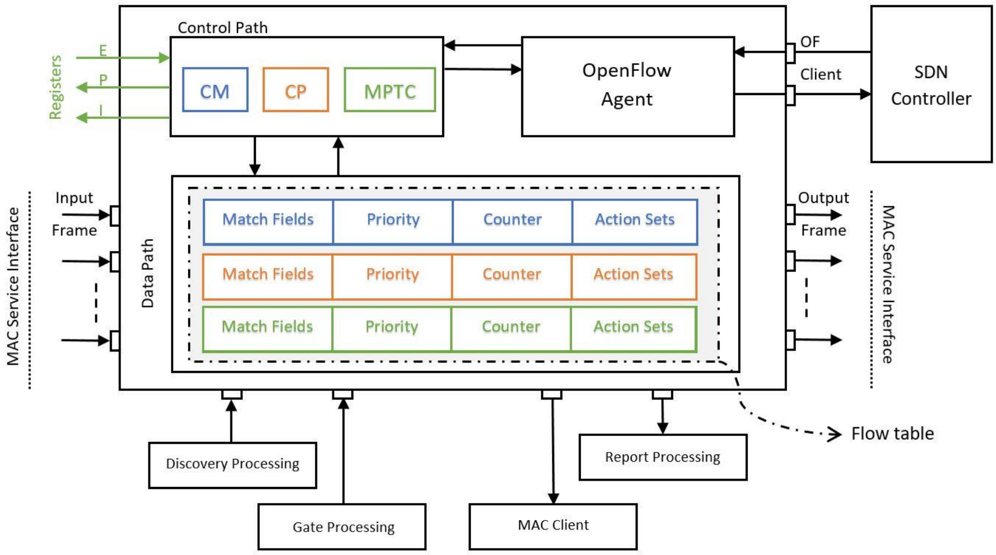


Figure 3-7 Architecture of the OF-switch implemented in the MPMC sublayer.

3.3.2 Example of the Operations Involved in the Generation of a Flow Entry

We now provide an example to illustrate the procedure for generating a new flow entry in the SDN-OLT for processing a DISCOVERY GATE, GATE and REPORT messages. This is an example of the operations involved in the generation of a flow entry.

Let’s recall that a GATE message tells a specific ONU when to send data (start time) and how much data to send, in response to a previous REPORT message from the ONU requesting a transmission opportunity. Regarding the discovery process, DISCOVERY GATE messages are broadcasted periodically to all ONUs, and are replied by the newly connected ONU with a REGISTER REQUEST (randomly delayed in order to minimize collision with other ONUs). The OLT confirms by sending a REGISTER message immediately followed by a GATE that grants and schedules an ONU response in the form of a REGISTER-ACK message, finishing the discovery and register process. Figure 8 illustrates the state diagram for the actions related to processing the aforementioned messages.

When the SDN-OLT establishes the communication with the SDN-EPON controller, the latter sends a series of FlowMod messages with pre-defined rules to be inserted in the flow table of the OpenFlow switch. These rules define how to process the DISCOVERY GATE, REGISTER_REQ, REGISTER, REGISTER_ACK, GATE, and REPORT messages. MAC control client sends a request to a discovery processing entity of an instance to creates a DISCOVERY GATE message with the required fields (discovery information and policy to use for the ONU) and sends it through the appropriate synchronous logical OF switch port. After receiving a packet, a flow table lookup is performed by comparing the packet header fields with the matching fields of the installed rules by observing a priority order. When the packet is matched, the attributed counter for the chosen flow entry is increased and the actions linked with the flow entry are performed.

When, later on, the controller receives a request (through Packet-In) from the SDN-OLT, for processing a GATE message, it gathers and analyses the received packet data (source and destination MAC addresses, PDU type, opcode, flag, grant number, LLID and VLAN IDs) to set a rule in the OF switch for determining the grants' number and establish the QoS policy, and prioritize packet delivery toward an output port. The rule is transmitted through a FlowMod and a Packet-Out message to the OpenFlow switch, and the received GATE message is processed accordingly.

In case of the DISCOVERY GATE (left side of Figure 3-10) and the GATE (centre of Figure 3-10) messages, once the packet is ready to be transmitted, the action of TransmitPending is executed on the packet by interacting with the register "P" to change the transmission state of the packet to "TransmitPending" (setting the flag from 0 to 1). Note that the packet is waiting in the queue until higher priority packets are processed by the DBA. The action of TransmitInProgress is executed by interacting with the register "I" to change the transmission state of the packet to "TransmitInProgress" (setting the flag from 0 to 1 and changing the TransmitPending flag from 1 to 0). The action of TransmitEnable is then executed when there are no more packets to transmit by interacting with register "E", and when the transmission state of the packet is changed to enable (setting the flag to 1 and changing the TransmitInProgress flag to 0). The next action is to forward the packet to a certain logical output port. Finally, timestamping is immediately executed by copying the local time of SDN-OLT on the timestamp field of the packet and then forwarding the packet to the RS layer. Figure 3-8 shows the state diagram of the actions related to the GATE message, and Figure 3-10 illustrates the procedures followed for the DISCOVERY GATE, GATE and REPORT messages.

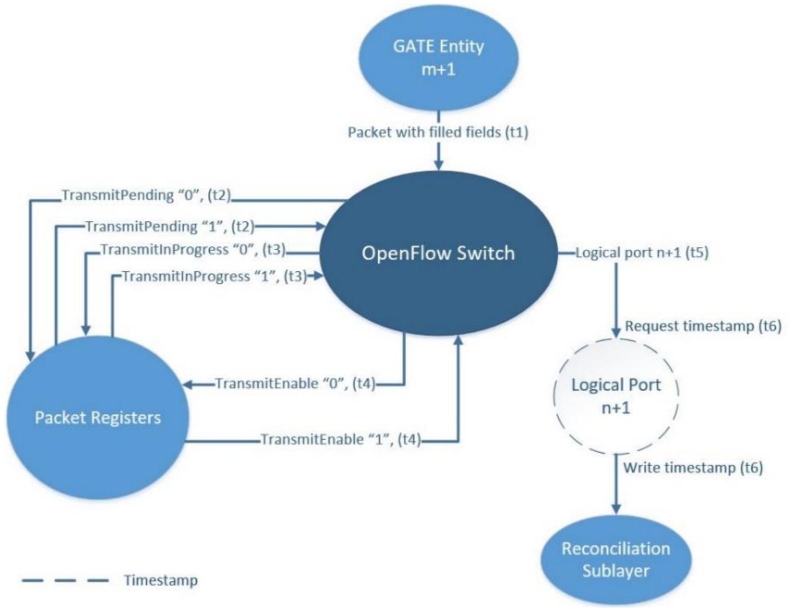


Figure 3-8 GATE related actions in OF-switch.

In the case of the REPORT message (on the right side of Figure 3-10), when the incoming REPORT message from an ONU is matched against a rule in the flow table, the action of calculating the RTT is executed on the packet (via port re-circulation and interacting with the packet registers). Once the RTT is calculated, the packet is forwarded to the REPORT entity for further processing. Figure 3-9 shows the state diagram of the actions related to the REPORT message.

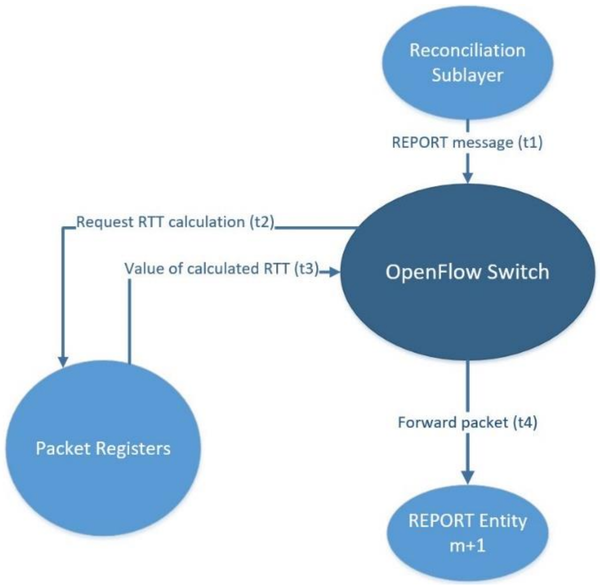


Figure 3-9 REPORT related actions in OF-switch

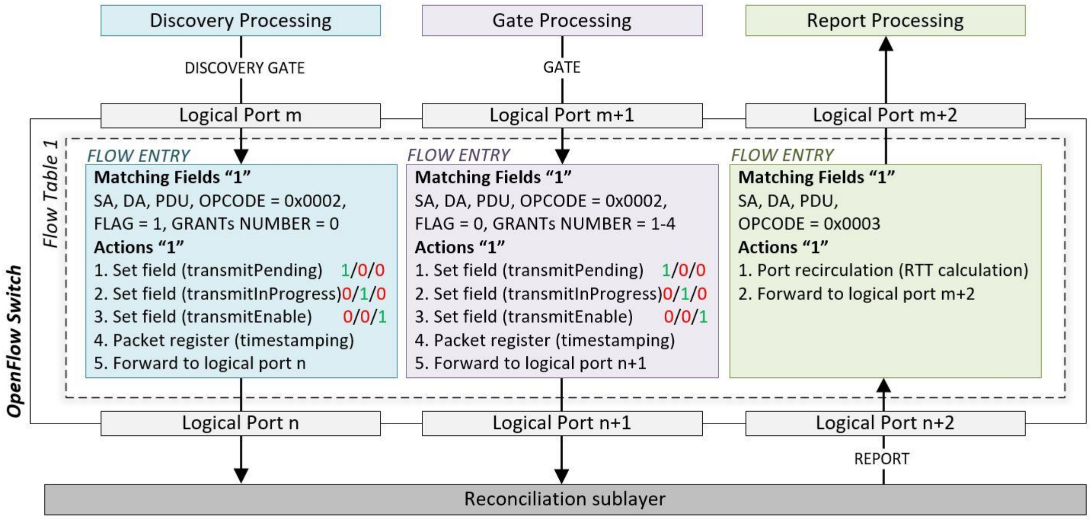


Figure 3-10 Procedures related to the processing of the DISCOVERY GATE, GATE and REPORT messages in the OF-Switch.

3.3.3 Comparison with the CORD architecture

Given the relevance of CORD architecture and its closeness with our work, we now provide more detail and a comparison with our proposal.

The migration from the current central office architecture to that envisioned by CORD includes two phases. The first step is to decouple and virtualize devices into software-based equipment. This is performed by virtualizing the legacy OLT, Customer Premises Equipment (CPE), and Broadband Network Gateways (BNGs). The second step is to provide a control framework for each of the decoupled and virtualized devices to make them a coherent end-to-end system.

Depending on the market, CORD defines three use cases [78]: 1) Enterprise-CORD (E-CORD) is targeted at the enterprise market, which enables service providers to offer SDN-WAN services; 2) Mobile-CORD (M-CORD) is oriented to the mobile market, and enables service providers to move towards 5G; and 3) Residential-CORD (R-CORD) combines virtual-CPE (vCPE) and virtualized access technologies (such as GPON, 10GPON and G.Fast) with cloud-based subscriber services (such as video delivery) to support the broadband access network. In R-CORD, which is the case of interest for our scenario, the CPE is virtualized as a virtual subscriber gateway (vSG) in the central office and runs a bundle of subscriber-selected functions. In the specific case of PON access networks, the OLT is virtualized as vOLT, and its control plane runs on the ONOS controller (including legacy OLT functionalities such as GPON protocol management and VLAN bridging). Finally, the BNG is virtualized as a virtual router (vRouter), also controlled by ONOS, and manages flows as they go through the switching fabric. The vOLT is the CORD element that is most closely related to our design. Its specification [79] defines a line card that supports 2.5 Gbps connectivity when connected to an Ethernet switch. It includes a vOLT agent that runs in a controller or virtual machine and is responsible for establishing the connection between ONOS and the hardware (organized as a datacenter, including several servers and switches) and abstract the entire datacenter-based PON system as a single switch, Figure 3-11.

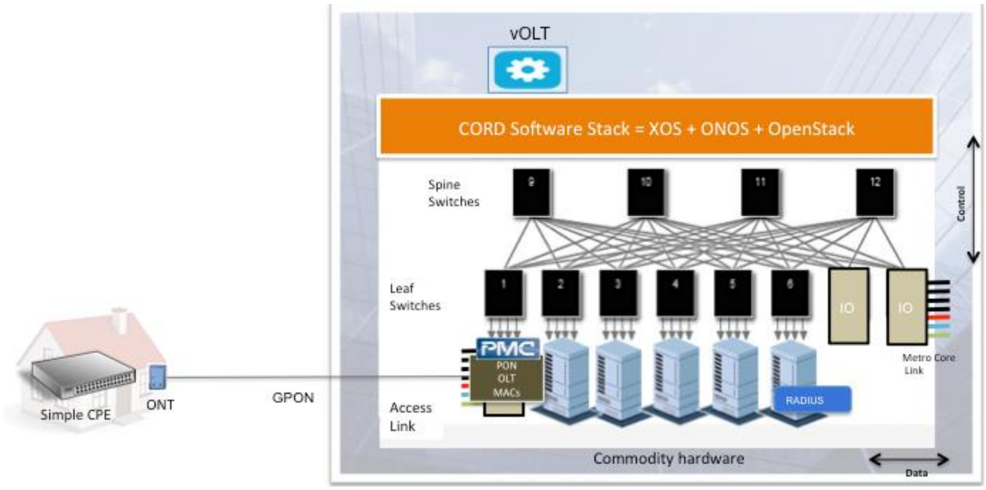


Figure 3-11 CORD hardware architecture [80].

Both our SDN-based SIEPON proposal and the CORD architecture have the same goal for a broadband access network: to simplify the architecture of the OLT by virtualizing and migrating part of its functionalities to the cloud/backbone network, thus building a virtualized OLT around a cheap and standard OpenFlow switch. Our architecture is focused on EPON networks, while CORD is aimed at GPON networks. ONOS is still based on OpenFlow v1.3 as of May 2017, while our approach is based on OpenFlow v1.5, thus enabling us to take advantage of new OF protocol functionalities - such as port recirculation and packet register. Our proposal is a first goal towards a synchronous solution for SDN-based devices, while to the best of our knowledge ONOS does not tackle this issue. In CORD, the whole MAC remains in the OLT, while in our solution, some parts of MAC functions and interfaces (such as the discovery, gate, and report processing entities) are virtualized and migrated to the cloud, and managed as rules in the flow table of OF-switch. Finally, our proposal is compliant with the SIEPON standard, making it easier to integrate with other equipment. To the best of our knowledge, CORD is not SIEPON compliant, although it includes a service framework that seems to (partially) cover similar functionalities.

3.4 Results and Discussion

To demonstrate the validity of our architecture, we built a simulation scenario in OPNET Modeler (see appendix A), and ran some simulations to evaluate the performance of the proposed SDN-EPON architecture. The result is compared with the legacy EPON case in terms of performance, delay, and throughput. The simulated network topology includes an SDN controller (whose operations are partially implemented in the simulator), an SDN-OLT, and 16 ONUs. OpenFlow is used as an interface between the SDN-EPON controller and the OF-switch included in the core of the SDN-OLT. The distance between the SDN controller and the SDN-OLT is set to 1 kilometer, while the ONUs are at distances ranging from 16 to 18 kilometers from the OLT.

Table 3-4 Simulation parameters.

Parameters	**Value**
Link rate	*1 Gb*
Maximum cycle time	*1ms*
Guard time	*1μs*
Message processing delays in SDN-OLT	*0.0164ms*
DBA algorithm	*IPACT*
Traffic source	*Self-similar*
Network topology	*Tree*
SDN-OLT distance (d)	$16 < d < 18$ *km*

Two traffic generators are used: constant bitrate and self-similar traffic [81], [82]. For CBR traffic, packets have a constant length of 791 bytes, while in the second case packet size is uniformly distributed between 64 and 1518 bytes (with a mean of 791 bytes, to compare with the CBR case), and the Hurst parameter (a measure of the long-range dependence of self-similar traffic) was set to 0.7, 0.8, and 0.9. Table 3-4 lists other parameters used in the simulation.

The results are independent of the hardware used for the simulations, since all the modules are simulated, not emulated. Each run of the simulator corresponds to half a second of real time, during which the ONUs generate traffic and the OLT manages it.

The validation comprises two parts. First, we study the low-level SDN-OLT performance, and later we discuss some high-level considerations related to network operations.

3.4.1 Low level SDN-OLT Performance

In this section, we evaluate three specific aspects of the proposed architecture.

- Delay from the SDN-EPON controller to the SDN-OLT.
- Performance of the OF-switch.
- QoS in the path between OF-switch and the ONUs.

Figure 3-12 illustrates the measurement points for traffic delay (green flow), queuing delay and system throughput (orange flow).

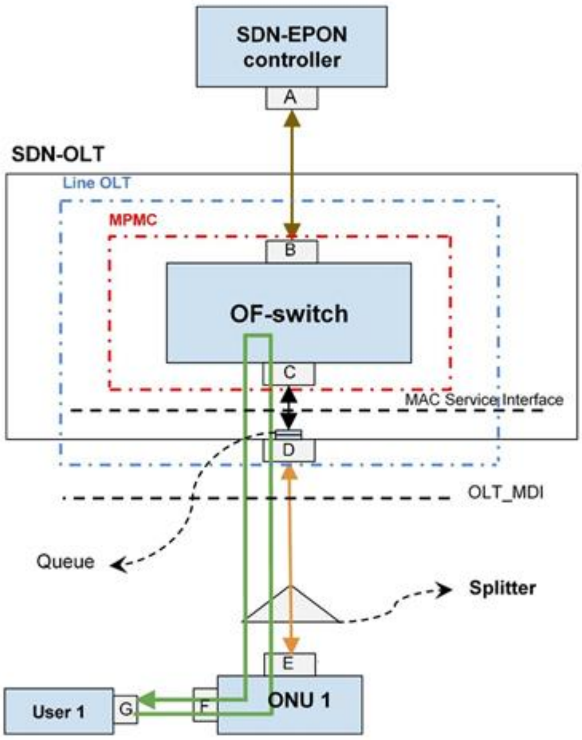


Figure 3-12 Measurement points in SDN-EPON architecture.

Below we define the notation used hereafter:

$T_{controller-process}$	*Average processing time of the packet at the controller*
T_{delay}	*Average transmission delay time of the packet*
$T_{controller-response}$	*Average response time of the controller for the packet*
$T_{P-OF-switch-process}$	*Average processing time in presence of the rules in flow table*
$T_{a-OF-switch-process}$	*Average processing time in absence of a rule in flow table*
$T_{firstTime-lookup}$	*First average time of packet look-up in flow table*
$T_{transmission}$	*Average transmission time of the packet*
$T_{secondTime-lookup\&actions}$	*Second average time of packet look-up in flow table and actions*

3.4.1.1 Delay from the SDN-EPON Controller to the SDN-OLT

The performance of the controller is analyzed by evaluating the processing time of the packets inside the controller and its transfer delay (the packet delay between the controller and the OF-switch). The average processing time needed by the controller for collecting and analyzing the content of the Packet-In message, create a rule (flow entry) and issue a FlowMod message and the Packet-Out message is $T_{controller-process} = 1.986\mu s$ with a 99% confidence interval of $\pm 0.0001\mu s$. The average delay[7] of a packet from the physical port A of the controller to the logical port B of the OF-switch (Port B is an asynchronous port that is embedded in the MPMC functional block of the SDN-OLT) is measured as $T_{delay} = 5.243\mu s$ with a 99% confidence interval of $\pm 0.001\mu s$. Hence, the average response time of the controller is:

$$T_{controller-response} = T_{controller-process} + 2 * T_{delay} = 12.472\mu s$$

Table 3-5 details the average packet delay values of the six initial FlowMods (GATE, REPORT, DISCOVERY GATE, REGISTER_REQ, REGISTER, and REGISTER_ACK) from the controller to the OF-switch at the initial time. The average packet delay is measured for two different cases of distance between the controller and the SDN-OLT, 1 Km and 10 Km. Even for the longer distance, in which a single controller

[7] The average delay is composed of propagation time and transmission time.

located in a data center could manage simultaneously several OLTs in a diameter of 20 Km (i.e., a medium-sized city), response times are almost negligible.

Table 3-5 Average values of packet processing delays.

FlowMod	Tx delay in the controller (port A)	Rx delay in the switch (port B), d =1 km	Rx delay in the switch (port B), d =10 km	Tx delay through path AB d=1 km	Tx delay through path AB d=10 km
DISCOVERY GATE	*0.52* µs	*5.763* µs	*51.66* µs	*5.243* µs	*51.14* µs
REGISTER_REQ	*1.04* µs	*6.283* µs	*52.18* µs		
REGISTER	*1.56* µs	*6.803* µs	*52.70* µs		
REGISTER_ACK	*2.08* µs	*7.323* µs	*53.22* µs		
GATE	*2.6* µs	*7.843* µs	*53.74* µs		
REPORT	*3.12* µs	*8.363* µs	*54.26* µs		

Based on the simulation results, the impact of distance on transmission delay of a packet between controller and SDN-OLT is shown that with increase the distance, the transmission delay will be increased. Hence, from distance of 1 km to 10 km, the transmission delay have been increased from 5.243 μs to 51.14 μs, respectively.

3.4.1.2 Performance of the OpenFlow Switch

We now evaluate the performance of the SDN-OLT and compare it with the legacy OLT. We study two cases: 1) a proactive system (presence of the rules flow entries in the OpenFlow table); and 2) a reactive system (absence of such rules). In the proactive system, the controller has uploaded the set of rules such as GATE, REPORT, DISCOVERY GATE, REGISTER_REQ, REGISTER, and REGISTER_ACK at the beginning of the connection, for all the ONUs. The delay measurement is performed from the instant when a packet (control frame/data frame) enters the OF-switch through a synchronous port, and includes the look-up in the flow table[8] for a rule to match and perform a set of actions (read and write a register, and forward the packet to the output port). The average processing time in presence of the rules in the flow table is $T_{p-OF-switch-process} = 0.488\mu s$ with a 99% confidence interval of $\pm 0.003\mu s$.

[8] We defined one flow table and the length of flow entries are dynamic.

In the reactive system, in the absence of a previously defined rule in the flow table, we measured the delay from the moment when the switch does not find a rule to match the packet in the look-up process. Then, the OF-switch sends a Packet-In message to the controller through the OpenFlow interface and, based on that, the controller generates a FlowMod message and a Packet-Out message to set a rule in the flow table. The remaining actions for this case are similar to the first case. The average processing time in absence of a rule is $T_{a-OF-switch-process}$. This time is composed of

A. The first time packet look-up $T_{firstTime-lookup} = 0.297\mu s$;
B. Packet transmission time to logical port B $T_{transmission} = 0.096\mu s$;
C. Round trip message delay from OF-switch to the controller and vice versa $2 * T_{delay} = 10.486\mu s$;
D. Controller processing time $T_{controller-process} = 1.986\mu s$;
E. Second time look-up and actions $T_{secondTime-lookup\&actions} = 0.488\mu s$.

$$T_{a-OF-switch-process} = T_{firstTime-lookup} + T_{transmission} + T_{round-trip-delay} + T_{controller-process} + T_{secondTime-lookup\&actions} = 13.353\mu s$$

Look-up time process will be vary based on the length of the flow table and the number of flow entries. Thus, for both cases (reactive and proactive systems) the minimum average of look-up time for the number of 10 flow entries is $0.294\mu s$, and the maximum average of look-up time for the number of more than 100 flow entries is increased to $0.3\mu s$.

3.4.1.3 QoS in the path between the OF-switch and the ONUs

Figure 3-13 shows the RTT delay versus the offered load for the cases of SDN-EPON and legacy EPON. The delay is measured as the round trip time of the data frames that are generated by the end user (physical port G), sent to the OF-switch (logical port C), and returned. This not the actual path that packets follow (they are either sent from the end user towards the core network, or vice versa) but it provides an idea of how the system performs overall. For the sake of brevity, we use the term "RTT" as a short name for the sum of the delays. Figure 3-13 includes box plots, where for each offered load we represent the average value of delay (center of the box), the first and third quartiles (bottom and top of the box), the minimum and maximum values (whiskers above and below the box) and the low minimum and high maximum values (red points scattered).

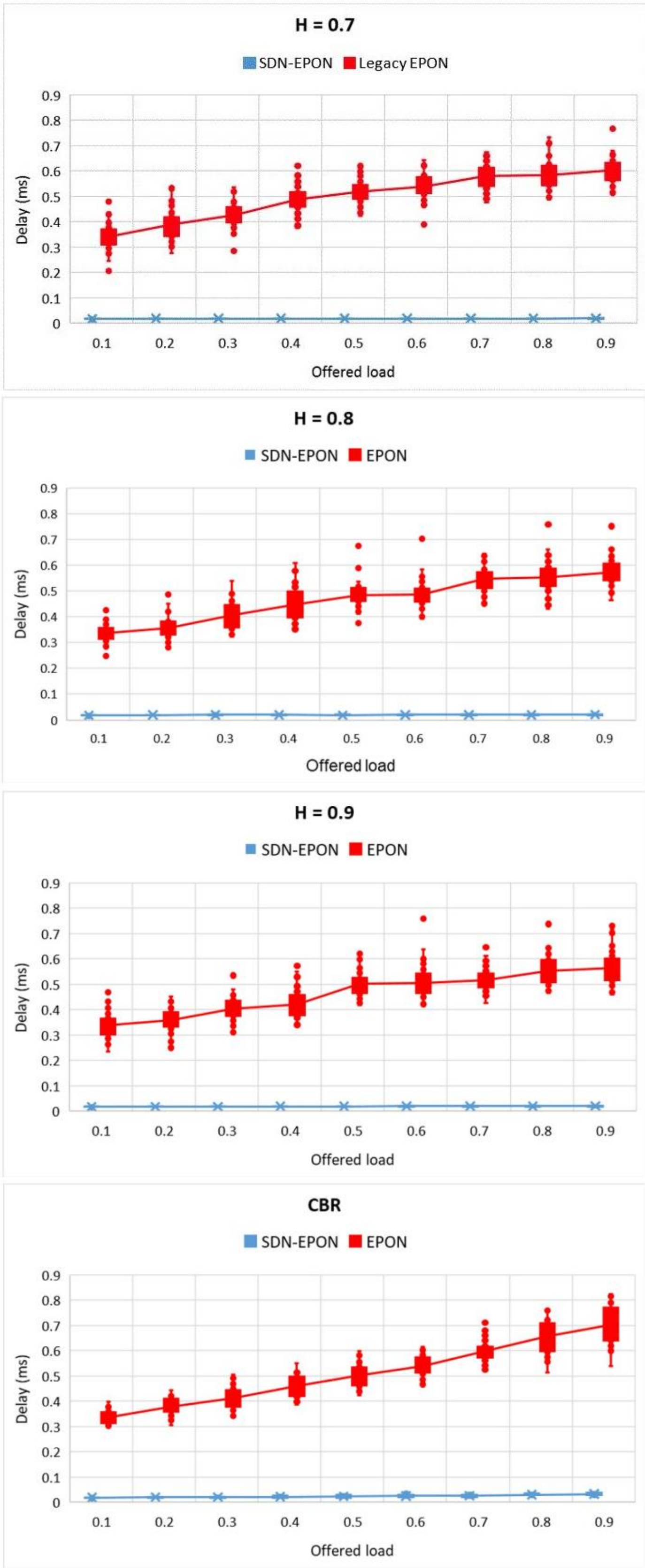
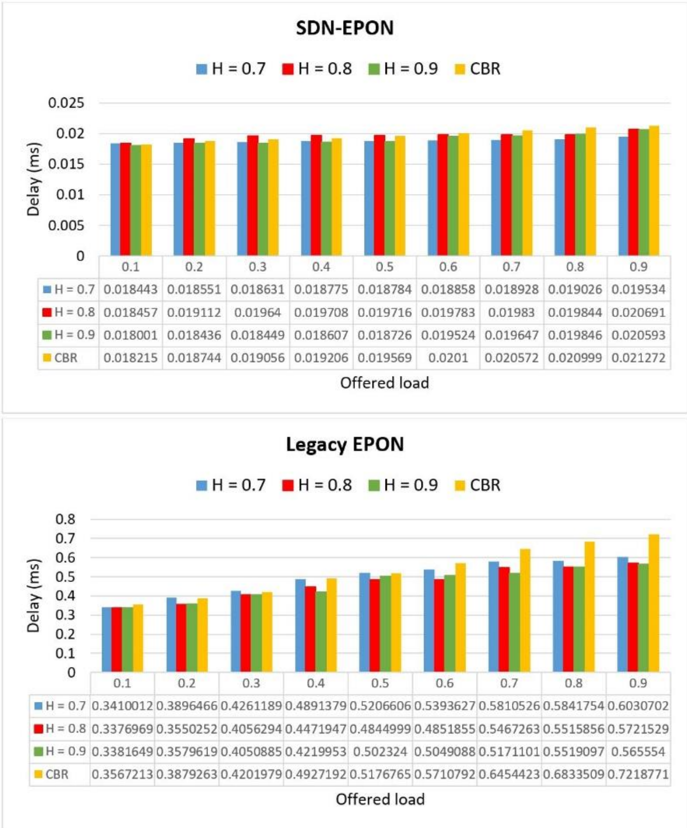


Figure 3-13 Comparison of RTT in SDN-EPON and legacy EPON scenarios under different offered load, for CBR and self-similar traffic patterns and different Hurst parameter.

As shown (as a green line) in Figure 3-12, our evaluation for one ONU includes the look-up and rule-matching time, port recirculation, time to read and write registers, and waiting time in the port queue (just before point D in the MAC service interface). To interpret correctly the Figure 3-13, Figure 3-15, Figure 3-19, and Figure 3-20, it is important to note that blue and red data points do not coincide in the same position in the x-axis in order to avoid the overlapping the points and thus facilitate visualization. Both scenarios are evaluated at the same offered load and each blue or red points in the figures corresponds to the average value obtained for all the ONUs. The offered load is expressed in steps of 10% fractions of the maximum capacity of the system.

The results in Figure 3-13 show that the RTT in the SDN-EPON scenario is notably improved compared to the legacy EPON scenario, especially under high load. In the legacy EPON scenario the RTT increases dramatically with the network load (in H = 0.7, RTT increase at load 0.1 equal to 0.341ms to load 0.9 equal to 0.603ms, in H = 0.8, it increases from 0.337ms to 0.572ms, and finally in H = 0.9, increase from 0.338ms to 0.565ms), while in the SDN scenario the RTT is almost constant (at load 0.1 is around 0.18ms to load 0.9 equal to 0.20ms). The traffic pattern does not have a significant affect in the results. CBR and self-similar traffic offer the same average results, with less variance in the case of constant traffic. Long-range dependence characteristics of the traffic pattern do not seem to affect significantly the performance, with only a slight increase in the dispersion of the measurements when H is increased (probably related to the presence of more time-correlated traffic bursts when the Hurst parameter is increased).

In Figure 3-14, we show the detailed values of the RTT delay for different scenarios, constant and self-similar traffic patterns and different Hurst parameters. In the SDN-EPON, the evaluated performance for H = 0.7 shows a small delay (mainly in high loads) at different loads. For H = 0.8, we can observe a slight penalty especially at the low loads, and as it can be seen, for H = 0.9 and CBR source, the RTT delay is decreased in the low loads and increased in the high loads, while in the legacy EPON, evaluations show the RTT for different H parameters is almost similar, but in CBR source, we observe more delay at load higher than 0.7. However, in both scenarios the variations are negligible. Simulation results illustrate how the H parameter impacts over the performance of the network, which somehow is what affects the level of the QoS that users can experience in the network.



SDN-EPON

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
H = 0.7	0.018443	0.018551	0.018631	0.018775	0.018784	0.018858	0.018928	0.019026	0.019534
H = 0.8	0.018457	0.019112	0.01964	0.019708	0.019716	0.019783	0.01983	0.019844	0.020691
H = 0.9	0.018001	0.018436	0.018449	0.018607	0.018726	0.019524	0.019647	0.019846	0.020593
CBR	0.018215	0.018744	0.019056	0.019206	0.019569	0.0201	0.020572	0.020999	0.021272

Legacy EPON

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
H = 0.7	0.3410012	0.3896466	0.4261189	0.4891379	0.5206606	0.5393627	0.5810526	0.5841754	0.6030702
H = 0.8	0.3376969	0.3550252	0.4056294	0.4471947	0.4844999	0.4851855	0.5467263	0.5515856	0.5721529
H = 0.9	0.3381649	0.3579619	0.4050885	0.4219953	0.502324	0.5049088	0.5171101	0.5519097	0.565554
CBR	0.3567213	0.3879263	0.4201979	0.4927192	0.5176765	0.5710792	0.6454423	0.6833509	0.7218771

Figure 3-14 RTT in SDN-EPON and legacy EPON scenarios under different offered load, traffic patterns, and Hurst parameter

The remarkably small confidence interval in the SDN scenario of Figure 3-13 (99% confidence interval of ±0.0005ms) is due to the existence of pre-defined rules in the flow table of the OF-switch, which makes it processing time almost constant and much faster than the procedures followed in the legacy EPON scenario. In the legacy EPON, the DBA must control and manage multiple Control Parser and Control Multiplexer entities (one per instance) to receive and send packets, respectively; while the DBA only controls the OF-switch in the SDN-EPON architecture. Therefore, the DBA procedures for the timing management and processing time of a packet inside the SDN-OLT are greatly reduced. This also explains the higher variability (red dots dispersion) in the legacy case, compared with the more deterministic behavior of the new architecture.

Figure 3-15 shows the downstream queuing delay versus the offered load at the OLT in both architectures and for different traffic sources (self-similar with different Hurst parameters and CBR). The queuing delay is measured for both control and data frames at the SDN-OLT/OLT physical port (port D in MAC service interface), and correspond to measurements of the packet waiting times in the downstream queue.

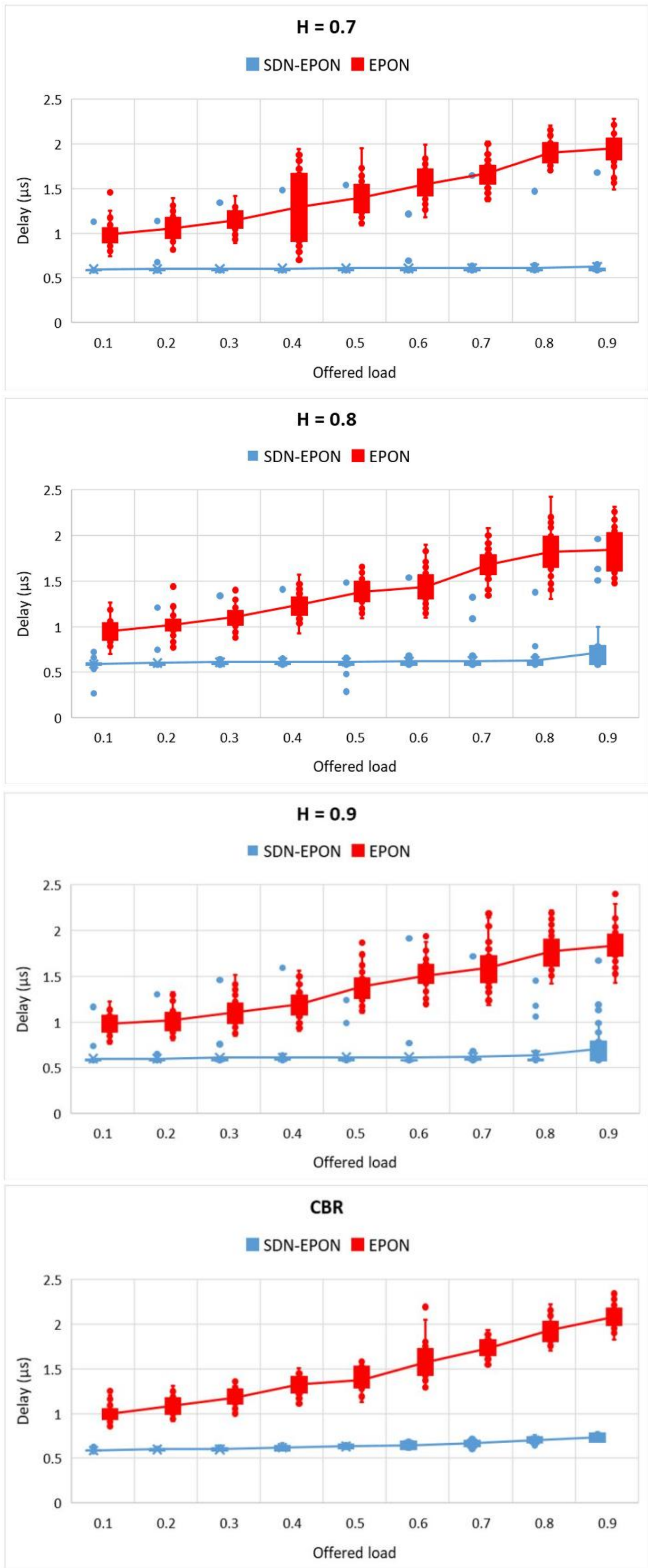


Figure 3-15 Comparison of queuing delays in SDN-EPON and legacy EPON scenarios under different offered load, for CBR and self-similar traffic patterns and different Hurst parameter.

The measurements are taken from the time a packet enters the transmitter channel queue to the time the last bit of the packet is transmitted. We evaluate the queuing delay to see how the new architecture affects the downstream transmission in the SDN-OLT side. The OF-switch affects the downstream transmission by controlling the synchronous logical ports in dialogue with the DBA.

Figure 3-15 shows that the queuing delay in both scenarios is quite different. Legacy EPON scenarios experience different behavior depending on the offered load. SDN-EPON shows a small queuing delay, that is almost independent of the offered load for all H parameters (for H = 0.8 and 0.9, at load 0.9, we can observe slight grows and maximum variation). This is because the OF-switch schedules the packets transmission via the queue and scheduler blocks of ESPs to the logical ports, and then executes the transmission of packets to the MAC service interface by interacting with the DBA through the P, I and E registers. However, the queuing delay in the legacy EPON scenario grows with the offered load. In this case, for each measurement value, we can see more variation in the values for different H parameters (with the maximum of variation at load 0.4 for H = 0.7, possibly due to statistical randomness of the packet generator at this value).

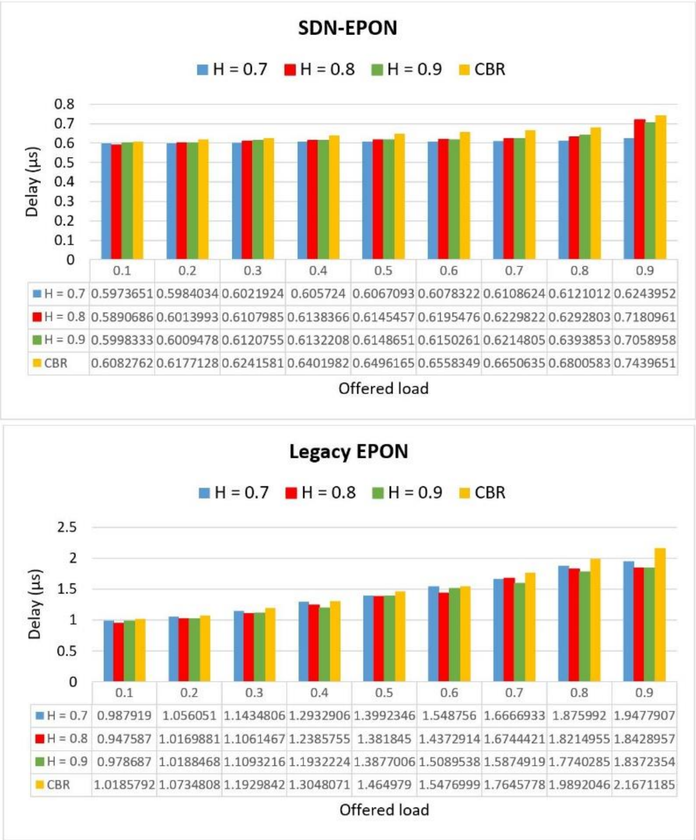


Figure 3-16 Queuing delay in SDN-EPON and legacy EPON scenarios under different offered load, traffic patterns, and Hurst parameter.

As it can be seen, Figure 3-16 detailed the simulation results in both scenarios for CBR and self-similar traffic patterns and different H parameters.

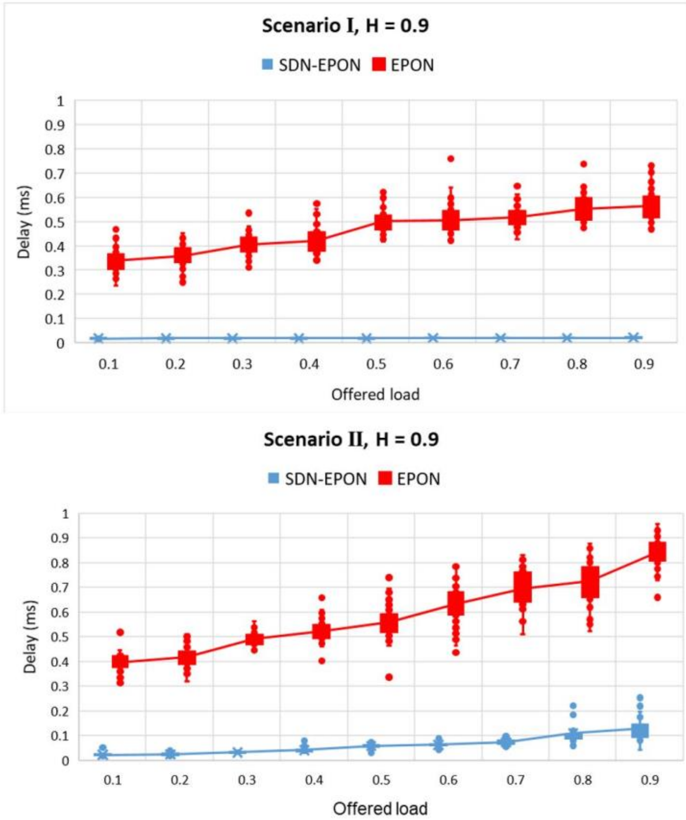


Figure 3-17 RTT delay in SDN-EPON and EPON architectures under different number of end users (sixteen active ONUs on the top, eight active ONUs on the bottom), for different offered traffic load.

The same phenomenon is seen in the queueing delay analysis, whose results are shown in Figure 3-18. As the number of packets per second increases at each ONU, the queueing delay increases too, but much faster for the legacy EPON scenario, and both scenarios suffer a slightly higher delay when the load is concentrated in a few the ONUs.

Figure 3-19 and Figure 3-20 show the aggregated throughput of all ONUs in both the downstream and upstream channel versus the offered load. The throughput is measured (including the bytes of control and data frames between the SDN-OLT/OLT port D) in Figure 3-12 and the ONU port E.

For all values of H, the SDN-EPON scenario shows an average improvement of 4% in the downstream channel at different traffic loads compared to the legacy EPON scenario, while in the upstream channel, SDN-EPON improvement is close to 3%.

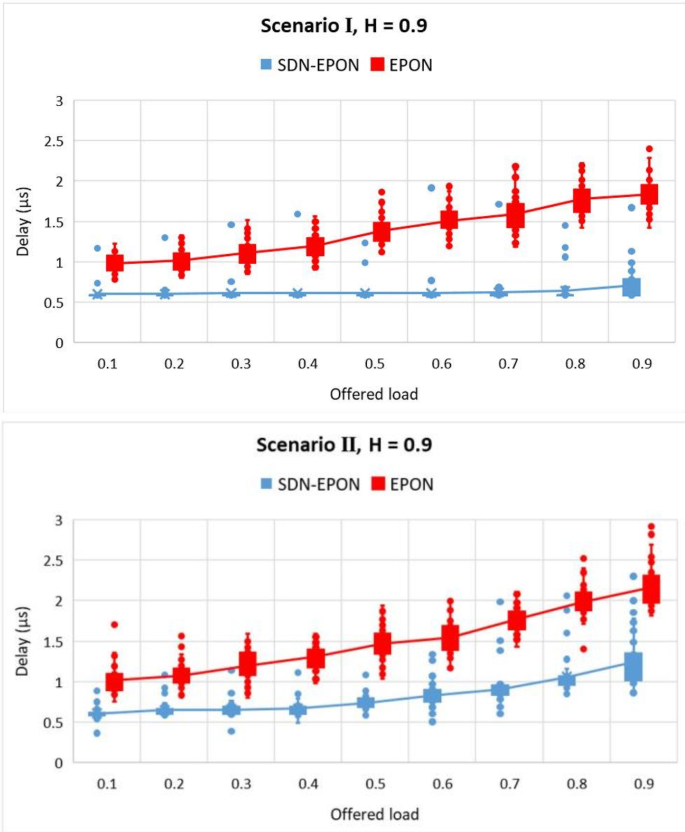


Figure 3-18 Queuing delay in SDN-EPON and EPON architectures under different number of end users (sixteen active ONUs on the top, eight active ONUs on the bottom), for different offered traffic load.

This improvement is caused by the reduced queuing delay observed in the SDN-EPON case, thanks to the faster packet processing once the OpenFlow rules have been set in the OF-switch.

Figure 3-21 and Figure 3-22 illustrate average throughput in both upstream and downstream channel for constant and self-similar traffic patterns and different H parameters. Results show the throughput for both scenarios is quite similar, no matter what sources and H parameters are used.

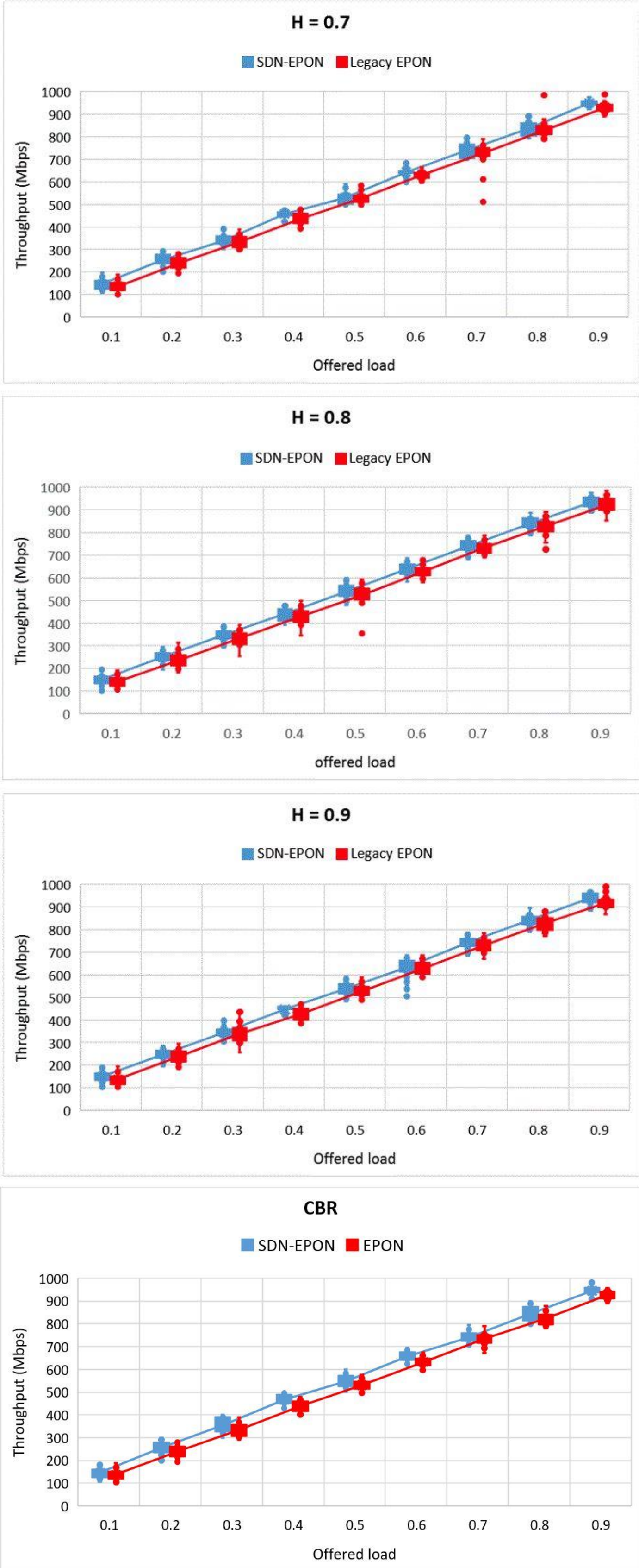


Figure 3-19 Comparison of upstream throughput in SDN-EPON and legacy EPON scenarios under different offered load and Hurst parameters

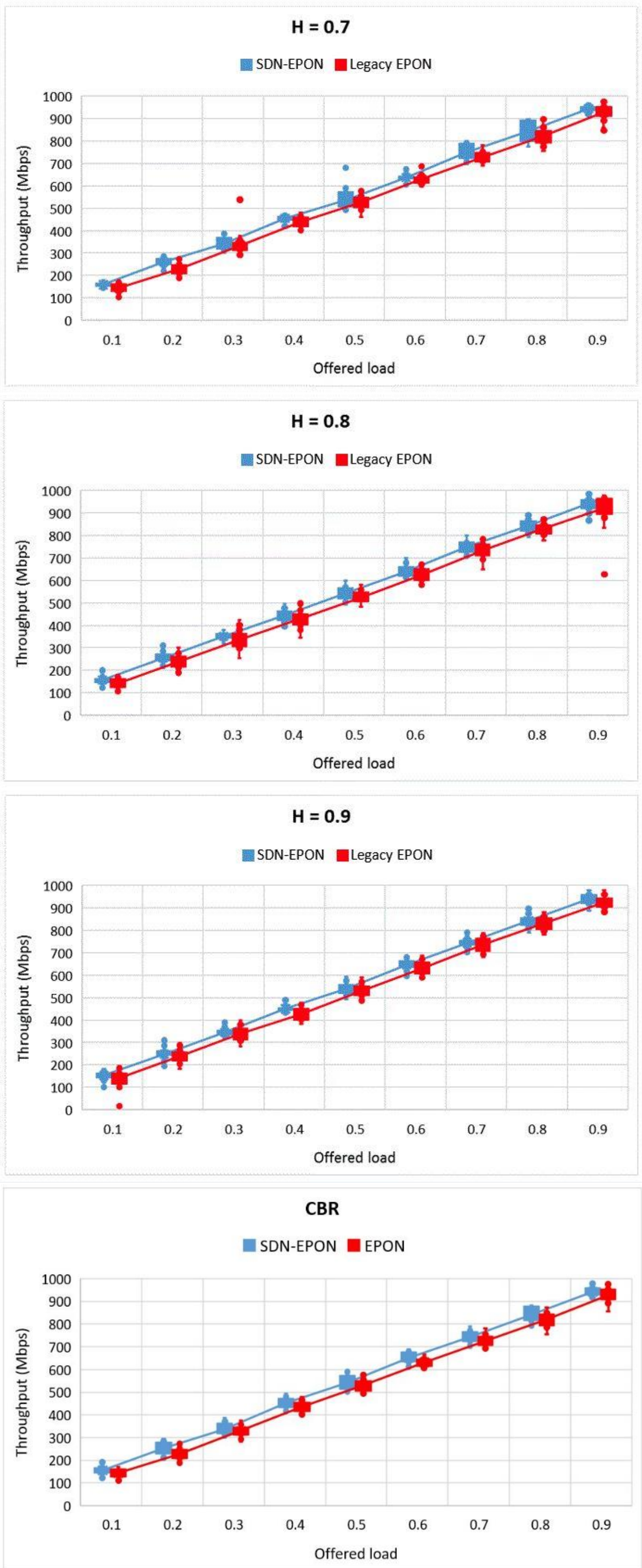


Figure 3-20 Comparison of downstream throughput in SDN-EPON and legacy EPON scenarios under different offered load and Hurst parameters

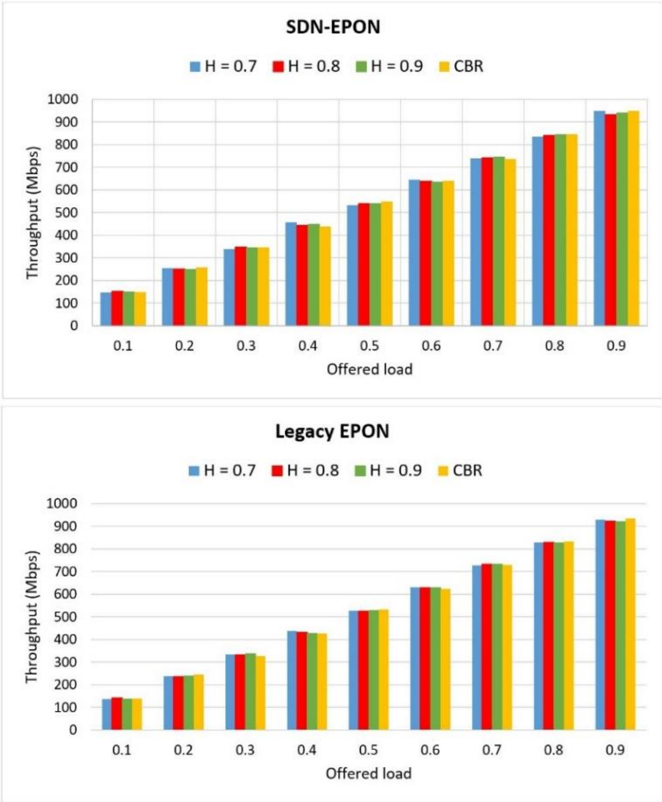


Figure 3-21 Upstream throughput in SDN-EPON and legacy EPON scenarios under different offered load, CBR and self-similar traffic patterns, and Hurst parameter.

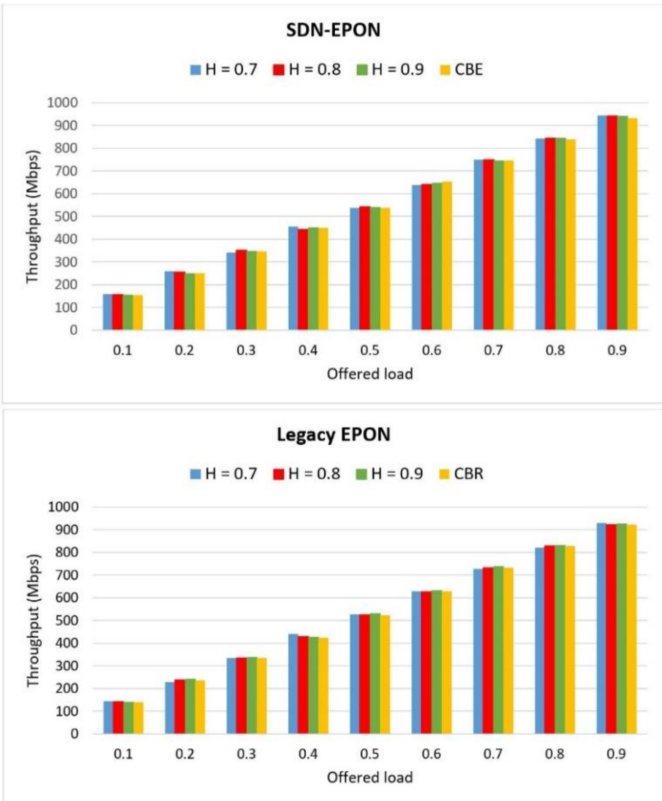


Figure 3-22 Downstream throughput in SDN-EPON and legacy EPON scenarios under different offered load, CBR and self-similar traffic patterns, and Hurst parameter.

The average delay in the SDN scenario under different packet size has been plotted in Figure 3-23, with self-similar traffic with H = 0.7. The results show that increasing the packet size distribution causes an increasing the average delay and thus a reduction of QoS in the network. The reason is that when the packet size increases, the transmission time increases as well and fewer packets can be queued in the buffer, and therefore larger packets take longer to be received, queued and transmitted than smaller packets.

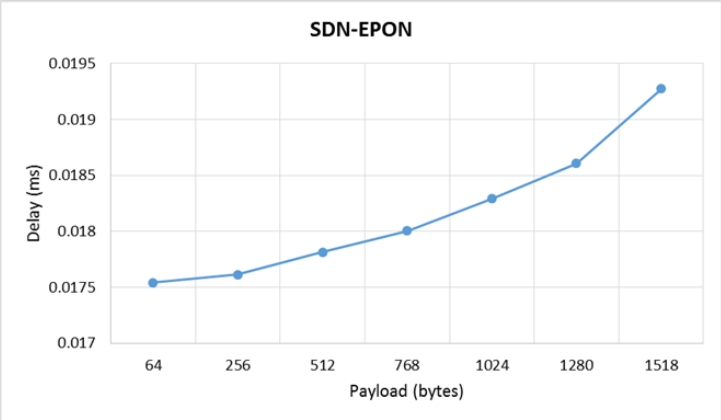


Figure 3-23 Average delay in SDN-EPON scenario under different packet size with H=7.

In term the SDN-EPON scenario, Figure 3-24 shows that the average traffic delay is increased when the number of packets are increased in the network. This result illustrates that as the number of packets are increased, the average transmission delay and the QoS will be degraded. The increase of the number of packets that send in the given period of time leads to an increase in throughput of the network (the larger packet rate, the higher throughput). But as it can be seen it does not significantly affect the SDN-EPON network performance.

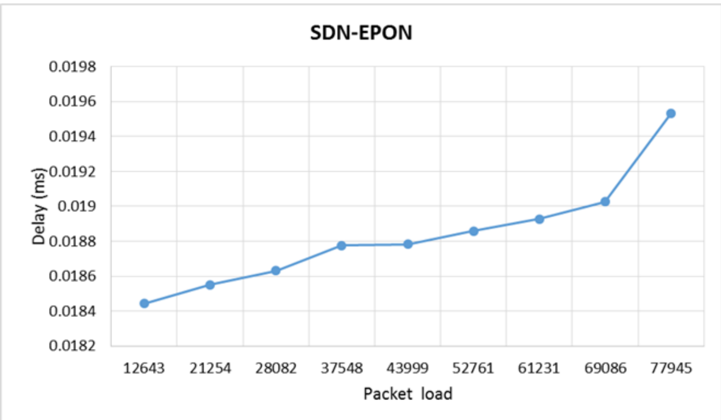


Figure 3-24 Comparison of average traffic delay in SDN-EPON scenario under different packet load with H=7.

Overall, the results show that the performance evaluation (throughput, queuing delay and traffic delay) of the SDN-EPON architecture is better than the legacy EPON architecture, and is remarkable in the case of the delay of data frames.

3.4.2 High-level Consideration related to operations

The previous section has described the advantages of our proposed architecture in a low-level validation. At a higher level, there is also an advantage in our proposal in terms of OPEX. Unlike legacy EPON architectures, where service providers are limited by the proprietary management interfaces and cannot easily modify the QoS parameters, the SDN-EPON architecture can easily manipulate QoS policy of the network via a flow message. It is as simple as setting a flow with QoS following the SIEPON standard and uploading the specific rule to the OF-switch. The controller can generate a new FlowMod with actions corresponding to QoS and upload it in a scale of 5 to 8μ*s* in the OF-switch, thus enabling very fast reconfigurations of the QoS profile assigned to each user. This results in a very flexible framework for network operators.

3.5 Summary

This chapter has presented a novel architecture to minimize complexity of operations and management of EPON networks while optimizing dynamically the use of resources. We proposed a new scheme based on the SDN paradigm applied to EPON, by defining an extension to the Multipoint MAC Control sublayer and splitting some of its functionalities between the SDN-EPON controller and the SDN-OLT, which is built around an OpenFlow switch. In this scheme, we migrate the control and management modules of the MAC control client and MAC client to the SDN-EPON controller - while keeping their main functional blocks in the SDN-OLT and integrating them with the OF-switch. The operations that must be executed at short timescales remain at the OLT, while those that can be executed at longer time scales can be migrated to the controller.

By extending the messages and rules of OpenFlow version 1.5, we could implement the synchronous operations required by EPON. Our design opens the way to the design of multi-tenant and multi-provider PON networks, thanks to the flexibility provided by virtualization. The design also enables the optimization of resource allocation and service availability thus enhancing SLA and QoS requirements. We evaluated our approach with simulations showing that the SDN-EPON approach is more flexible than the legacy

EPON in terms of operations, and improves the average throughput of the ONUs in downstream and upstream channels and the average data packet delay.

4 AN ENERGY-EFFICIENT DYNAMIC BANDWIDTH ALLOCATION ALGORITHM FOR EPON

EPON is more energy-efficient (in terms of energy spent per bandwidth and distance) than other access solutions such as xDSL, due to higher bandwidth and lower transmission losses [83]. With the rapid deployment of EPONs, a large number of end user equipments are added to the access segment, thus increasing the global energy consumption of the access network. To minimize energy consumption, several researchers have proposed various energy-saving solutions that target the EPON scenario, as described in Section 2.3.3.

In this chapter, we focus on the minimization of energy consumption in EPON networks. We present a distributed energy-efficient Dynamic Bandwidth Allocation (DBA) algorithm to power-off the transmitter and/or receiver of an ONU whenever there is no upstream or downstream traffic. We have been able to combine the advantage of a distributed DBA such as DDSPON (a smaller packet delay, due to the shorter time needed by DDSPON to allocate the transmission slots) and the energy-saving features (that come at a price of longer packet delays due to the fact that switching off the transmitters make the packet queues grow). Our proposed DBA algorithm minimizes the ONU energy consumption across a wide range of network loads. Our simulation show reductions of energy consumption up to 78% at high network load at the ONU side, while maintaining at an acceptable level the penalty introduced in terms of channel utilization and packet delay. To the best of our knowledge, this is the first time an energy saving algorithm is applied to a distributed DBA.

This chapter is organized as follows. Section 4.1 describes the algorithm. In Section 4.2, we present the simulation environment scenarios and describe the results obtained from the performance evaluation, and compare our results with the proposals from other authors. Section 4.3 concludes the chapter.

4.1 Minimizing Energy Consumption in ONUs

This section presents the concepts related to power states and power saving period, an energy-saving algorithm, extended control frames, and functionalities and operational schemes of OLT and ONUs. To implement our mechanism, we use the DDSPON DBA algorithm [84], which can distribute bandwidth allocation and decentralized the calculation of timeslot for each ONU (described in Section 2.3.1.2). In Table 4-1, we define the notation used hereafter.

Table 4-1 Energy-saving algorithm notation

Symbol	Description
n	*Current cycle of the DBA*
$\vec{\Phi}$	*Weight vector*
$\Phi_i(n)$	*Weight value of the* ONU_i *computed in cycle n*
Φ_i^{conf}	*Initially configured weight value of the* ONU_i
$\Phi_j(n)$	*Remaining weight value for the rest of ONUs in cycle n*
$Q_i(n)$	*Queue size in cycle n, in bits*
$Q_{UP_i}(n)$	*Size of the upstream queue of the* ONU_i *in cycle n, in bits*
$Q_{DS_i}(n)$	*Size of the downstream queue of the OLT for* ONU_i *in cycle n, in bits*
$R_i(n)$	*Requested transmission length by* ONU_i *in cycle n, in bits*
$Q_{m-UP_i}(n)$	*Moving average of the queue of the* ONU_i *in cycle n, in bits*
$Q_{m-DS_i}(n)$	*Moving average of the queue of the OLT for* ONU_i *in cycle n, in bits*
$R_{m_i}(n)$	*Moving average of the transmission length requested by* ONU_i *in cycle n, in bits*
$DT_i(n)$	*Downstream transmission length sent by the OLT for* ONU_i *in cycle n, in bits*
$DT_{m_i}(n)$	*Moving average of the downstream transmission length sent by the OLT to* ONU_i *in cycle n, in bits*
$W_i(n)$	*Transmission window size of* ONU_i *in cycle n, in bits*
W_{max}	*Maximum transmission window size of the DBA, in bits*
T_{max-sc}	*Maximum sleep cycle of the ONUs, in ms*
T_{max}	*Maximum period for a DBA cycle, in ms*
α	*Weight of the moving average*
D_{UP_i}	*Ratio between queue size and requested transmission length of* ONU_i
D_{DS_i}	*Ratio between queue size and downstream transmission length of the OLT for* ONU_i
Rx	*Receiver element at the ONU*
Tx	*Transmitter element at the ONU*
TRx	*Transmitter and receiver elements at the ONU*
T_{st}	*Transmission start-time*
$T-low-power_{UP}$	*Duration of the power saving state in doze mode, in ms*
$T-low-power_{DS}$	*Duration of the power saving state in sleep mode, in ms*
$T-sleep$	*Sleep duration for an ONU, in ms*

Our energy-saving mechanism minimizes power consumption by switching the ONUs from the active mode to the sleep or doze (listen) mode. This transition is decided by the OLT, which maintains the grant information (bandwidth request and allocated transmission time) and calculates the transmission window size of each ONU. In addition, every ONU calculates its own bandwidth request by monitoring its queue size and the amount of traffic received from the end user. This process is completed via the exchange of control frames (i.e. REPORT and GATE messages) between the OLT and the ONUs. Based on the information conveyed on the control frames and the status of the queues' size, the OLT can decide whether to keep the ONU in active mode or switch it to the sleep or doze modes. An example of sleep and doze operation of the ONU is shown in Figure 4-1, where ONU 1 and ONU 2 are put to doze and sleep mode for the duration specified by the OLT, respectively (see section 4.1.1 for more detail). In step A of Figure 4-1, ONU 1 starts to transmit data and REPORT message at its transmission time, and then at step B ONU 1 goes to doze mode for the duration specified by OLT, because there is no data to transmit in upstream. In the next cycle, at step C, ONU 1 wakes up to transmit the data and REPORT message. In step D, ONU 2 goes to sleep mode when there are no upstream and downstream transmissions. In step E, ONU 2 sleeps for the duration specified by the OLT for n cycles (sleep cycles depend on the value of T_{max-sc} and amount of traffic in both direction). In step F, when the sleep cycle has elapsed, ONU 2 wakes up at the end of sleep cycle in order to receive and transmit the data, GATE and REPORT messages.

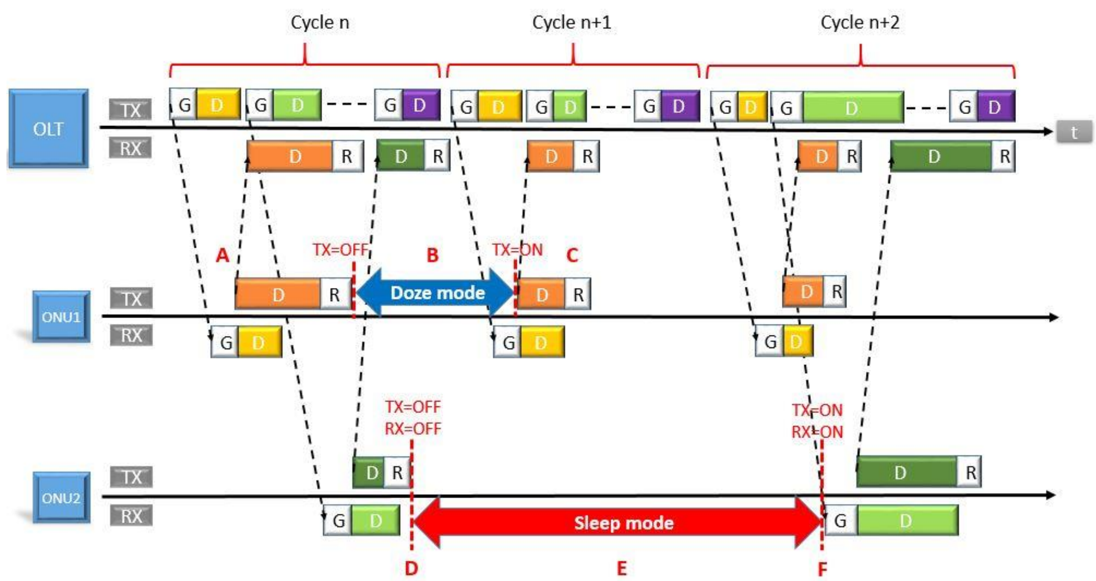


Figure 4-1 Sleep and doze state of the ONU, where G stands for GATE; R for REPORT; and D for Data.

4.1.1 Proposed DBA algorithm

We now address an important aspect of our proposed algorithm, which determines the sleep and doze duration at the ONUs. For this purpose, it is necessary to divide the procedure into two phases that describe which activities are executed in the OLT and which are performed in the ONUs.

As discussed earlier, the energy-saving algorithm uses the information of the upstream queue of an ONU and the OLT downstream queue for the same ONU. The obtained values are shared with the OLT in order to calculate the sleep cycle through the predicted value obtained with a moving average technique. Depending on the instantaneous queue sizes at ONU and OLT, the doze and sleep periods will vary:

- Doze mode: if the upstream queue size in the ONU is smaller than the requested upstream transmission length, the queue will empty, and thus the ONU enters in power-saving mode after transmitting the data and REPORT message. The transmitter of the ONU is turned off for the duration specified by the OLT and the receiver remains on.
- Sleep mode: if the queue size in ONU and OLT are both smaller than the requested upstream transmission length and downstream transmission length, respectively, then both queues will empty, and thus the ONU enters sleep mode (power-saving mode). Both the receiver and the transmitter of the ONU are turned off for the duration specified by the OLT.

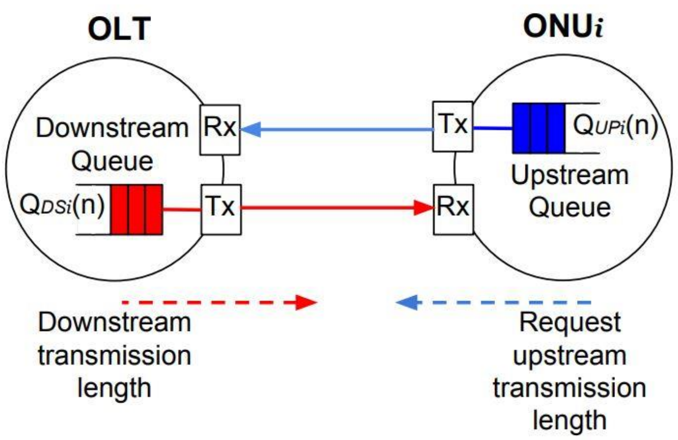


Figure 4-2 Upstream and downstream queues at ONU and OLT.

In case both the upstream and downstream queue size are bigger than the requested upstream transmission length and downstream transmission length at the ONU and OLT, respectively, the ONU will be completely active and cannot enter power-saving mode. Upstream and downstream queue size of the ONU and the OLT are shown in Figure 4-2.

The first phase of the algorithm defines the actions to be executed in the OLT. Figure 4-3 shows the flow chart of the DBA algorithm performed at the OLT. Upon reception of the REPORT message from ONU_i, the OLT extracts and analyzes the information of the REPORT message. The information includes: ONU_i weight $\Phi_i(\mathrm{n})$, request $R_i(n)$ and the current state of the queue $Q_{UP_i}(n)$, Figure 4-2. Once the information is extracted, the OLT computes 1) the moving average of the ONU's queue size $Q_{m-UP_i}(n)$ to predict the state of the queue; 2) the moving average of the request $R_{m_i}(n)$; 3) the moving average of the OLT's queue size $Q_{m-DS_i}(n)$ for ONU_i to predict the state of the queue; and 4) the moving average of the downstream transmission length $DT_{m_i}(n)$ for ONU_i. The moving average of queue in the ONU and OLT at instance *n* is calculated as follows:

$$Q_{m-UP_i}(n) = \alpha * Q_{m-UP_i}(n-1) + (1-\alpha) * Q_{UP_i}(n)$$

$$Q_{m-DS_i}(n) = \alpha * Q_{m-DS_i}(n-1) + (1-\alpha) * Q_{DS_i}(n)$$

where α is the weight of the moving average. $Q_{m-UP_i}(n-1)$ is the last value of the moving average of the queue and $Q_{UP_i}(n)$ is the current state value of the queue (instantaneous queue size) of ONU_i. $Q_{m-DS_i}(n-1)$ is the last value of the moving average of the queue and $Q_{DS_i}(n)$ the current state value of the queue of the OLT. Finally, the moving average of the request and downstream transmission length at instant *n* is calculated as follow:

$$R_{m_i}(n) = \alpha * R_{m_i}(n-1) + (1-\alpha) * R_i(n)$$

$$DT_{m_i}(n) = \alpha * DT_{m_i}(n-1) + (1-\alpha) * DT_i(n)$$

where $R_{m_i}(n-1)$ and $R_i(n)$ are the moving average of the last value of the request, and the current value of the request of ONU_i, respectively. $DT_{m_i}(n-1)$ and $DT_i(n)$ are the moving average of the last value of the downstream transmission length at instance $(n-1)$, and the current value of downstream transmission length sent by the OLT. Note that the traffic sent in the downstream channel is independent from that sent on the upstream channel.

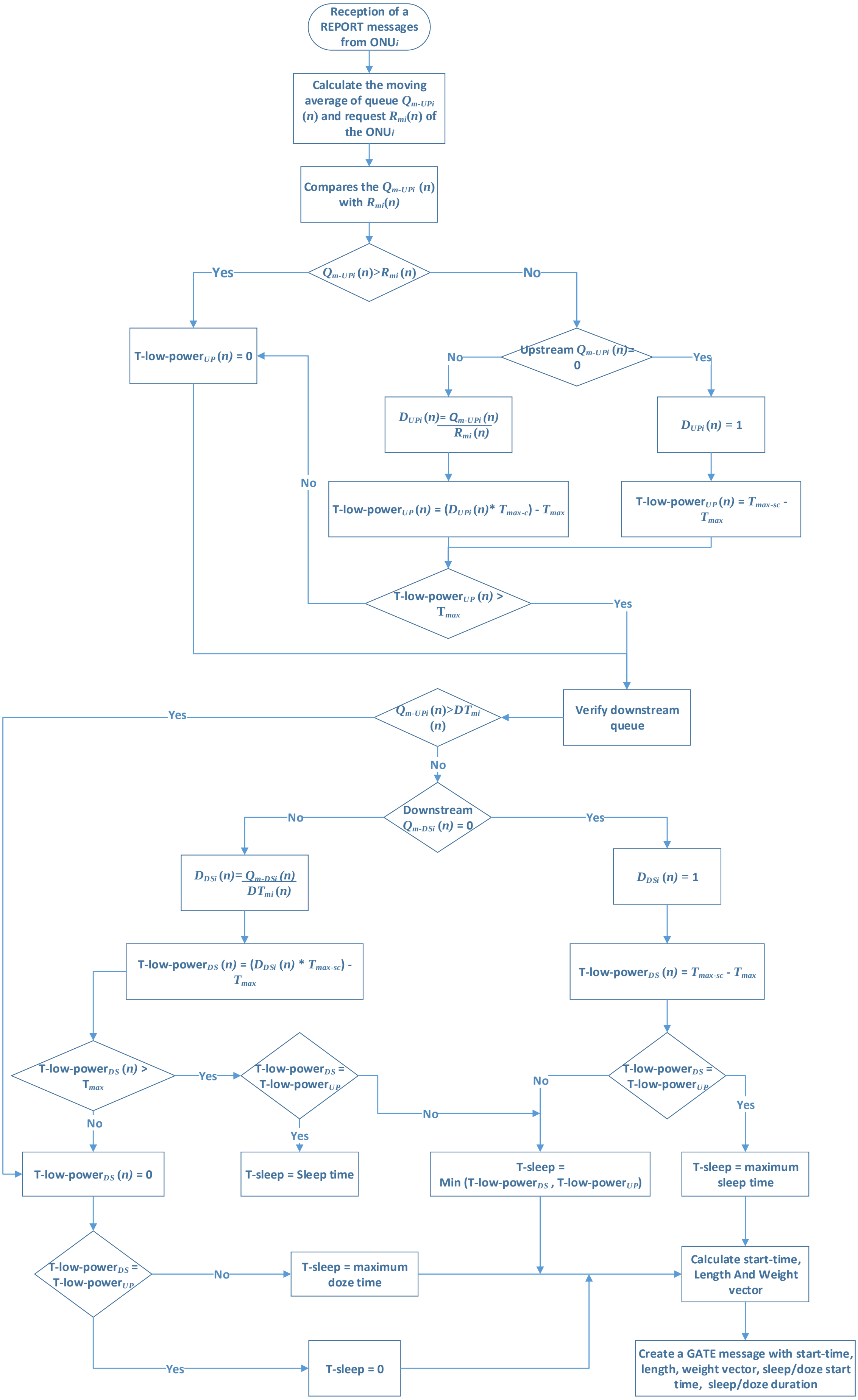


Figure 4-3 Flow diagram of bandwidth and energy management in the OLT.

The OLT then compares the predicted value of the queue (the moving average of the queue) of the ONU with the request of the same ONU. If the ONU queue size is bigger than the request, the power saving for the next cycle will be zero (ONU remains active) because there will still be some packets remaining in the queue to be sent to the OLT. Then, the DBA computes the next start time and length (slot allocation) for the ONU, and updates the weight vector for the next cycle. In this case, the ONU is completely active and cannot enter in a low power mode (doze mode). Following that, the OLT compares the moving average of the queue of the OLT with the downstream transmission length of the same queue. The OLT performs the same process as described above, which for the queue size bigger than the downstream transmission length, the ONU cannot enter in sleep mode, and thus both transmitter and receiver of ONU must remain active.

On the other hand, if the queue size of the ONU is less than the request and the queue size of the OLT is less than the downstream transmission length then the ONU can enter into doze or sleep mode. In this case, the OLT analyzes the queue of the ONU and the OLT for performing the doze and sleep mode. Depending on the decision arising from the queue size, the OLT will put the ONU into the sleep or doze mode for the zero queue size or any value of the queue. For the zero queue size, the OLT calculates the $T - low - power$ upstream and downstream as follows:

$$T - low - power_{UP/DS-Q=0} = \left(D_{UP/DS} T_{max-sc}\right) - T_{max} \qquad 4.1$$

where the factor D_{UP} is defined as the ratio between the average queue size and the average request, and D_{DS} is the ratio between the average queue size and the average downstream transmission length (see Equation 4.2). By definition, D_{UP} and D_{DS} are always less than one ($0 < D_{UP}$ or $D_{DS} \leq 1$), which can adjusting the sleep period between minimum and maximum sleep cycle. For the zero queue size, the value of D_{UP} and D_{DS} are equal to one (Equation 4.1). T_{max-sc} is a maximum sleep cycle, which is used to maintain the link health, with a maximum value of 50 ms ($2\ ms \leq T_{max-sc} \leq 50\ ms$). For bigger values of T_{max-sc}, ONUs can stay longer in sleep mode and vice versa. T_{max} is the maximum cycle time, i.e. the maximum period that an ONU can transmit its traffic in upstream channel.

Likewise, for average queue sizes bigger than zero, the value of factors D_{UP} and D_{DS} are computed based on the moving average of the instant queue of the ONU and its instantaneous request, and the moving average of the instantaneous queue of the OLT and its instantaneous downstream transmission length (see Equation 4.2). Then we can evaluate the power saving by computing $T-low-power_{UP}$ and $T-low-power_{DS}$ based on D_{UP} and D_{DS} and T_{max-sc} (equations 4.3 and 4.4).

$$D_{UP} = \frac{Q_{m-UP_i}(n)}{R_{m_i}(n)},\ D_{DS} = \frac{Q_{m-DS_i}(n)}{DT_{m_i}(n)} \qquad 4.2$$

$$T-low-Power_{UP-Q\neq 0} = (D_{UP_i} * T_{max-sc}) - T_{max} \qquad 4.3$$

$$T-low-Power_{DS-Q\neq 0} = (D_{DS_i} * T_{max-sc}) - T_{max} \qquad 4.4$$

In the case of power saving, by comparing the ONU and OLT queues, the OLT should select the minimum value of $T-low-power_{UP/DS}$ and assign the minimum sleep time, which can put the ONU in doze mode or sleep mode.

$$T-sleep = \min(T-low-power_{UP}, T-low-power_{DS})$$

The algorithm is designed to choose the minimum value of both cycles for a specific ONU. Although this can waste energy that could be saved, it can also minimize the network delay. Therefore, the minimum value is selected to ensure a good trade-off between energy and delay even when the sleep cycle in the doze mode is more than the sleep mode, and vice versa.

Once the $T-low-power$ for upstream and downstream is calculated, the OLT creates a GATE message with the required information and sends it to the ONU.

The second phase of the algorithm is related to the actions that must be executed in the ONUs. Figure 4-4 shows the flow chart of the DBA algorithm executed at the ONUs. In each cycle, the ONU extracts the information of the GATE message fields and verifies the accumulated traffic in the queue. The information includes transmission start time, granted period of transmission, weight vector, sleep and doze start time, and sleep and doze duration.

The weight is calculated based on the weight vector ($\overrightarrow{\Phi}$) that is carried by the GATE messages (i.e. $\overrightarrow{\Phi} = \Phi_1, \Phi_2, \Phi_3, \dots, \Phi_n$) [84]. With the weight vector, each ONU knows the current state of the bandwidth assignment and thus each ONU can calculate its own required transmission window size for the next transmission cycle. Such parameter is a proportional weight, set up according to each ONU guaranteed bandwidth agreement. The maximum transmission window size for the next cycle is calculated as follows:

$$W_i(\mathrm{n}+1) = \frac{\Phi_i^{conf}}{\Phi_i^{conf} + \sum_{j=1;j\neq i}^{N} \Phi_j(n)} W_{max}$$

where $W_i(\mathrm{n}+1)$ is the transmission window size for next cycle of the ONU_i, Φ_i^{conf} is the initial weight value for ONU_i, $\Phi_\mathrm{j}(n)$ is the remaining weight value in cycle n of the rest of stations, and W_{max} is the maximum transmission window size of all ONUs.

In the next step, ONU_i calculates the next requested transmission size $R_i(n+1)$, by comparing the next transmission window size with the current queue size of the ONU_i, and picking the minimum value between both values. The advantage of this is that ONU can allocate the right time slot size and avoid over-granting.

$$R_i(n+1) = \min(W_i(n+1), \mathrm{Q}_{UP_i}(n+1))$$

Then, the new weight $\Phi_i(\mathrm{n}+1)$ of ONU_i for updating the weight vector for the next cycle is computed as follows:

$$\Phi_i(\mathrm{n}+1) = \frac{R_i(n+1)\left[\Phi_i^{conf} + \sum_{j=1;j\neq i}^{N}\Phi_j(n)\right]}{W_{max}}$$

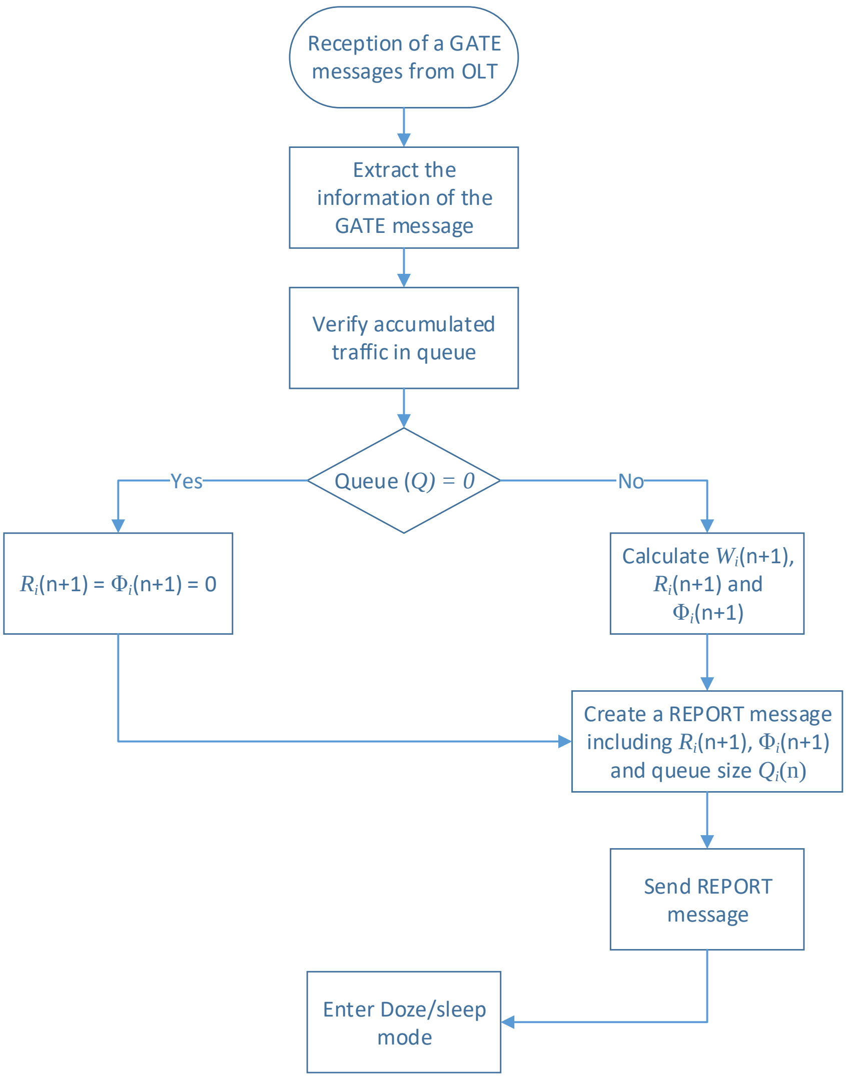


Figure 4-4 Flow diagram of bandwidth management in ONUs.

Finally, the ONU analyzes the sleep start time and doze start time of the GATE message to see if there is a non-zero value assigned to these fields. If a non-zero value is determined in any of them, the ONU enters into sleep or doze mode for a duration specified in the sleep duration or doze duration fields, and wakes up at the ends of the period.

The REPORT message generated by the ONU_i includes 1) the next requested transmission size $R_i(n+1)$; 2) the new weight for the next cycle $\Phi_i(\mathrm{n}+1)$; and 3) the instantaneous queue size (current state of the queue) $\mathrm{Q}_i(n)$. The same process is performed by each ONU. Therefore, ONUs are responsible for scheduling dynamically its transmission window size.

Algorithm 1: Pseudocode of the enhanced DDSPON DBA algorithm

```
Calculate the moving average of queue (Q_{m-UP_i}(n)), (Q_{m-DS_i}(n)), request (R_{m_i}(n)) and
downstream transmission length (DT_{m_i}(n)) for ONU_i
IF (R_{m_i}(n) > Q_{m-UP_i}(n)) & (DT_{m_i}(n) > Q_{m-DS_i}(n))
   IF (Q_{m-UP_i}(n) = 0) & (Q_{m-DS_i}(n) = 0)
      ONU_i goes to sleep mode for maximum time
   ELSE
      calculate D_{UP_i}(n) = Q_{m-UP_i}(n) / R_{m_i}(n) & D_{DS_i}(n) = Q_{m-DS_i}(n) / DT_{m_i}(n)
      T-low-power_{UP}(n) = (D_{UP_i}(n) * T_{max-sc}) - T_{max}
      T-low-power_{DS}(n) = (D_{DS_i}(n) * T_{max-sc}) - T_{max}
        IF (T-low-power_{UP} > T_{max}) & (T-low-power_{DS} > T_{max})
           T-sleep = min(T-low-power_{UP}, T-low-power_{DS})
           IF T-sleep_{min} = T-low-power_{UP}
              ONU_i goes to doze mode
           ELSE
              ONU_i goes to sleep mode
        ELSE IF (T-low-power_{UP} > T_{max}) & (T-low-power_{DS} < T_{max})
           ONU_i goes to doze mode
        ELSE
           ONU_i remains in active mode
ELSE
   T-low-power_{UP}(n) = 0
   T-low-power_{DS}(n) = 0
   T-sleep = 0
   ONU_i remains in active mode
```

The scheduling process is executed without the need to wait until all the reports arrive from the ONUs to the OLT, as is the case in on-line DBAs. In addition, by collecting the weight vector, each ONU is able to get a global view of the state of the load of other ONUs.

In order to perform the doze or sleep mode in the ONUs, we propose Algorithm 1 to be implemented in the OLT.

4.1.2 Extension of MPCP control frames for the energy-efficient DBA algorithm

Our algorithm requires an extension of the MPCP, because some extra information must be carried in the control frames. Apart from the weight vector ($\vec{\Phi}$) introduced by the DDSPON algorithm, we add extra information for the current state of the queue (instantaneous queue size) $Q_i(n)$ in the REPORT message, in order to allow to each ONU to calculate its own weight based on the traffic received from the users and the traffic present in the queue. This additional information must be sent to the OLT, so it can compute the next ONU sleep or doze time and evaluate the amount of energy that can be saved. Figure 4-5 shows the extension of the REPORT message in the DDSPON algorithm for the energy saving mechanism.

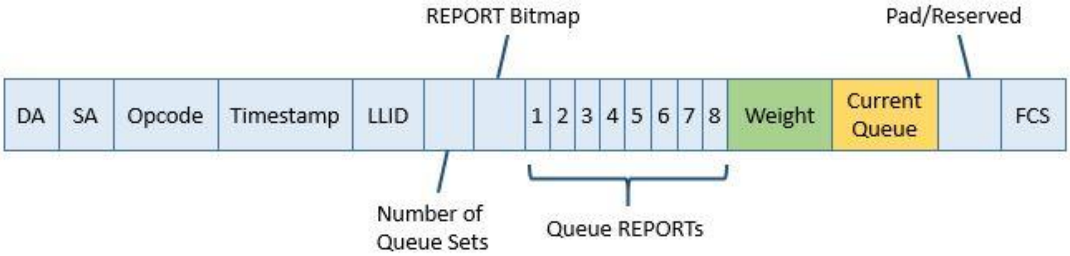


Figure 4-5 Current (green color) and extended field (yellow color) of the REPORT message of the DDSPON algorithm for the energy saving mechanism.

It is also necessary to extend the GATE messages with four extra fields carrying sleep start time and doze start time, sleep duration and doze duration, as shown in Figure 4-6. Sleep and doze start times indicate ONUs when to power-off its receiver and transmitter (sleep mode) or only transmitter (doze mode), respectively. Sleep and doze durations specify the period that the receiver and transmitter (in sleep mode) or only the transmitter (in doze mode) must be OFF during the specified time and wake-up ONU when the sleep or doze cycle elapses. The GATE message also includes the weight vector (for the next cycle), start time, and timeslot allocation. The start time indicates the data transmission start time and timeslot allocation indicates data transmission duration.

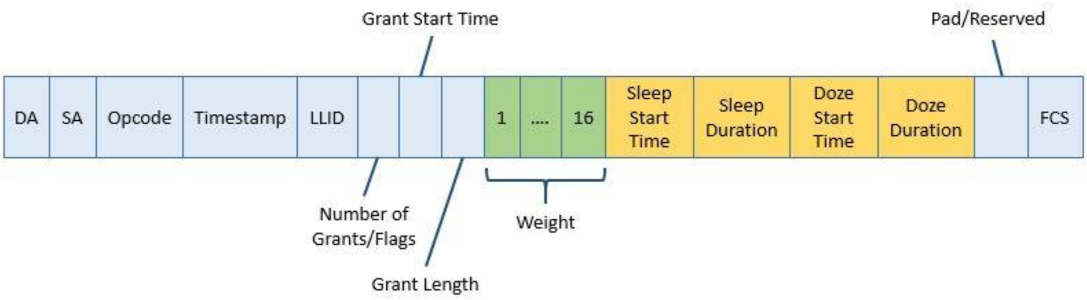


Figure 4-6 Current (green color) and extended fields (yellow color) of the GATE message of the DDSPON algorithm for the energy saving mechanism.

4.1.3 Power Saving State Diagram at ONUs

The power saving state diagram in Figure 4-7 shows the transitions of the ONU state between sleep and active. Transition among the states should be properly designed to maximize energy saving without degrading the performance of the access network. The state diagram is composed of four states:

1) *All-awake* is an active mode, in which the transmitter and receiver of an ONU remain on (TRx-ON). This is a state of normal operation, without power saving.
2) *All-sleep* is a sleep mode, in which the transmitter and receiver of an ONU are turned off (TRx-OFF).
3) *Transmitter-sleep* is a doze mode, in which the transmitter of an ONU is turned off (Tx-OFF), and the receiver remains on.
4) *Receiver-sleep*, in which the receiver of an ONU is turned off (Rx-OFF) and the transmitter remains on.

Figure 4-7 presents the operation of each power-saving function by describing which ONU element, transmitter (Tx) and receiver (Rx), can be switched off. Let's suppose the ONU is in state doze mode.

- The transition from transmitter-sleep to other states can be triggered by the arrival of the GATE message to the ONU in downstream traffic, where the Rx is ON and Tx is OFF. Thus, the ONU can receive control and data frames from OLT, while it cannot transmit any frames in the upstream channel. The state will be changed depending the values of $T-low-power_{UP}$ and $T-low-power_{DS}$.
- The ONU transmitter and receiver can be put into sleep mode for any values of $T-low-power_{UP}$ and $T-low-power_{DS}$ and wake-up when the values of the $T-low-power_{UP}$ and $T-low-power_{DS}=0$. The ONU goes to all-

sleep state, if $T-low-power_{UP}$ and $T-low-power_{DS}$ are greater than zero and both of them are greater than T_{max}.

- The transition from transmitter-sleep to receiver-sleep will be changed when $T-low-power_{UP}=0$ and $T-low-power_{DS}>0$ and $T-low-power_{DS}>T_{max}$. In this state, the receiver will be OFF, because there is no data to transmit in the downstream channel.
- Finally, both transmitter and receiver of the ONU are in awake state (TRx are active) when the value of $T-low-power_{UP}$ and $T-low-power_{DS}$ is less than T_{max} and the local time of the ONU is equal to the transmission start time (start-time, T_{st}). In this state the ONU receives and transmits control and data frames.

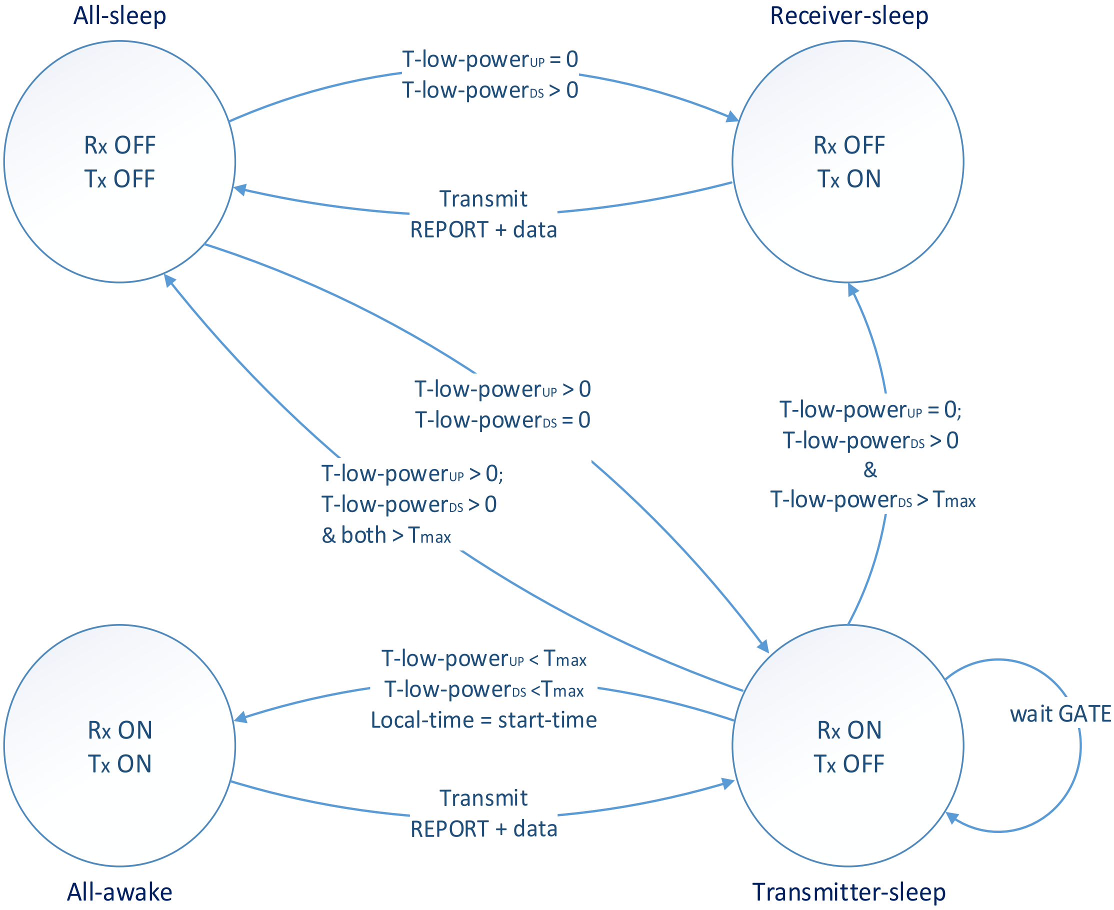


Figure 4-7 ONU state diagram.

4.1.4 Operational scheme

This section describes the operations performed by ONUs and OLT to minimize power consumption.

The sleep and doze modes of ONUs are triggered by the GATE message that is received from the OLT, when certain conditions in downstream and upstream traffic are detected. As illustrated in Figure 4-8, upon receiving of the GATE message (ONU_i will discard non-matching GATE message), ONU_i analyses the associated fields. After that, ONU_i starts sending its buffered data up to the size of the granted time-slot, while it keeps receiving data from the users and accumulates in the buffer. At the end of the granted time-slot, ONU_i generates its REPORT message and ends the upstream data transmission. Once the data and REPORT messages are sent, ONU_i then enters into sleep or doze mode for a duration specified by the GATE message. During this time, ONU_i continues updating the queue size of received packets by using the moving average technique. When the sleep or doze time finish, ONU_i wakes up to receive the next GATE message and downstream traffic, and transmits upstream traffic from/to the OLT in the next cycle, respectively.

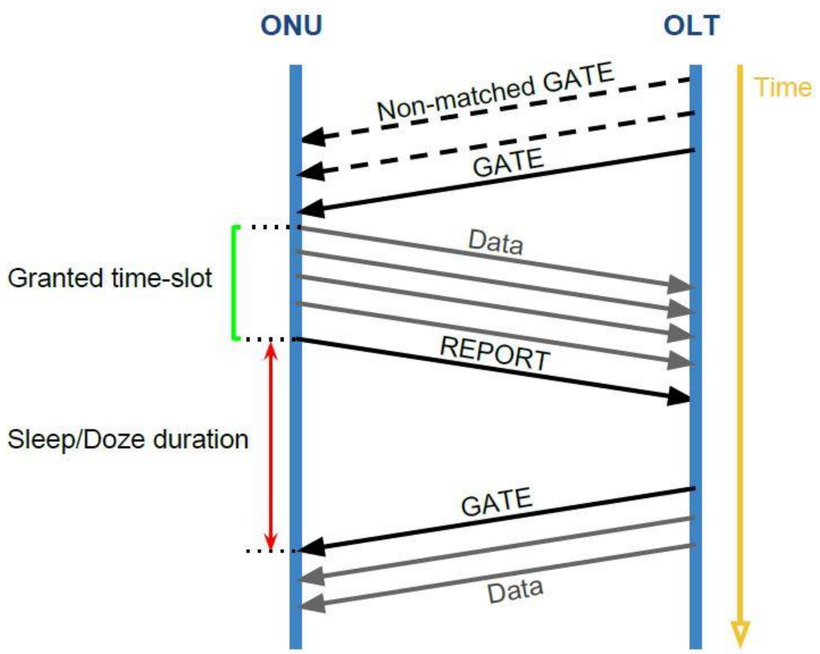


Figure 4-8 Message exchange between the OLT and an ONU.

As shown in Figure 4-9, upon receiving a REPORT message, the OLT analyses the REPORT message and compares the predicted value of the moving average of the queue $Q_{m-UP_i}(n)$ of the ONU_i with its moving average of the request $R_{m_i}(n)$, and simultaneously the OLT compares the predicted value of the moving average of the queue $Q_{m-DS_i}(n)$ of the OLT with its moving average of the downstream transmission length $DT_{m_i}(n)$. The OLT then compares the moving average of the queue size $Q_{m-UP_i}(n)$ of ONU_i with the corresponding moving average of the queue size $Q_{m-DS_i}(n)$ in the OLT (we implement sixteen queues, each one assigned to an ONU). The queue entities in the OLT are allocated to the downstream channel, while the queue entity in each ONU is related to the upstream channel, as shown in Figure 4-9. Based on the comparison of these queues status, the OLT then takes the decisions related to energy saving in ONU_i (as discussed in section 4.1.1). During the sleep and doze period, the OLT keeps monitoring the amount of incoming traffic received to each queue of the OLT. In this case, the OLT can take the decision to perform the required power saving technique for the subsequent cycle in order to optimize trade-off with energy and delay.

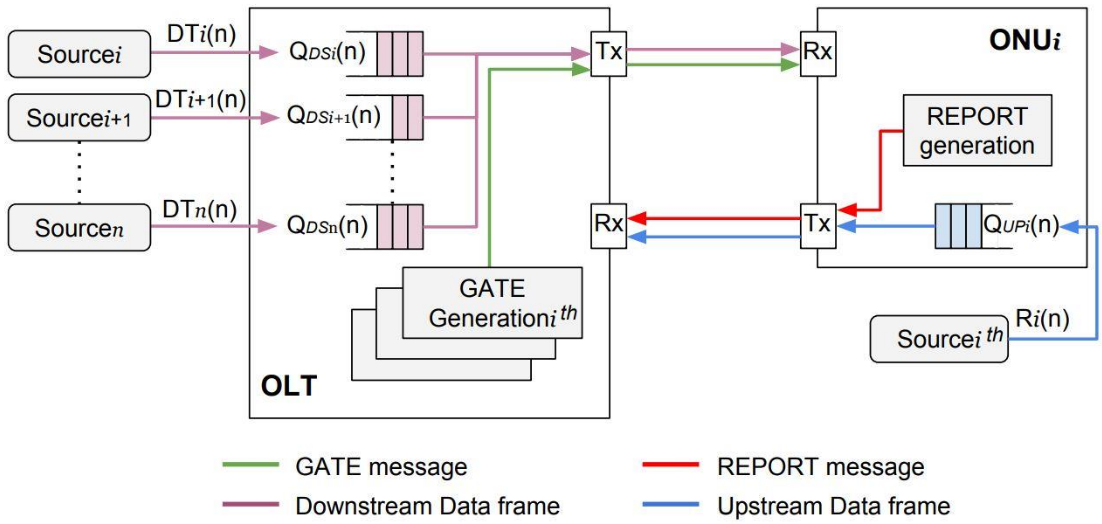


Figure 4-9 Operational scheme of power saving algorithm for the ONU and OLT.

4.2 Results and Discussion

In this section we evaluate the effectiveness and performance of the proposed power saving algorithm + DDSPON against the plain DDSPON version in terms of power consumption and average packet delay. To evaluate the proposed algorithm, we run extensive simulations, using our developed simulation network model with OPNET Modeler. The goal is to show that our scheme can reduce the power consumption of the PON network while guaranteeing the QoS requirement of real-time services. The maximum delay in this environment is bounded to 10 ms from ONU to the OLT and vice versa [26].

In our testbed, the initial values of the simulation parameters have been configured as follows: the number of ONU is set up to N = 16, and ONUs are placed at distance of 18 to 20 kilometres from the OLT. Both upstream and downstream channels operate at a data rate of 1 Gb/s; the maximum cycle time is set to 1 ms; the α value varies from 0.3 to 0.9 and the sleep cycle varies from 2 to 50 ms. We assume that the capacity of the queue in each ONU and OLT is infinite. The power consumption of an ONU in active mode is equal to 5.052 watt, in doze mode is 3.85 watt and in sleep mode is 0.75 watt as reported in [18] and [48]. The simulation is performed in two scenarios. In the first scenario, sources generate a constant traffic being the packet length of 1518 bytes, while in the second scenario, the traffic is self-similar with a Hurst parameter of H = 0.7 and packet sizes are uniformly distributed from 64 to 1518 bytes. Table 4-2 lists the network and protocol parameters used in our simulations.

Table 4-2 Simulation parameters.

Parameters	**Value**
Link rate	*1 Gb*
ONU power consumption (active)	*5.052 W*
ONU power consumption (doze)	*3.85 W*
ONU power consumption (sleep)	*0.75 W*
Maximum cycle time	*1 ms*
α	*0.3, 0.6, 0.9*
Initial weight value (Φ_i^{conf})	*1/N = 0.0625, N=16*
DBA algorithm for upstream scheduling	*DDSPON*
DBA algorithm for downstream scheduling	*Round-robin*
Traffic source	*Constant (CBR)/self-similar*
Network topology	*Tree*
OLT distance (d)	*18 < d < 20 km*
Sleep cycle (T_{max-sc})	*[2, 3, 4, 5, 10, 20, 30, 40, 50] ms*

The initial weight value for each ONU is set as a $\Phi_i^{conf} = 0.0625$ (which is $\frac{1}{N}$). Note that, when an ONU is in a sleep mode, the weight of the sleeping ONU is given to the rest of ONUs. Hence, a higher fraction of bandwidth can be allocate to other ONUs.

The simulations were performed with asymmetric load: the upstream load is not equal to the downstream load. In addition, upstream and downstream sources are independent from each other. Upstream and downstream network traffic is balanced among ONUs (each ONU sends $54\ Mbits/s$ upstream traffic and receives different traffic load from the OLT: by doing different simulations with downstream traffic of 10, 30 and $50\ Mbits/s$). All graphs show the average values and 95% confidence interval.

The evaluation is based on comparing the energy consumed in the power-saving algorithm + DDSPON and the energy consumed by the plain DDSPON. The energy consumed in an ONU is calculated as the sum of the energy consumption during each cycle (energy consumption in active, doze and sleep mode). E_{ONU-PS_i} (the energy consumed by power saving algorithm) is composed of several cycles and the total energy consumption of ONU_i is the sum of each cycle.

$$E_{ONU_{i-active}} = \sum_{1}^{n} P_{i-active} \times T_{i-active}$$

$$E_{ONU_{i-doze}} = \sum_{1}^{n} P_{i-doze} \times T_{i-doze}$$

$$E_{ONU_{i-sleep}} = \sum_{1}^{n} P_{i-sleep} \times T_{i-sleep}$$

$$E_{ONU-PS_i} = E_{ONU_{i-sleep}} + E_{ONU_{i-doze}} + E_{ONU_{i-sleep}}$$

where the parameters P_{i-doze}, $P_{i-sleep}$, and $P_{i-active}$ represent the power consumption of an ONU_i in doze, sleep and active mode, respectively. The parameter T_{i-doze} and $T_{i-sleep}$ represents the state transition overhead time of the ONU_i , and $T_{i-active}$ represents the active time of ONU_i. The synchronization time for changing from sleep mode to active mode is 0.125 *ms* and for changing from doze mode to active mode is 760 *ns* as reported in [18]. Parameter *n* is the total number of cycles that an ONU can handle during the simulation. Following that, the percentage of energy-saving η is calculated as follows:

$$\eta = (1 - \frac{E_{ONU-PS_i}}{E_{ONU-NPS_i}})\%$$

where the parameter E_{ONU-PS_i} represents energy consumption for the power saving mechanism and the parameter $E_{ONU-NPS_i}$ represents for the non-power-saving scenario.

In our evaluation we first analyse the influence of the weight of the moving average used in several estimations (the instantaneous queue $Q_{m-UP_i}(n)$ of ONU_i and its request $R_{m_i}(n)$, and instantaneous queue $Q_{m-DS_i}(n)$ of the OLT and its downstream transmission length $DT_{m_i}(n)$). In order to find an optimum value of α for the power saving algorithm, we vary the value of the weight and runs different number of simulations under different weight values (α = 0.3, 0.6, and 0.9) for both upstream and downstream scheduling. In this set of simulations, we assume the traffic input in upstream direction for each ONU is set to 54 *Mbps* and for OLT is 50 *Mbps* (the highest traffic for the OLT is chosen to evaluate the network in the worst case). The constant traffic (CBR) is selected as an input traffic source and the sleep cycle of 10 *ms* selected as an average sleep cycle to comment result for all.

4.2.1 Evaluation of the α Parameter

Figure 4-10 shows the comparison between the moving average of the instantaneous queue and the request of the ONUs in the upstream direction. The main conclusion is that the higher weight value of the moving average (α = 0.9) performs much better than the lower weight value (α = 0.3) in sense of power saving and delay. Higher weight shows a shorter queue and consequently, packets experience less delay. In addition, by comparing $Q_{m-UP_i}(n)$ and $R_{m_i}(n)$, we observe that there are more matching points between Q_m and R_m for α = 0.9 than 0.6 and 0.3 (matching points represents the occurrence of sleep period in the power saving algorithm + DDSPON). As discussed earlier in section 4.1.1, when $Q_{m-UP_i}(n) \leq R_{m_i}(n)$, we can observe power saving in the network. Therefore, α = 0.9 performs better than other two α values in terms of power saving and delay.

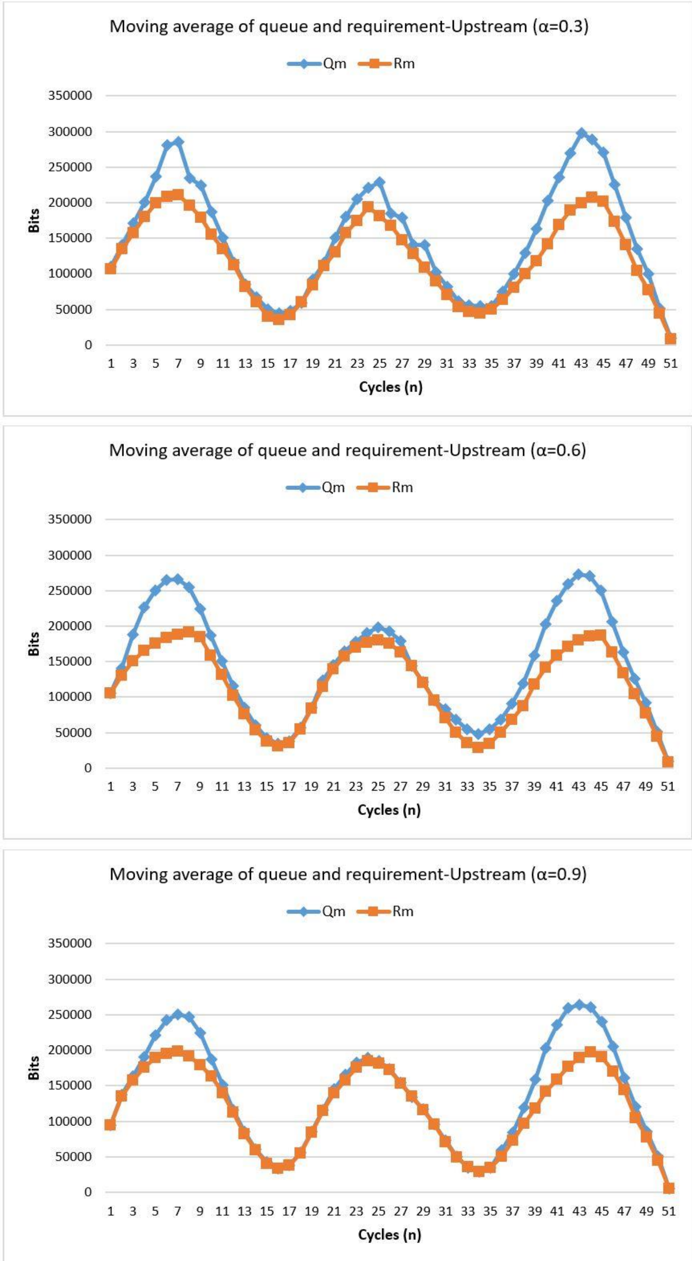


Figure 4-10 Comparison of the moving average of the queue and the request in the upstream direction (ONU side) for different values of α.

Figure 4-11 shows the comparison between the moving average of the instantaneous queue and downstream transmission length of the OLT in the downstream direction. The results show a behaviour that is similar to the upstream case. When comparing the moving average of $Q_{m-DS_i}(n)$ and $DT_{m_i}(n)$, α = 0.9 shows better results than α = 0.6 and 0.3. Based on obtained results from different weight values, we now can evaluate our algorithm under different circumstances.

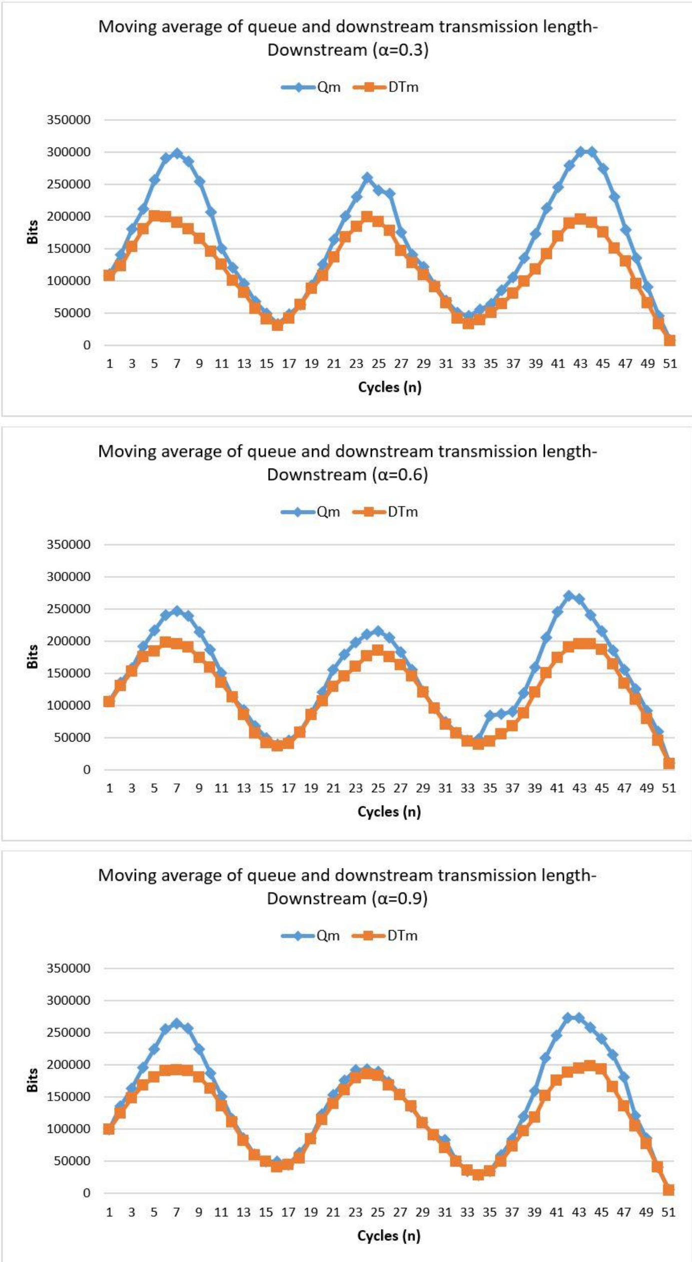


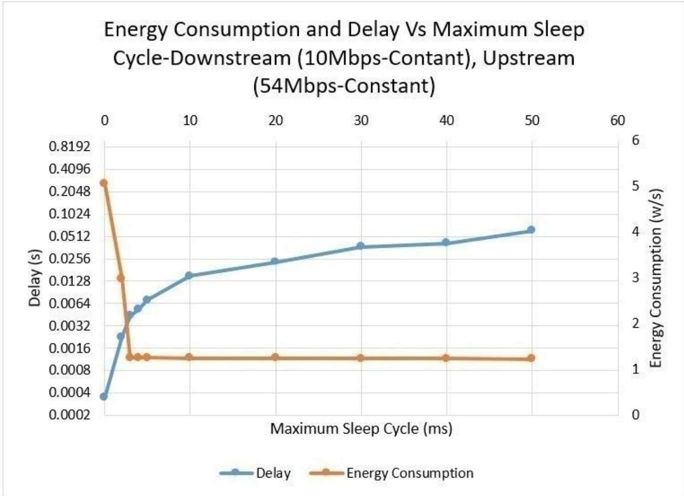
Figure 4-11 Comparison of the moving average of the queue and the downstream transmission length in downstream direction (OLT side) for different values of α.

4.2.2 Evaluation with CBR Traffic

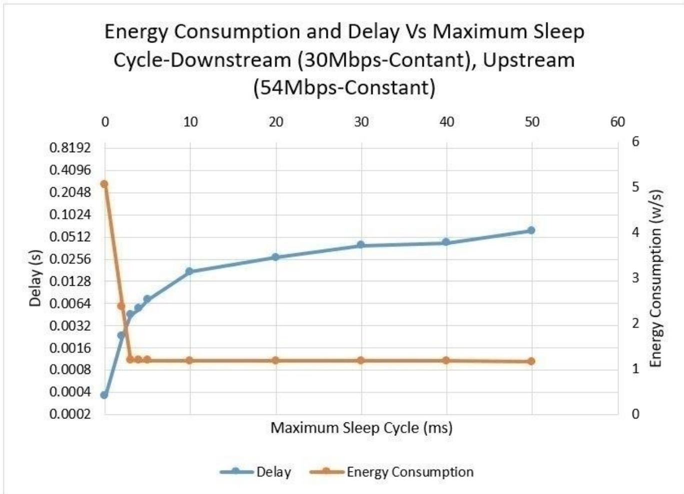
We now show the simulation results for the scenario 1, where the sources generate constant traffic (CBR). Figure 4-12 shows the average power consumption and packet delay for the sleep cycle (sleep and doze mode) of the ONU compared to the active mode of the same ONU (an ONU with always ON-TRx) in function of: 1) sleep cycle and 2) offered load in the downstream direction (10, 30, 50 *Mbps*).

We can see that by increasing the maximum sleep cycle time, the average packet delay increases for the energy saving case and the power consumption is reduced.

a)



b)



c)

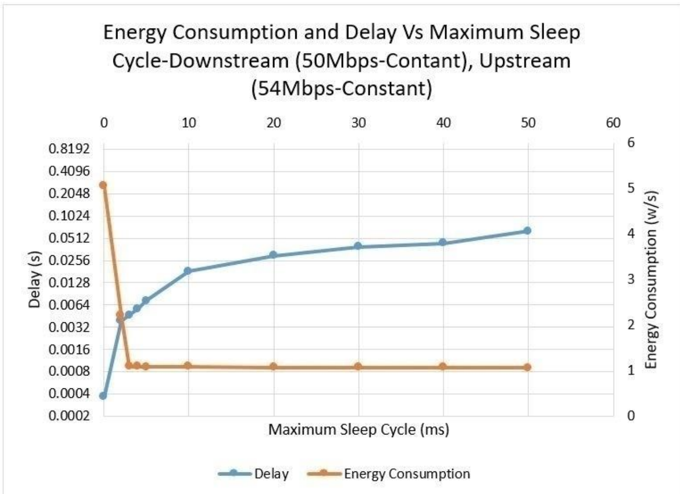


Figure 4-12 Energy consumption and average delay versus sleep cycle time for power saving and non-power saving mechanism (constant source), for different sleep cycles.

With the power-saving mechanism, at an offered load of 10 *Mbps* (Figure 4-12 - a) and sleep cycle 2 *ms*, the energy consumption reaches 2.98 *watt/s* with an average packet delay of 2 *ms*. For the same load, with an sleep cycle between 3 *ms* and 50 *ms*, energy consumption varies from 1.25 to 1.22 *watt/s* and the delay ranges from 2.2 *ms* to 60 *ms*, while the power consumption in non-power saving mechanism (an active ONU) is 5.052 *watt/s* with the average packet delay of 0.3 *ms*. Likewise, at load 30 (Figure 4-12 - b) at sleep cycle 2 *ms* to 50 *ms*, the power consumption decrease from 2.37 to 1.16 *watt/s* and followed by, the delay increases from 2.3 *ms* to 61 *ms*. In the similar way for load 50 *Mbps* (Figure 4-12 - c) at sleep cycle 2 to 50 *ms*, the power consumption decrease from 2.22 to 1.06 *watt/s* and the delay increases from 2.5 *ms* to 64 *ms*, respectively. For a given network loads (30 and 50 *Mbps)*, the packet must wait for a longer duration of time until they are allocated to be transmitted in the next transmission time slot.

Note that, in all these figures, the energy consumption does not have a significant variation, whereas the delay for sending the packets are increase dramatically for values of sleep cycle of 5 to 50 *ms*.

Figure 4-13 illustrates the cumulative distribution of delay and energy consumption in upstream and downstream direction for both power saving and non-power saving scenarios in order to find the optimal value of the sleep cycle for the power-saving scenario algorithm. Based on the obtained results from energy consumption and packet delay in upstream and downstream direction, we consider a sleep cycle of 5 *ms* as an optimal value for the power-saving scenario. The absolute value of delay is acceptable, because the delay for sending 100% of packets must be below of 10 *ms* to satisfy the QoS requirements [26]. For the sake of brevity, we show only the results obtained with an offered load of 50 *Mbps* at the OLT (downstream direction) and 54 *Mbps* at the ONU side (upstream direction).

Figure 4-14 illustrates the percentage of energy-saving η_{CBR} for both the power saving and non-power saving mechanisms as a function of the maximum sleep cycle, where the downstream offered load is 50 *Mbps* and the upstream load is 54 *Mbps*. The most outstanding result in terms of power saving is obtained with a sleep cycle of 50 *ms* but as we discussed in the CDF results, sleep cycle 5 *ms* is selected to fulfill the QoS requirements.

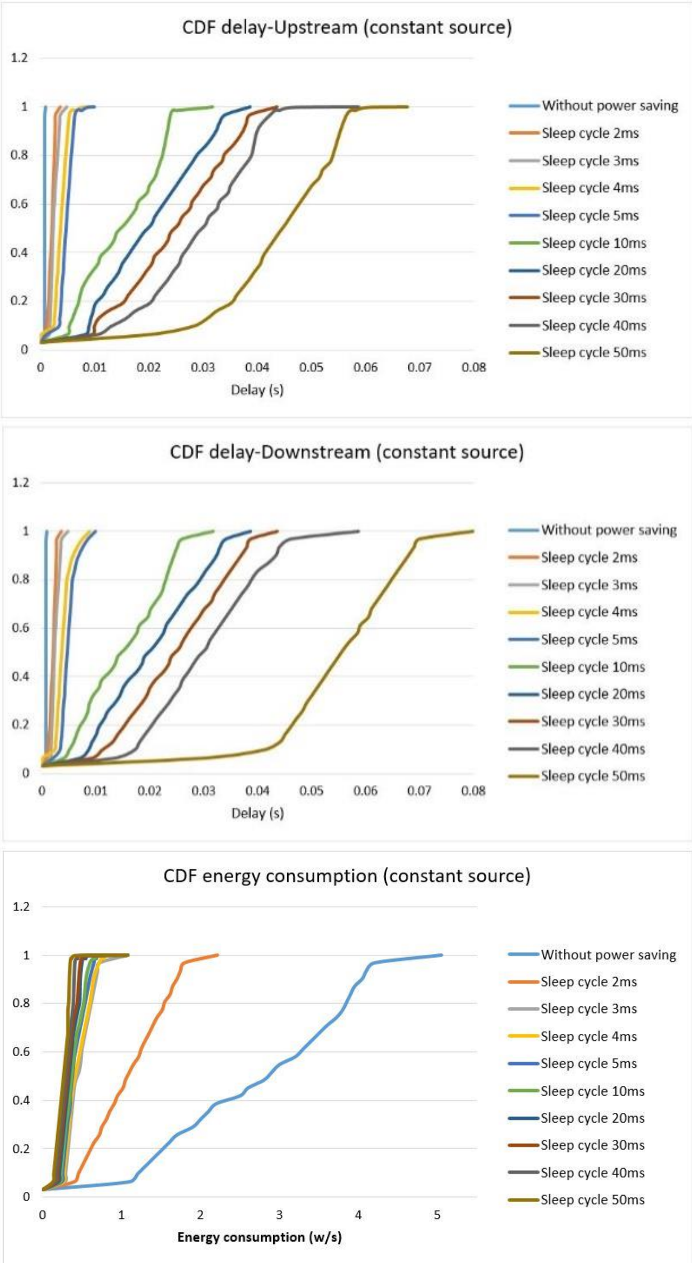


Figure 4-13 Upstream (top) and downstream (middle) CDF of the delay and the energy consumption (bottom) for power saving and non-power saving mechanism with constant traffic, for different sleep cycles.

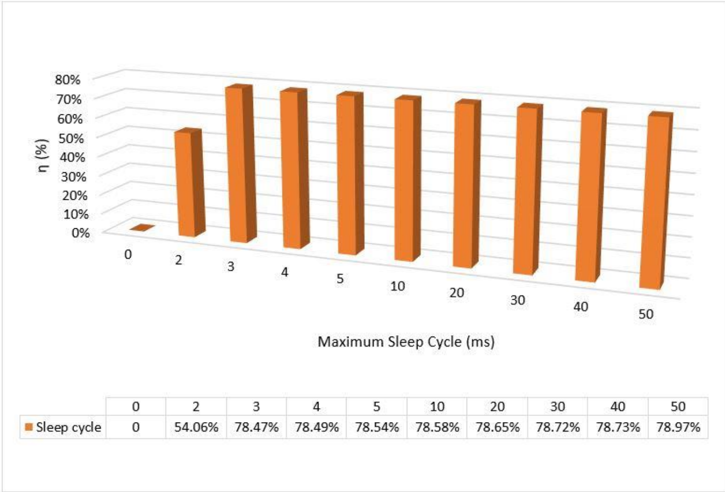


Figure 4-14 Percentage of energy saving (η_{CBR}) for both non-power saving (sleep cycle 0) and power saving (sleep cycle 2 to 50 *ms*) scenario.

4.2.3 Evaluation with Self-similar Traffic

We now show the simulation results with self-similar sources. As shown in Figure 4-15, similar to CBR scenario, the average energy consumption with self-similar traffic decreases by increasing the sleep cycle time, while the average packet delay increases. Compared to non-power saving, the energy consumption is reduced by half for the sleep cycle of 2 *ms* and by 80% for the sleep cycle of 3 *ms* to 50 *ms*, with a light slope. This reduction in energy consumption is more obvious at sleep cycle 2 *ms* for different downstream network load (10, 30, and 50 *Mbps*).

As demonstrated in Figure 4-15 - a, the network shows more energy consumption and average packet delay at load 10Mbps (from 2.81 to 1.01 *watt/s* and 2.4 *ms* to 53 *ms* for sleep cycle 2 *ms* to 50 *ms*), while more improvement can obtain at higher load (30 *Mbps* and 50 *Mbps*). For sleep cycle 2 ms to 50 ms, at load 30 *Mbps* (Figure 4-15 - b), energy consumption reduce from 2.21 to 0.9 *watt/s* with the average packet delay between 2.6 *ms* and 54 *ms.*

Likewise, at load 50 *Mbps* (Figure 4-15 - c), energy consumption reduces from 1.97 to 0.78 *watt/s* with the average packet delay between 2.9 *ms* and 55 *ms*. Self-similarity traffic results show more power saving for different sleep cycles and almost less packet delay compared to the CBR traffic results.

a)

b)

c)

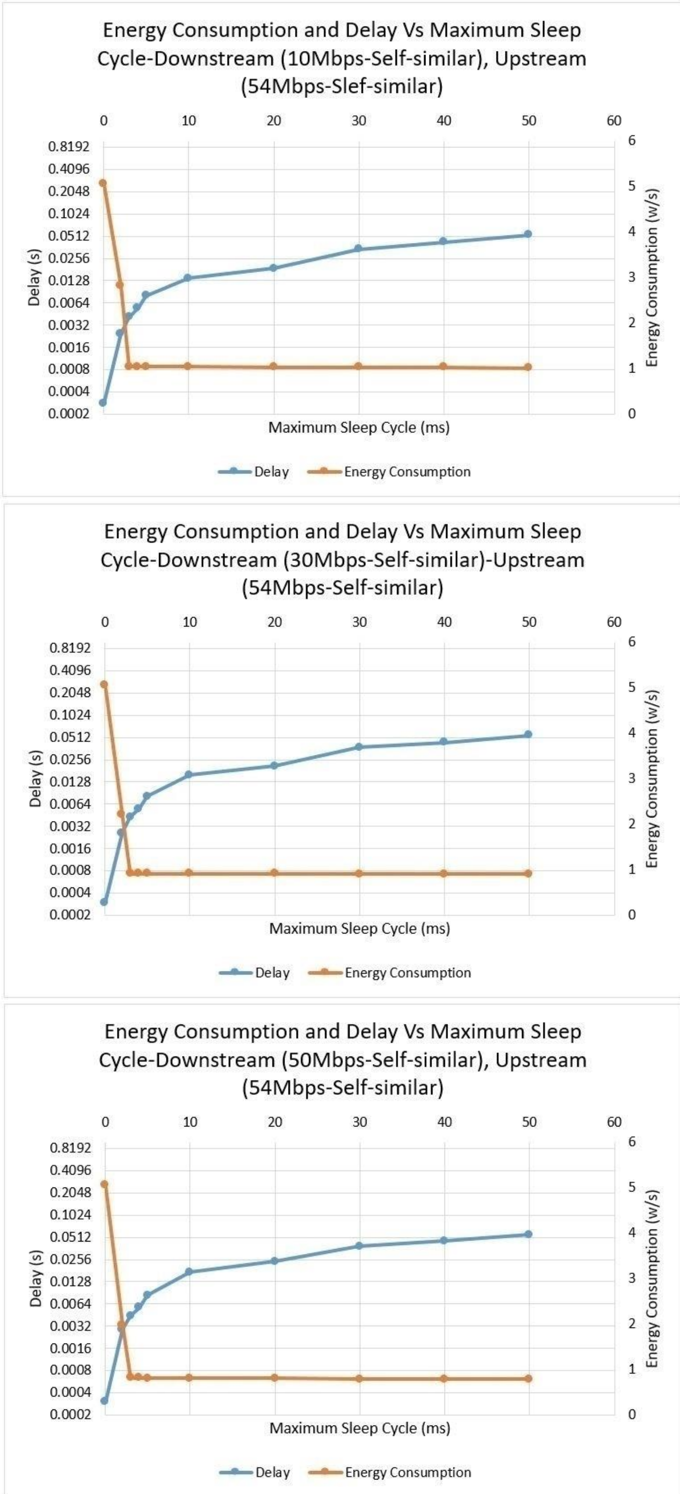


Figure 4-15 Energy consumption based on average delay and sleep cycle time for power saving and non-power saving mechanism (self-similar source), for different sleep cycles.

In order to find an optimum value of the sleep cycle in the proposed algorithm, the cumulative distribution delay in upstream, downstream direction, and energy consumption for both power saving and non-power saving scenarios are depicted in

Figure 4-16. The optimal sleep cycle (in sense of delay) for sending 100% of the packets (all transmitted packets receive to the destination under 10 *ms*) in self-similar source is 5 *ms*, which is similar to CBR source. For sleep cycle values of more than 5 *ms*, the delay for sending 100% of the packets increases dramatically (goes over 20 *ms*), violating the QoS requirements (10 *ms*).

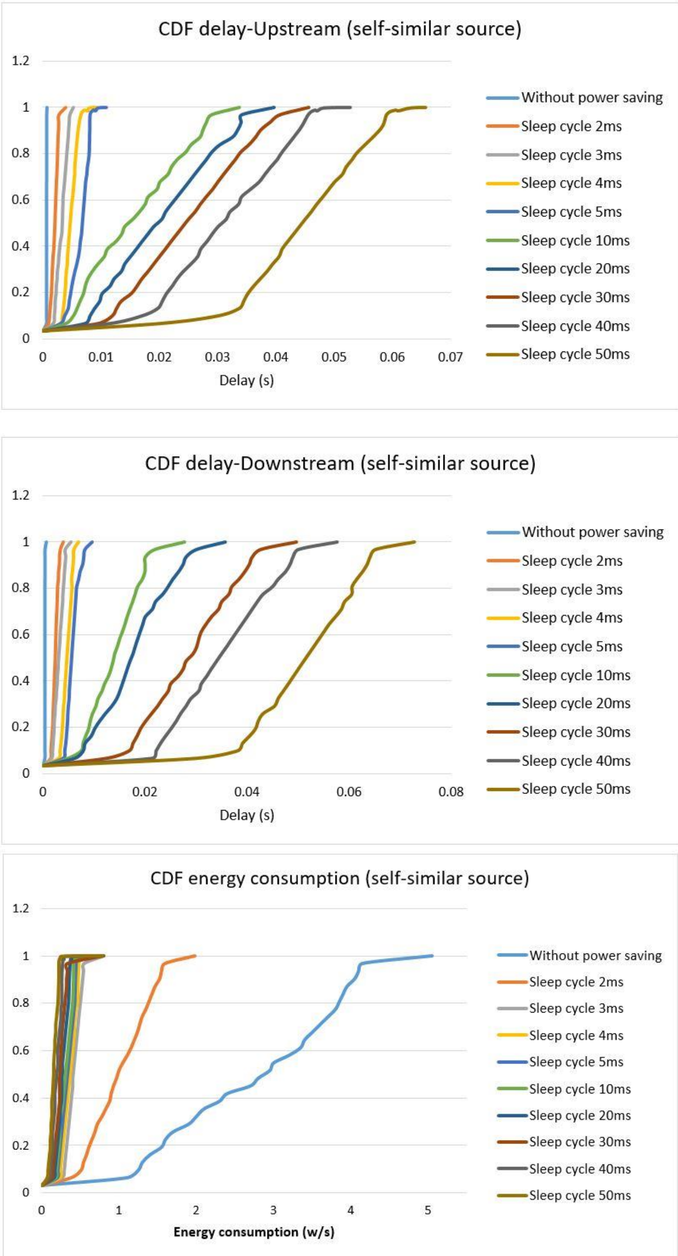


Figure 4-16 Upstream (top) and downstream (middle) CDF of the delay and energy consumption (bottom) for power saving and non-power saving mechanism (self-similar source), for different sleep cycles.

Figure 4-17 illustrates the percentage of energy saving ($\eta_{Self-similar}$) as a function of the maximum sleep cycle. The power saving mechanism shows more than 50% energy saving when sleep cycle is 2 *ms* and around 80% for value of sleep cycle from 3 to 50 *ms*. As we discussed in the CDF results, similar to the CBR traffic case, a sleep cycle of 5 *ms* is selected as an optimal value to fulfill the QoS requirements.

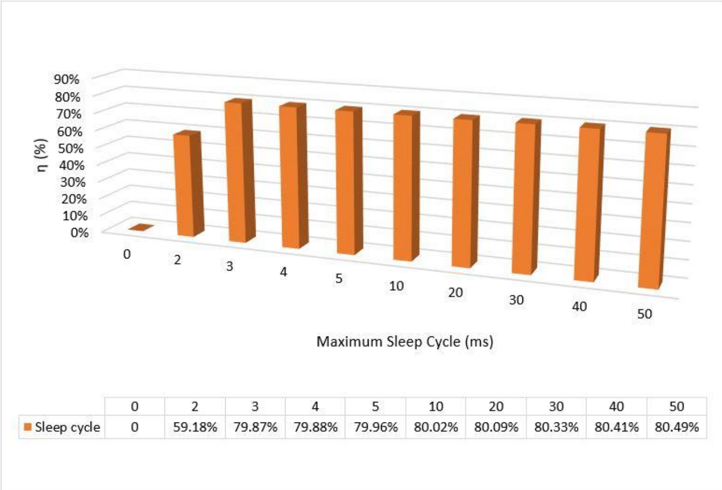


Figure 4-17 Percentage of energy saving ($\eta_{self-similar}$) for both power saving and non-power saving scenarios as a function of maximum sleep cycle.

4.3 Discussion

The performance of the system in both scenarios (CBR and self-similar) shows a similar behavior in terms of power saving and delay. The maximum percentage of power savings achieved with CBR traffic for sleep mode is 78.97%, whereas for self-similar traffic is 80.49%. The reason for this difference is probably the size of the packet: the packet size of CBR source is 1518 bytes (the maximum size of Ethernet frames), while the self-similar generates frames with size between 64 to 1518 bytes. Followed by that, we observe a reduction in queue size and packet transmission delay at both OLT and ONUs.

In comparison with other outstanding power saving algorithms ([43], [46], [47], [48] and [85]), our proposal achieves a good trade-off between energy and delay. The power saving achieved in [43], [46], [47] and [48] are less than 70% and in [85] is 84.7%, while [43], [46], [47] and [48] show less power saving with acceptable packet delay and [85] shows more power saving than our proposed algorithm but shows more delay, and thus not satisfying QoS.

This power saving algorithm can be developed in SDN-OLT and ONUs of the SDN-based SIEPON architecture and be managed by the SDN control plane. Power saving can be obtained by setting the laser transmitters and the receivers of both the OLT and ONUs to a sleep state when there is no traffic. The SDN controller could modify the behavior of the DBA and use the GATE/REPORT messages to switch the state of the ONU between sleep, doze, and active mode. The decision could be taken with the information transported in the REPORT messages about the queue status of the ONU for the upstream channel, and the OLT queue information for the downstream channel. The action of setting off transmitters affects the traffic patterns by grouping the data frames in burst, thus introducing delays and jitter, and affecting the QoS. That is why a global approach in which the SDN controller has both the control over the QoS and the energy saving, and can reach an appropriate trade-off.

4.4 Summary

We have presented a novel energy-saving DBA algorithm to improve the energy efficiency in the EPON network. Our proposal is based on the DDSPON DBA, which is enhanced by incorporating doze and sleep modes to reduce the energy consumption of the ONUs. The algorithm uses control frames (GATE and REPORT) for allocating the bandwidth and transition the ONUs from the active mode to the doze/sleep modes. The GATE message is used to force doze/sleep mode and the time that ONUs must wake up to transmit the data; the REPORT message transports the status of the queue and request of the ONUs.

In our study, the ONUs are switched to doze and sleep mode when $T - low - power_{UP}$ or $T - low - power_{DS}$ or both of them are bigger than the maximum transmission time (T_{max}). This only happened by comparing the queue size and the requested transmission length of the ONU with the queue size and downstream transmission traffic of the OLT. The doze mode and sleep mode ensure energy saving for both low load and high load cases. However, doze mode and sleep mode introduce a higher delay in the network, but our results demonstrate that the average delay values can remain below $10\ ms$ for both upstream and downstream direction, which is in acceptable range to support the QoS requirements for the access networks.

In addition, simulation results indicate that our proposed power saving algorithm + DDSPON outperforms than plain DDSPON algorithm in term of energy saving. In doze and sleep mode operation energy efficiency is improved in average by more than 78% for each ONU. The results presented in this chapter have been submitted for publication to [4].

5 Conclusions and Future work

This chapter summarizes the main achievements from this PhD dissertation. The chapter is divided in two parts as follows: conclusions are presented in Section 5.1 and future work is described in Section 5.2.

5.1 Conclusions

In this dissertation, we have focused on optical networks, specifically in EPON networks, in order to improve network manageability, operations and functionalities. Many researchers have proposed mechanisms and algorithms to optimize the bandwidth allocation, energy management and end-to-end quality of services. Few researchers have tackled the topic of applying the software-defined networking paradigm to passive optical networks. SDN-based architectures applied to optical networks have the potential to increase the flexibility and functionality of the network while reducing the capital expenditure and operational costs.

One of the most important goal of this thesis has been to analyze the EPON technology and propose and evaluate a novel architecture based on the SDN paradigms in order to decrease CAPEX and OPEX in the networks while optimize performance and functionalities. To reach our goal, we pursued four main objectives:

- The first objective was to analyze and virtualize an EPON architecture, distributing virtualizable and non-virtualizable parts. Following this objective, we defined a control plane for our architecture based on SDN principle. The SDN-EPON controller defined with a set of functions to control and manage the operations and functionalities of the OLT (SDN-OLT).
- The second objective was related to the MPMC sublayer of the OLT, to be splitted and partially virtualized and migrated to the centralized SDN-EPON controller. The control and management modules of the MAC control client and MAC client are migrated to the SDN-EPON controller. This virtualization enables the controller to manage and configure parameters of the MAC control client, SIEPON control manager functionalities, and MAC client policies (to name a few)

in the SDN-OLT. The controller can manage the MAC functions of the SDN-OLT by defining a set of rules.

- The third objective was to create a synchronous EPON SDN-based architecture in order to support tight timing communication between OLT and ONU. For this reason, we extended the MPMC sublayer with a virtual OpenFlow switch. The OF-switch is extended with synchronous ports to connect with the ONUs and internal processes of the SDN-OLT in order to retain the time-critical nature of the EPON network. Thus, the inherently asynchronous nature of SDN is extended to a synchronous SDN. In addition, the OF-switch can emulate the functionalities of the Control Parser, Control Multiplexer and Multipoint Transmission Control modules of the legacy OLT through the defined synchronous ports and a set of registers. The results show that the SDN approach reduces the network delay and increases the performance of the network in terms of throughput.
- Finally, we presented an energy-efficient solution to reduce power consumption in an EPON network with distributed DBA (DDSPON). The algorithm addresses the challenge of incorporating doze and sleep mode capabilities of an ONU to improve the energy efficiency of the EPON network. The proposed algorithm analyzes the queue status of ONUs and OLT to turn the ONU to the doze or sleep modes when there is no upstream and downstream traffic. In this scheme, the control frames (such as GATE and REPORT messages) are used to exchange doze and sleep mode operations between OLT and ONUs. The simulation results show significant energy saving of more than 78% at 80% network load while maintaining the average delay within acceptable range for delay-sensitive applications.

All of these goals allow to improve the performance and operation of EPON networks. We demonstrated our proposed solutions on the EPON technology that can help companies and research organizations to optimize, evaluate and making better cost-effective decisions on how such technologies can be designed and implemented.

5.2 Future work

This section presents several future work directions that can be derived from the research presented in this document. Some specific areas are listed below, where future work can be carried out to extend the work presented in this PhD:

5.2.1 SDN-EPON architecture for 5G EPON network

A point to explore is how to apply the SDN-EPON architecture to the 5G scenario, where multiple Radio Access Networks (RANs) are converged to the PON nodes at the network edge to deal with the 5G application and services, together with the DDSPON-based power saving solution to minimize the energy consumption of EPON networks without imposing additional network delay. For this, we can define a 5G RAN Manager module in the SDN-EPON Controller to provide a set 5G RAN functionalities for the OpenFlow switch of the SDN-OLT. The 5G RAN Manager keeps information about admission control for connecting different RANs and frequency bands to the backhaul. In addition, it allows third parties to deploy their applications and services for the mobile users.

5G RAN Manager is the radio side of the 5G network, which connects the user terminals to the core network via the EPON ONU and SDN-OLT. We propose that some functionalities of the RAN (such as support for various traffic types (e.g. IoT), QoS guarantees and latency optimization for the end users, to name a few possibilities) are migrated and performed in the SDN controller and managed via the OpenFlow switch. In addition, the proposed framework must be capable to align the DBA power saving mechanism of the ONU with the RAN scheduler in order to perform the adequate power saving. For example, when traffic load is low and is not affected by latency guarantees (e.g. data from IoT sensors) and can thus be delayed and “packed” into bursts, it would be feasible to switch off the Remote Radio Units (RRUs) during the inactivity periods between bursts. This solution could minimize energy consumption across the wide range of the network, while maintaining the penalty introduced in terms of channel utilization and packet delay at a suitable level.

This framework allows integrating the EPON and RAN power saving techniques. In addition, it enables network provider to minimize the expenses associated with capital expenditure by deploying new services over different slice for several tenants. Regarding this topic, we recently published a paper in the ICTON 2018 conference [86].

5.2.2 P4-based SDN architecture for PON networks

Another area to explore is to extend the OpenFlow-based SDN-EPON architecture to the P4[9] architecture [87]. This can be done by replacement of the OpenFlow-based SDN-OLT with a P4 programmable switch. Similarly to the SDN-EPON architecture, the functionalities of the MPMC (Multi-Point MAC Control) sub-layer of the OLT are virtualized and split into two parts: one is migrated to the P4ON controller, which runs on the ONOS controller while the other part remains at the P4ON-OLT and integrated with the P4-enabled switch. The main functional block of the MAC client remains in the P4ON-OLT, to be managed by the tables of the P4-based switch.

This design can enhanced with P4-runtime working as the southbound interface between the controller (ONOS) and the OLT (P4 switch), and also as a control API. P4 runtime allows the controller to provide media access control (MAC) and software-defined functionalities to the PON system, by matching and acting on specific tables. The ONOS controller supports P4 and P4 runtime, and the possibility to introduce slicing capabilities. This approach can open the way for the integration of network slicing with Cloud RAN, SDN and NFV technologies to dynamically create programmable virtual networks and offer differentiated traffic and services.

P4 allows us to push some functionalities of the network into the programmable data plane (P4-OLT). This can greatly reduce cost and improve performance while improving the service agility and scalability in the network. Describing the forwarding behavior of the data plane with the P4 language makes it possible for a packet-accurate, executable model of an entire network to be built prior to deployment. Regarding this topic, one of our paper has been accepted to URSI 2018 conference [88].

5.2.3 Other topics to be explored

Apart from the two aforementioned topics, other paths worth to be explored are:

- The successful implementation of SDN-EPON architecture in simulation environment leads us to focus and implement it in the enterprise (a real EPON environment). This can help to evaluate the SDN-EPON architecture and its real effect in the network by virtualizing some of functionalities of the OLT and

[9] P4: programming protocol-independent packet processors.

migrate them to the real SDN-based EPON controller. The virtualization of a real OLT device can reduce the capital expenditure of the network and improve the operational functionalities of the same network.

- An analysis of future PON technologies is needed to evaluate if we can extend the study of contributions related to the SDN-EPON architecture to other PON technologies such as 10G-EPON and 100G-EPON networks [89]. For this, we need to improve the exiting SDN management layer of our architecture in order to consider the management of wavelength allocation [90] and the coexistence of 1G/10G/100G network.
- Another topic of interest is the evaluation and potential improvement of the SDN-EPON architecture working with DBAs other than IPACT, for example a distributed algorithm such as DDSPON. This can help us to improve bandwidth management, performance, user experience, throughput and QoS, and reduce the end-to-end delay [91] and queue size in the network.
- Finally, it would be interesting to create an analytical model of the system to optimize energy saving and quality-of-service in the next generation PONs-based on SDN paradigm [92]. This can help to extend our contributions related to the SDN-EPON architecture to be combined with the energy saving technologies and provision of QoS.

6 REFERENCES

[1] H. Khalili, D. Rincón and S. Sallent, "Towards an Integrated SDN-NFV Architecture for EPON Networks," *Advances in Communication Networking, Springer,* vol. 8846, pp. 74-84, December 2014, Lecture Notes in Computer Science.

[2] H. Khalili, S. Sallent, J. R. Piney and D. Rincon, "A proposal for an SDN-based SIEPON architecture," *Optics Communications,* vol. 403, pp. 9-21, November, 2017.

[3] H. Khalili, D. Rincon, S. Sallent and J. R. Piney, "An integrated SDN-based architecture for Passive Optical Networks," *Chapter accepted for publication in Broadband Communications Networks - Recent Advances and Lessons from Practice, Eds A. Haidine and A. Aqqal, Intech,* 2018.

[4] H. Khalili, S. Sallent, D. Rincón and J. R. Piney, "An Energy Efficient Dynamic Bandwidth Allocation Algorithm for Passive Optical Networks," *Submitted to IEEE Access Journal,* 2018.

[5] OPNET Technologies - Network Simulator | Riverbed, [Online]. Available: https://www.riverbed.com/es/products/steelcentral/opnet.html?redirect=opnet.

[6] ITU-T G.984.1 SG 15. Gigabit-Capable Passive Optical Networks (G-PON): General Characteristics, ITU-T standard, March, 2003.

[7] ITU-T G.987, 10-Gigabit-capable passive optical network (XG-PON) systems: Definitions, abbreviations and acronyms, ITU-T standard, Telecommunication Standardization sector of ITU, Jun, 2012.

[8] IEEE 802.3ah working group, IEEE Standard for Information Technology. Ethernet in the First Mile. IEEE Std 802.3ak-2004 (Amendment to IEEE Std 802.3-2002 as amended by IEEE Stds 802.3ae-2002, 802.3af-2003 and 802.3aj-2003), IEEE, New York, Sep, 2004.

[9] K. Tanaka, A. Agata and Y. Horiuchi, "IEEE 802.3 av 10G-EPON standardization and its research and development status," *Journal of Lightwave Technology, IEEE*, vol. 28, pp. 651-661, 2010.

[10] G. Kramer, "What is Next for Ethernet PON," *The Joint International Conference on Optical Internet and Next Generation Network. COIN-NGNCON*, pp. 49-54, Jeju, South Korea, 2006.

[11] G. Kramer, "Ethernet Passive Optical Networks," McGraw-Hill, New York, 2005.

[12] L. Chang-Hee, S. Wayne V. and K. Byoung Yoon, "Fiber to the Home using PON infrastructure," *Journal of Lightwave Technology*, vol. 24, no. 12, pp. 4568-4583, 2006.

[13] M. P. McGarry, M. Maier and M. Reisslein, "Ethernet PONs: a Survey of Dynamic Bandwidth Allocation (DBA) Algorithms," *IEEE Communications Magazine*, vol. 42, no. 8, pp. 8-15, 2004.

[14] M. P. McGarry, M. Reisslein and M. Maier, "Ethernet Passive Optical Network Architectures and Dynamic Bandwidth Allocation Algorithms," *IEEE Communication Surveys and Tutorials*, vol. 10, no. 3, pp. 46-60, 2008.

[15] B. Skubic, J. Chen, J. Ahmed, B. Chen and L. Wosinska, "Dynamic bandwidth allocation in EPON and GPON," *Chapter in Convergence of Mobile and Stationary Next-Generation Networks*, pp. 227-252, 2010.

[16] H. Naser and H. T. Mouftah, "Survey of Dynamic Bandwidth Allocation for Ethernet Passive Optical Networks," *Optical Switching and Networking*, vol. 6, no. 3, pp. 151-162, 2009.

[17] F. Ziyi, J. Bai and Q. Wang, "A Novel Dynamic Bandwidth Allocation Algorithm with Correction-based the Multiple Traffic Prediction in EPON," *Journal of Networks*, vol. 10, pp. 1554-1560, 2012.

[18] C. Li, W. Guo, W. Hu and M. Xia, "Energy-efficient dynamic bandwidth allocation for EPON networks with sleep mode ONUs," *Optical Switching and Networking*, vol. 15, pp. 121-133, 2015.

[19] G. Kramer, B. Mukherjee and G. Pesavento, “Interleaved Polling with Adaptive Cycle Time (IPACT): A Dynamic Bandwidth Distribution Scheme in an Optical Access Network,” *Photonic Network Communications,* vol. 4, no. 1, pp. 89-107, 2002.

[20] S. Chuan-Ching, . W.-N. Sung and H.-Y. Chen, “Active Scheduling Algorithm for Intra-ONU Dynamic Bandwidth Allocation in Ethernet Passive Optical Networks,” *IEEE, First International Conference on Ubiquitous and Future Networks ,* pp. 240-245, 2009.

[21] H.-J. Byun, J.-M. Nho and J.-T. Lim, “Dynamic Bandwidth Allocation Algorithm in Ethernet Passive Optical Networks,” *IET, Electronics Letters,* vol. 39, no. 13, pp. 1001-1002, 2003.

[22] M. De Andrade, L. Gutierrez and S. Sallent, “DDSPON: A Distributed Dynamic Scheduling for EPON,” *IEEE International Conference on Signal Processing and Communication ICSPC,* pp. 840-843, 2007.

[23] A. R. Dhaini, C. M. Assi, A. Shami and N. Ghani, “Adaptive Fairness through intra-ONU Scheduling for Ethernet Passive Optical Networks,” *IEEE International Conference on Communications, 2006. ICC '06. ,* p. 2687–2692, June, 2006.

[24] S. Sadon, N. Radzi, N. Din and M. Al-Mansoori, “Efficient decentralized dynamic bandwidth allocation in EPON,” *Proceedings of the World Congress on Engineering and Computer Science,* vol. 2, pp. 23-25, 2013.

[25] A. Dhaini and C. Assi, “Quality of Service in Ethernet Passive Optical Networks (EPONs),” in *Broadband Access Networks,* Springer, 2009, pp. 169-196.

[26] G. Kramer, B. Mukherjee, S. Biswanath, Y. Ye and R. Hirth, “Supporting Differentiated Classes of Service in Ethernet Passive Optical Networks,” *Journal of Optical Networking,* vol. 1, no. 8, pp. 280-298, 2002.

[27] N. Ghami, A. Shami, C. Assi and M. Y. A. Raja, “Quality of Service in Ethernet passive Optical Networks,” in *Chapter in Advances in Wired and Wireless Communication,* IEEE, 2004, pp. 161-165.

[28] S.-i. Choi and J. Park, "SLA-aware dynamic bandwidth allocation for QoS in EPONs," *Journal of Optical Communications and Networking,* vol. 2, no. 9, pp. 773-781, 2010.

[29] A. Shami, X. Bai, C. M. Assi and N. Ghani, "Jitter performance in Ethernet passive optical networks," *Journal of Lightwave Technology, IEEE,* vol. 23, no. 4, pp. 1745-1753, 2005.

[30] J. R. Ferguson, M. Reisslein and M. P. McGarry, "Online excess bandwidth distribution for Ethernet passive optical networks," *Journal of Optical Networking,* vol. 8, no. 4, pp. 358-369, 2009.

[31] C. M. Assi, Y. Ye, S. Dixit and M. A. Ali, "Dynamic bandwidth allocation for quality-of-service over Ethernet PONs," *IEEE Journal on Selected Areas in Communications,* vol. 21, no. 9, pp. 1467-1477, 2003.

[32] Y. Yan, S.-W. Wong, L. Valcarenghi, S.-H. Yen, D. R. Campelo, S. Yamashita, L. Kazovsky and L. Dittmann , "Energy management mechanism for Ethernet passive optical networks (EPONs)," *IEEE International Conference on Communications (ICC),* pp. 1-5, Cape Town, South Africa, 2010.

[33] E. Goma, M. Canini, A. Lopez Toledo, N. Laoutaris, D. Kostic, P. Rodriguez, R. Stanojevic and P. Yagüe Valentín, "Insomnia in the Access or How to Curb Access Network Related Energy Consumption," *ACM SIGCOMM Computer Communication Review,* vol. 41, no. 4, pp. 338-349, 2011.

[34] K. Langendoen and G. Halkes, "Energy-Efficient Medium Access Control," in *Embedded Systems Handbook*, Netherlands, CRC press, 2005, pp. 34.1-34.29.

[35] T. Kim, I. H. Kim, Y. Sun and Z. Jin, "Physical layer and medium access control design in energy efficient sensor networks: An overview," *IEEE Transactions on Industrial Informatics,* vol. 11, no. 1, pp. 2-15, 2015.

[36] S. Satoshi, K. Jun-ichi, Y. Naoto and H. Hisaya, "Novel sleep control for EPON optical line terminal employing layer-2 switch functions," *IEEE, Global Telecommunications Conference (GLOBECOM),* pp. 1-5, Miami, FL, USA, 2010.

[37] L. Shi, S.-S. Lee and B. Mukherjee, “Energy-Efficient Long-Reach Passive Optical Network: A Dynamic Wavelength Allocation Scheme,” *Photonics in Switching, Optical Society of America (OSA),* p. PTuB1, 2010.

[38] J. Zhang, T. Wang and N. Ansari, “Designing energy-efficient optical line terminal for TDM passive optical networks,” *34th IEEE, Sarnoff Symposium,* pp. 1-5, Princeton, NJ, USA, 2011.

[39] ITU-T, G.Sup45: GPON power conservation, ITU-T standard, 2009.

[40] A. R. Dhaini, P. H. Ho and G. Shen, “Toward Green Next-Generation Passive Optical Network,” *IEEE Communications Magazine,* vol. 49, no. 11, pp. 94-101, 2011.

[41] Y. Luo, M. Sui and F. Effenberger, “Energy-efficient next generation passive optical network supported access networking,” *Optical Switching and Networking, Elsevier,* vol. 14, pp. 43-52, 2014.

[42] Y. Zhou, R. Gu and Y. Ji, “An energy-efficient mechanism of passive optical access network based on Ethernet,” *12th International Conference on IEEE, Optical Communications and Networks (ICOCN),* pp. 1-4, Chengdu, China, 2013.

[43] J. Zhang, M. Taheri Hosseinabadi and N. Ansari, “Standards-compliant EPON sleep control for energy efficiency: Design and analysis,” *Journal of Optical Communications and Networking,* vol. 5, no. 7, pp. 677-685, 2013.

[44] C. Li, W. Guo, W. Hu and M. Xia, “Energy-efficient dynamic bandwidth allocation for EPON networks with sleep mode ONUs,” *Optical Switching and Networking,* vol. 15, pp. 121-133, 2015.

[45] A. Dixit, B. Lannoo, D. Colle, M. Pickavet and P. Demeester, “Energy efficient dynamic bandwidth allocation for Ethernet passive optical networks: Overview, challenges, and solutions,” *Optical Switching and Networking, (Elsevier),* vol. 18, pp. 169-179, 2015.

[46] D. A. Khotimsky, D. Zhang, L. Yuan, R. O. Hirafuji and D. R. Campelo, “Unifying sleep and doze modes for energy-efficient PON systems,” *IEEE Communications Letters,* vol. 18, no. 4, pp. 688-691, 2014.

[47] M. T. Hosseinabadi and N. Ansari, "Multi-power-level energy saving management for passive optical networks," *OSA, Journal of Optical Communications and Networking,* vol. 6, no. 11, pp. 965-973, 2014.

[48] M. P. I. Dias and E. Wong, "Sleep/doze controlled dynamic bandwidth allocation algorithms for energy-efficient passive optical networks," *Optics express, Optical Society of America,* vol. 21, no. 8, pp. 9931-9946, 2013.

[49] G. Kramer, L. Khermosh, F. Daido, A. Brown, H. Yoon, K. I. Suzuki and W. Bo, "The IEEE 1904.1 standard: SIEPON architecture and model," *IEEE Communications Magazine,* vol. 50, no. 9, pp. 98-108, 2012.

[50] "IEEE standards association, IEEE 1904.1 Standard for Service Interoperability in Ethernet Passive Optical Networks," IEEE, New York, 2013.

[51] I. Cale, A. Salihovic and M. Ivekovic, "Gigabit passive optical-GPON," *29th International Conference on Information Technology Interface,* pp. 679-684, Cavtat, Croatia, 2007.

[52] K. I. Suzuki, "G.EPON and Current Status of Related Standardization," Global Snatdardization Activities, NTT, 2014.

[53] I. S. wang, A. Nikoukar, A. F. Pakpahan and A. T. Liem, "ONU-Initiated Transceiver Mode Energy-saving Mechanism in SIEPON," *Proceedings of the International MultiConference of Engineers and Computer Scientists,* vol. 2, pp. 499-502, Hong Kong, 2016.

[54] A. Nikoukar, I. S. Hwang, C.-J. Wang, M. S. Ab-Rahman and A. T. Liem, "A SIEPON based transmitter sleep mode energy-efficient mechanism in EPON," *Optical Fiber Technology,* vol. 23, pp. 78-89, 2015.

[55] B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials,* vol. 16, no. 3, pp. 1617-1634, 2014.

[56] Open Networking Foundation, Software Defined Networking: The New Norm For Networks, [Online]. Available:

https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf.

[57] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review,* vol. 38, no. 2, p. 69–74, 2008.

[58] B. Hedlund, "Network Diary: On data center scale, OpenFlow, and SDN," [Online]. Available: http://onurserter.blogspot.com.es/2011/11/on-data-center-scale-openflow-and-sdn.html.

[59] W. Stallings, Software-Defined Networks and Openflow, [Online]. Available: https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-59/161-sdn.html.

[60] Open Network Fundation, OpenFlow Switch Specification version 1.5.0, 2014. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf.

[61] A. Mercian, M. P. McGarry and M. Reisslein, "Software Defined Optical Access Networks (SDOANs): A Comprehensive Survey," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 4, pp. 2738-2786, 2016.

[62] S. Das, G. Parulkar and N. McKeown, "Simple unified control for packet and circuit networks," *in Proc. IEEE/Lasers and Electro-Optics Society (LEOSST) Meet,* pp. 147-148, Newport Beach, CA, USA, 2009.

[63] S. Das, G. Parulkar and N. McKeown, "Unifying packet and circuit switched networks," *IEEE Global Communications Conference (GLOBECOM) Workshops,* pp. 1-6, 2009.

[64] L. Liu, T. Tsuritani, I. Morita, H. Guo and J. Wu, "OpenFlow-based wavelength path control in transparent optical networks: A proof-of concept demonstration," *37th European Conference and Exposition on Optical Communications, ECOC,* pp. 1-3, Geneva, Switzerland, 2011.

[65] L. Liu, D. Zhang, T. Tsuritani, R. Vilalta, R. Casellas, L. Hong, I. Morita, H. Guo and J. Wu, "First field trial of an OpenFlow-based unified control plane for multi-layer multi-granularity optical networks," *Optical Fiber Communication Conference,* pp. PDP5D-2, Los Angeles, 2012.

[66] F. N. Farias, J. J. Salvatti, E. C. Cerqueira and A. J. Abelem, "A proposal management of the legacy network environment using OpenFlow control plane," *Network Operations and Management Symposium (NOMS), IEEE,* pp. 1143-1150, Maui, HI, USA, 2012.

[67] N. Cvijetic, "Software-defined optical access networks for multiple broadband access solutions," *In OptoElectronics and Comm. Conf (OECC),* pp. 1-2, Kyoto, Japan, 2013.

[68] P. Parol and M. Pawlowski, "Towards networks of the future: SDN paradigm introduction to PON networking for business applications," *Federated Conference on Computer Science and Information Systems (FedCSIS),* pp. 829-836, Krakow, Poland, 2013.

[69] A. Sgambelluri, F. Paolucci, F. Cugini, L. Valcarenghi and P. Castoldi, "Generalized SDN control for access/metro/core integration in the framework of the interface to the Routing System (I2RS)," *IEEE Globecom Workshops (GC Wkshps),* pp. 1216-1220, Atlanta, GA, USA, 2013.

[70] R. Gu, Y. Ji, P. Wei and S. Zhang, "Software defined flexible and efficient passive optical Networks for intra-datacenter communications," *Optical Switching and Networking, Elsevier,* vol. 14, no. 3, pp. 289-302, 2014.

[71] M. Ruffini, F. Slyne, C. Bluemm, N. Kitsuwan and S. McGettrick, "Software defined networking for next generation converged metro-access networks," *Optical Fiber Technology, Elsevier,* vol. 26, pp. 31-41, 2015.

[72] H. Yang, J. Zhang, Y. Zhao, J. Wu, Y. Ji, Y. Lin, J. Han and Y. Lee, "Experimental demonstration of remote unified control for OpenFlow-based software-defined optical access networks," *Photonic Network Communications, Springer,* vol. 31, no. 3, pp. 568-577, 2016.

[73] A. Amokrane, J. Hwang, J. Xiao and N. Anerousis, “Software defined enterprise passive optical network,” *In 10th International Conference on Network and Service Management (CNSM) and Workshop, IEEE,* pp. 406-411, Rio de Janeiro, Brazil, 2014.

[74] C. Li, W. Guo, W. Wang, W. Hu and M. Xia, “Programmable bandwidth management in software-defined EPON architecture,” *Optics Communications,* vol. 370, pp. 43-48, 2016.

[75] L. Peterson, A. Al-Shabibi, T. Anshutz, S. Baker, A. Bavier, S. Das, J. Hart, G. Palukar and W. Snow, “Central office re-architected as a data center,” *IEEE Communications magazine,* vol. 54, no. 10, pp. 96-101, 2016.

[76] “Open Network Operating System (ONOS),” [Online]. Available: http://onosproject.org/.

[77] T. Kusano, “Architecture for an Access Network System Management Protocol Control under Heterogeneous Network Management”. US Application Patent 14/256,011, 2014, http://www.patentsencyclopedia.com/app/20140314087.

[78] “ONF Demonstrates CORD-Powered Open Innovation Pipeline Solution at ONS,” [Online]. Available: https://opencord.org/tag/ons/.

[79] “AT&T Open GPON, NFV OLT Line Card Specification, revision 1.2,” [Online]. Available: https://www.scribd.com/document/343393490/ATT-vOLT-v1-2-1-pdf.

[80] A. Al-Shabibi and H. Jonathan, “Virtual OLT (vOLT),” CORD, Central Office Re-architectured as a Datacenter, 12 February 2016. [Online]. Available: http://opencord.org/wp-content/uploads/2016/03/Virtual-OLT.pdf.

[81] D. Rincón, “Contributions to the Wavelet-based Characterization of Network Traffic,” PhD Thesis, Universitat Politecnica de Catalunya - Barcelona, July 2007.

[82] D. Rincón and S. Sallent, “Scaling Properties of Network Traffic, chapter in Encyclopedia of Internet Technologies and Applications, Eds M. Freire and M. Pereira,” *IGI Global,* pp. 483-489, October 2005.

[83] L. Valcarenghi, D. P. Van, P. G. Raponi, P. Castoldi, D. R. Campelo, S.-W. Wong, S.-H. Yen, L. G. Kazovsky and S. Yamashita, "Energy efficiency in passive optical networks: where, when, and how?," *IEEE Network,* vol. 26, no. 6, pp. 61-68, 2012.

[84] M. De Andrade, L. Gutierrez and S. Sallent, "A Distributed Scheduling Approach for Ethernet-based Passive Optical Networks," in *32nd IEEE Conference on Local Computer Networks, LCN 2007,* Dublin, Ireland, 29 October 2007.

[85] E. Wong, M. Mueller, M. P. Dias, C. A. Chan and M. C. Amann, "Energy-efficiency of optical network units with vertical-cavity surface-emitting lasers," *Optics express,* vol. 20, no. 14, pp. 14960-14970, 2012.

[86] H. Khalili, P. Sayyad Khodashenas, D. Rincon, S. Siddiqui, J. R. Piney and S. Sallent, "Design Considerations for an Energy-Aware SDN-based Architecture in 5G EPON Nodes," *Accepted for publication in the 20th Anniversary International Conference on Transparent Optical Networks (ICTON 2018),* July, 2018.

[87] P. Bosshart, D. Dali, G. Gibb and et. al, "P4: programming protocol-independent packet processors," *ACM SIGCOMM Computer Communication Review,* vol. 44, no. 3, pp. 87-95, 2014.

[88] A. Haastrup, D. Rincón, H. Khalili and S. Sallent, "P4ON: A 5G-Aware P4-Based PON Architecture," *Accepted for publication in the XXXIII National Symposium of the International Scientific Union of Radio (URSI 2018),* September, 2018.

[89] C. Knittle, "IEEE 100G-EPON," *Optical Fiber Communication Conference, Optical Society of America,* pp. Th1I-6, 2016.

[90] F. Wang, B. Liu, L. Zhang and et. al, "Dynamic bandwidth allocation based on multiservice in software-defined wavelength-division multiplexing time-division multiplexing passive optical network," *Optical Engineering, International Society for Optics and Photonics,* vol. 56, no. 3, p. 036104, 2017.

[91] K. Kondepu, A. Sgambelluri, L. Valcarenghi, F. Cugini and P. Castoldi, "Exploiting SDN for integrating green TWDM-PONS and metro networks preserving end-to-end delay," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 1, pp. 67-74, 2017.

[92] B. Yan, J. Zhou, J. Wu and Y. Zhao, "SDN based energy management system for optical access network," *IEEE International Conference on Communications and Networking in China (CHINACOM), 2014 9th,* pp. 658-659, 2014.

APPENDIX A

In this thesis, we designed a novel architecture for EPON networks based on the SDN concept in order to model the EPON standard issues and limitations. As there are no available SDN-EPON implementation, we build a simulation network model with the OPNET Modeler tool to emulate the SDN and OpenFlow protocol functionalities. Hence, an integrated network model has been developed for the EPON networks; including an SDN-EPON controller (as an intelligent part), SDN-OLT (as a main element for handling and forwarding the packets), ONUs, and splitter/combiners. Figure A- 1 shows the SDN-EPON network model.

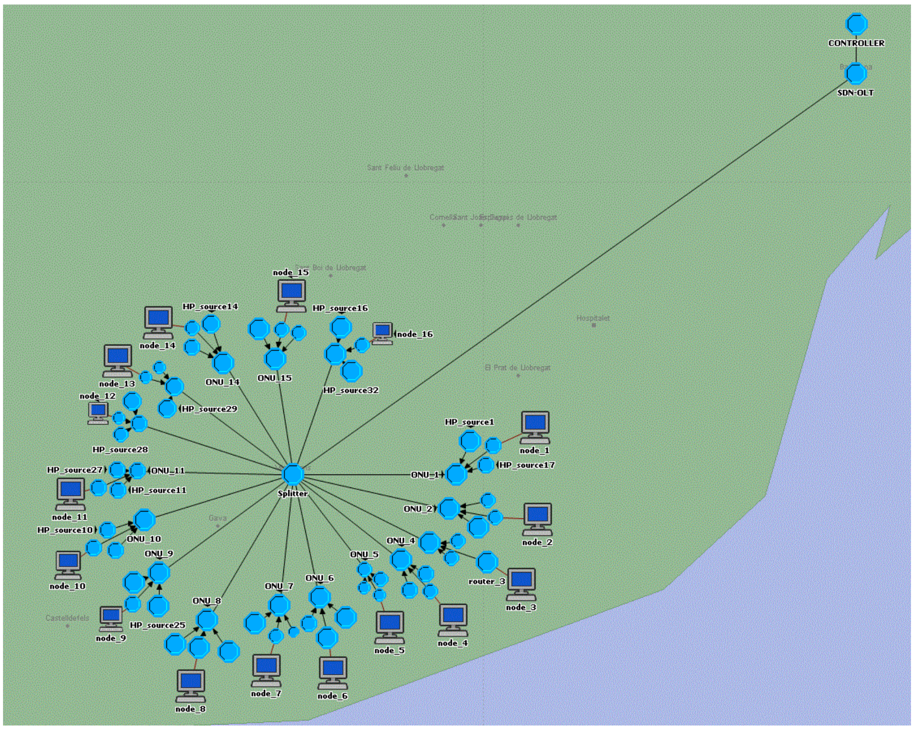


Figure A- 1 SDN-EPON network model.

A.1 SDN-EPON Controller Node Model

This section describe the processes that comprise the SDN-EPON controller node model, shown in Figure A- 2.

A.1.1 FlowMode generation

The FlowMod process is responsible to define FlowMod messages and rules for the normal GATE, REPORT, DISCOVERY GATE, REGISTER_REQ, REGISTER, and REGISTER_ACK messages. When a control message or a data message is received via a Packet-In in the SDN-EPON controller from the SDN-OLT, the FlowMod process collects and analyzes the data of the incoming message, and generates a rule via FlowMode message or Packet-Out with the set of actions, and then sends it as a new rule to the SDN-OLT node.

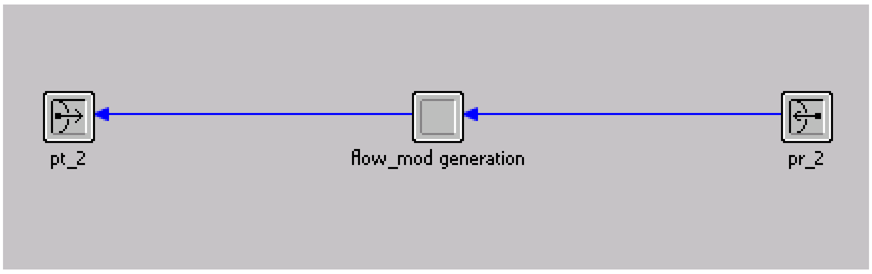


Figure A- 2 SDN-EPON controller node model.

A.2 SDN-OLT Node Model

This section describe the processes that comprise the novel architecture of SDN-OLT node model, shown in Figure A- 3.

A.2.1 OpenFlow Switch

This entity is the main element of the SDN-OLT, and it performs the three main functionalities of the legacy OLT (namely CP, CM and MPTC). It also emulates and executes OpenFlow switch functionalities (such as flow match and execute actions). It has a global view of all the MPCP instances and is responsible for allowing only one instance to transmit at specific time in order to avoid collision and loss of the packet.

It is also responsible for analyzing the received packets - it distinguishes control and data messages when the packets are received from the ONUs. Moreover, the OF-switch is in charge of calculating the RTT of incoming packets by interacting with the DBA agent through the defined packet registers. With the arrival of each message, the RTT of control messages is calculated and compared with the previous RTT. In addition, the OF-switch is responsible for prioritizing and matching the packets and execute the actions. If the transmitted message is a control message, this module is responsible for stamping the current time of the SDN-OLT in the corresponding field.

A.2.2 DBA Agent

This entity keeps the algorithm of the SDN-OLT. The DBA agent is responsible for calculating new GATE message parameters in order to grant start time and the transmission (slot allocation) length for the ONU. In addition, it calculates the RTT of the received REPORT message for keeping synchronization between the SDN-OLT and the ONU.

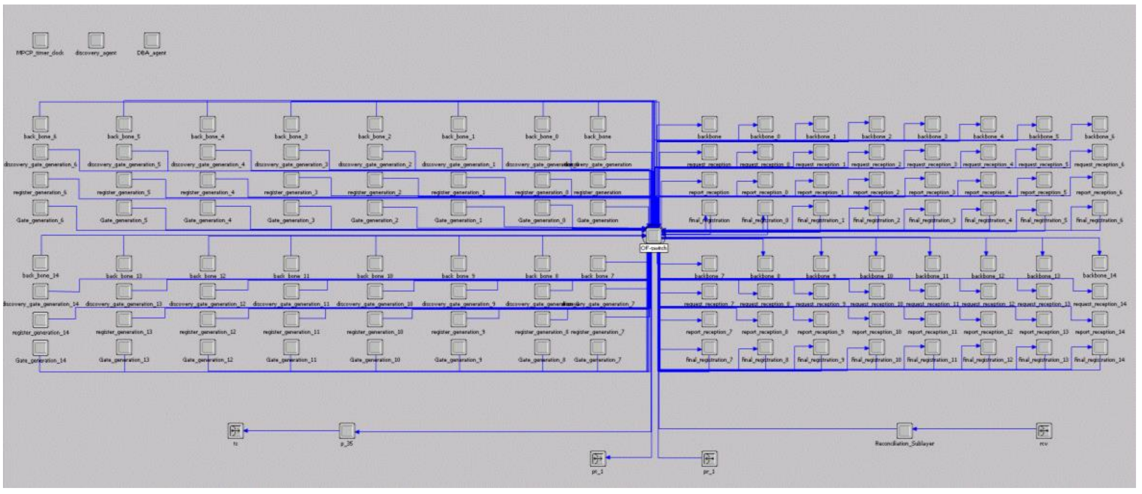


Figure A- 3 SDN-OLT node model.

A.2.3 Discovery Agent

The discovery agent decides when to initiate a discovery round and allocates a discovery window when no previously initialized ONUs are allowed to transmit. The DBA agent ensures that no active ONUs are scheduled to transmit during the discovery window. The discovery agent instructs the discovery process to send a special GATE message, called discovery GATE message, advertising the start time of the discovery slot and its length.

A.2.4 MPCP Timer Clock

MPCP clock is used to maintain a local clock of the SDN-OLT. The MPCP clock is a 32-bit counter that counts time in units of time quanta (TQ). The TQ is defined to be a 16-ns interval, or the time required to transmit 2 bytes of data at 1 Gbps line rate.

Correspondingly, the timeslot start times and lengths in GATE messages, as well as queue lengths in REPORT messages, are expressed in TQ units.

A.2.5 Gate Generation

This entity is responsible for generating GATE messages to grant transmission windows (by indicating the start time and the length) for each ONU. This operation is executed through dialogue with the DBA agent. Upon receiving a request from DBA, the gate generation process generates a GATE message for a specific ONU and forwards the message to the OF-switch for transmitting it to the corresponding ONU. The number of grants can also be set to zero for using the GATE messages as an MPCP keep-alive from the SDN-OLT to the ONU.

A.2.6 Discovery Gate Generation

This entity manages the discovery process, through which an ONU is discovered and registered with the network while compensating for the RTT delay. It periodically generates a DISCOVERY GATE control message indicating the start time and duration of the window discovery along with the Discovery Information flag set. Through this, the SDN-OLT notifies the all ONUs about its downstream and upstream channel transmission capabilities. Based on that, SDN-OLT has enough information to schedule the ONU for access to the PON and transmit the standard GATE message. The discovery gate process also informs to the Register Reception process about the start time and end of the discovery window to receive only registration request messages within the reserved time.

A.2.7 Register Generation

The register generation entity is responsible for generating a REGISTER message that may indicate the ONU registers or deregisters. The register process is driven by the discovery agent, in which it waits until the discovery agent issues a request to transmit a REGISTER message, and then the register generation process builds a REGISTER message and transmits it to the ONU.

A.2.8 Report Reception

This module is responsible for receiving the REPORT messages and passing the received data to the DBA agent in order to plan the next transmission window. A timer is embedded in the report reception process that activated by receiving REPORT message. The timer is used to measure the interval between the arrival of the REPORT message and it set to 1 second, so if no REPORT messages received, the timer will be expired and the ONU will be deregistered from the associated instance.

A.2.9 Request Reception

The request reception process is responsible for receiving the REGISTER_REQ messages and passing the received data to the DBA agent. This process only runs on the instance of MPCP associated with the broadcast LLID. The request reception process only accepts REGISTER_REQ messages that arrive within the discovery window. All messages that arrive outside the discovery window are discarded without informing the discovery agent.

A.2.1 Final Registration

The final registration process at the OLT is responsible for issuing a unicast GATE message to an ONU to grant transmission time for sending a registration confirmation message and receiving the REGISTER_ACK message by the ONU. Upon receipt of the REGISTER_ACK, the discovery process for that ONU is complete, the ONU is registered and normal message traffic can begin.

A.2.2 Backbone

This entity is defined as an upper layer (MAC client), which it is used to communicate with the OF-switch entity for receiving and forwarding data frames from and to the MPMC sublayer, respectively.

A.2.3 Reconciliation Sublayer

This is an interconnection between MAC sublayer and the PHY sublayers. The reconciliation sublayer enables multiple data link layers to interface with a single physical layer.