

A proposal for a SDN-based SIEPON architecture

Hamzeh Khalili^a, Sebastià Sallent^a, Jose Ramon Piney^a
, David Rincón^a

^a*Department of Network Engineering, Universitat Politècnica de Catalunya (UPC),
Castelldefels, 08860 Barcelona, Spain*

Abstract

Passive Optical Network (PON) elements such as Optical Line Terminal (OLT) and Optical Network Units (ONUs) are currently managed by legacy, non-flexible network management systems. Software-Defined Networking (SDN) is a new networking paradigm that improves the operation and management of networks. In this paper, we propose a novel architecture, based on the SDN concept, for Ethernet Passive Optical Networks (EPON) that includes the Service Interoperability (SIEPON) standard. In our proposal 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 (OF) switch. A new MultiPoint MAC Control sublayer extension based on the OpenFlow protocol is presented. The controller is designed to effectively manage and enhance the resource utilization, flow monitoring, bandwidth assignment, quality-of-service (QoS) guarantees, and energy management of the optical network access. The OpenFlow switch is extended with synchronous ports to keep 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). Our simulation-based results demonstrate the effectiveness of the new architecture, while keeping similar or improved performance compared to legacy PONs, in terms of delay and throughput.

Key words: Passive Optical Network (PON), Ethernet Passive Optical Network (EPON), Service Interoperability in EPON (SIEPON), Software Defined Networking (SDN), OpenFlow (OF).

Email address: khalili.hamzeh@entel.upc.edu (Hamzeh Khalili).

1 Introduction

1.1 EPON networks

Ethernet Passive Optical Networks (EPON) [1] are used in broadband access. They follow a tree topology with a central office located at the root node (Optical Line Terminal, OLT), a set of passive optical splitters, and the users connected to the leaf nodes of the tree (Optical Network Units, ONU). This architecture uses time-division multiplexing (TDM) as medium access control (MAC) to deliver data encapsulated in Ethernet packets (following the IEEE 802.3 standard) from the ONUs to the OLT. Due to the directional properties of the optical splitters, the OLT broadcasts data to all ONUs in the downstream channel. In the upstream channel each ONU must use a specific TDM channel to avoid collisions, since all ONUs share the transmission medium. In order to increase bandwidth utilization, the TDM channels are changed dynamically by the OLT based on the bandwidth demands of the ONUs, using dynamic bandwidth allocation (DBA) algorithms.

The Multipoint Control Protocol (MPCP) controls the point-to-multipoint fiber network in the upstream channel. MPCP provides the signaling infrastructure to coordinate the data transmission from the ONUs to the OLT. It also performs the operations of auto-discovery, registration, bandwidth assignment and ranging calculation (Round Trip Time computation) for newly added ONUs. Two types of messages are used to facilitate arbitration: GATE and REPORT. The OLT uses GATE messages to issue transmission grants, which contain the transmission start time and transmission length of the corresponding ONU, while ONUs use REPORT message to report bandwidth requirements to the OLT.

1.2 SIEPON architecture

The IEEE 1904.1 Service Interoperability in Ethernet Passive Optical Networks (SIEPON) standard [2][3] 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 [4] environment. 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 architecture model of SIEPON is separated into Line OLT (L-OLT)/Line ONU (L-ONU) having functions covered by the IEEE 802.3 standards, Client OLT (C-OLT)/Client ONU (C-ONU) having functions covered by the SIEPON

standards, and Service OLT (S-OLT)/Service ONU (S-ONU) having additional functions defined by the operators or vendors. Figure 1 shows a detailed view of the SIEPON OLT internal architectures and the respective interfaces.

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 EPON service paths (ESPs) as a conceptual path of a frame through the functional blocks of the MAC client to provide basic functional blocks of the MAC client to achieve provisioning, connection and QoS. The basic functional blocks in ESPs that provide data forwarding in OLTs and ONUs are: Input [I], Classifier [C], Modifier [M], Policer/Shaper [PS], Cross-connect [X], Queues [Q], Scheduler [S], and Output [O]. Each of these blocks may include multiple independent instances that act on different traffic flows.

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.

1.3 SDN architecture

Software Defined Networking (SDN) [5] has gained a lot of attention in recent years, because it addresses the lack of programmability in existing 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 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 programability, global optimization of network operations (load balancing), high flexibility, and the possibility of realizing a centralized network view (and this opening the way to the optimization of network operations).

The control plane communicates with the data plane through a southbound interface. OpenFlow (OF) was the original southbound API and remains one of the most common protocols. It allows a remote controller to specify the path of network packets through the network of switches, using the flow concept to manage network traffic based on predefined match rules which are statically or dynamically adjusted by the controller. OF allows the network manager to define a set of rules to match against packet headers, and also the actions to

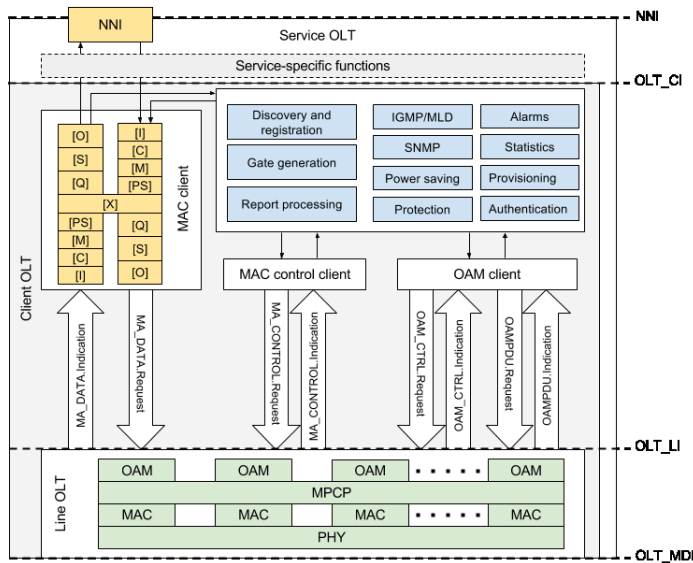


Figure 1. OLT SIEPON architecture (based on [3]).

execute on the matched packet such as forward to a certain port, modify its content or headers, or drop the packet.

1.4 State of the art of SDN applied to PONs

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. [6] 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. We now comment the main contributions related to our work.

In [7] a SDN-based architecture for PON networks is proposed with an extension of OF to provide traffic mapping and forwarding capabilities without affecting the latency of the data link layer (L2). The authors of [8] proposed a meta-MAC algorithm for applying SDN principles to the optical access and mobile backhaul networks, which allow to coordinate the medium access control algorithms of different PON technology and physical layers. [9] presents a control plane architecture for SDN-driven converged metro-access networks. The SDN controller maintains communication with the network orchestrator, and exports an abstracted network view to the controllers. In [10] a new software-defined optical access network (SDOAN) architecture is proposed. A service-aware flow scheduling strategy (SA-FS) is introduced to efficiently and flexibly allocate bandwidth resources of the network. An efficient software-

defined PON with a management scheme of Network Coding (NC) is presented in [11] to flexibly implement the coding operation on local peer traffic and keep the compatibility with current Multipoint MAC Control sublayer in a single OLT.

[12] 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. In [13], a novel EPON architecture based on the SDN and NFV (Network Functions Virtualization) paradigms is presented, where OLT and ONUs are partially virtualized and migrated to the centralized controller to reduce OPEX, CAPEX and complexity of EPON operations. [14] proposes 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.

Finally [15] is, to the best of our knowledge, the only work that presents an application of the SDN paradigm to the SIEPON standard. It describes efficient ways to allow nodes to interoperate with an existing management protocol and with other protocols, including a control and data planes interface for supporting SDN. Our contributions are aligned, and we go farther by providing a complete architectural solution, low-level implementation details, a functional validation and performance measurements.

1.5 Our proposal

Our proposal is to extend our previous work [13] to apply SDN to SIEPON in order to separate and virtualize the control and management functions of the OLT together with the OAM functions. This proposal provides faster service setup and increased flexibility of operations. Our SDN-based SIEPON proposal supports service specific functions, where the main functionalities of the EPON and SIEPON model are located and run on the SIEPON manager of the SDN-EPON controller. Our architecture opens the way to multi-tenant and multi-provider optical access networks, where different service providers share the underlying infrastructure.

An OF-switch is defined in the OLT, enabling it to behave as a SDN forwarding plane (SDN-OLT), and perform EPON service path functionalities for each single packet. The OF-switch in SDN-OLT carries out the following operations: matching of incoming packets against existing rules, modifying packet fields, limiting data transmission rates, scheduling and ensuring QoS, and forwarding packets to the destination ports. Additionally the OF-switch emulates the

functionalities of the control parser, the control multiplexer and the Multipoint Transmission Control (MPTC) modules of the OLT through the definition of synchronous ports and a set of registers. The OF-switch performs the ranging calculation of RTT (Round Trip Time) of a packet by dialoguing with the DBA, through the registers. Therefore, the actions that are performed at short time scales are still performed at the OLT, while those that are executed at longer time scales can migrate to the SDN controller.

The rest of the paper is organized as follows. Section II defines the novel SDN-based SIEPON architecture, SDN-OLT functionalities, implementation and operation details. Section III discusses the validation of the proposal, with results obtained from a reference implementation based on simulation. Section IV concludes the paper.

2 SDN-based SIEPON ARCHITECTURE

We will describe now the design and operation principles of our proposal.

2.1 Design principles

The main concept behind our approach is to build an OLT around an Open-Flow switch. This novel architecture is based on five principles:

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

2.1.1 Split the EPON functionality into virtualizable and non-virtualizable parts

From the experience gained in our previous work [13], one of the problems that has to be resolved is how to split the functionality of the Multipoint MAC Control sub-layer of the OLT that defines the MAC control operation for optical point-to-multipoint networks into two parts. One part has to be migrated to the centralized SDN-EPON controller and the other part remains at the SDN-OLT. As illustrated in Figure 2, 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.

The SDN-OLT is the main element for handling and forwarding packets. It is physically 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) to emulate the forwarding functions of the Multipoint MAC Control sublayer and functionalities and 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 which is connected with the core network. The number of synchronous logical ports associated with ONUs are variable from 1 to $64^1 + 1$ ports, where the 64 ports are unicast and dedicated to each one of 64 possible connected ONUs and the extra port is a broadcast port associated with all available ONUs. All the 64 logical ports and their corresponding 64 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 2 illustrates the SDN-OLT architecture with the OF-switch embedded in the MultiPoint MAC Control (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).

The SDN-EPON controller is the intelligent part of the system that dialogues with the SDN-OLT and can be run either in a data center 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 it is responsible for configuring the MAC control client parameters (for example the Logical Link Identifier (LLID)), MAC client policies, SIEPON control management, energy management control, QoS, linking high level QoS of SIEPON with low level MAC QoS, Path Computation Element 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

¹ The maximum number of ONUs supported in an EPON is 64.

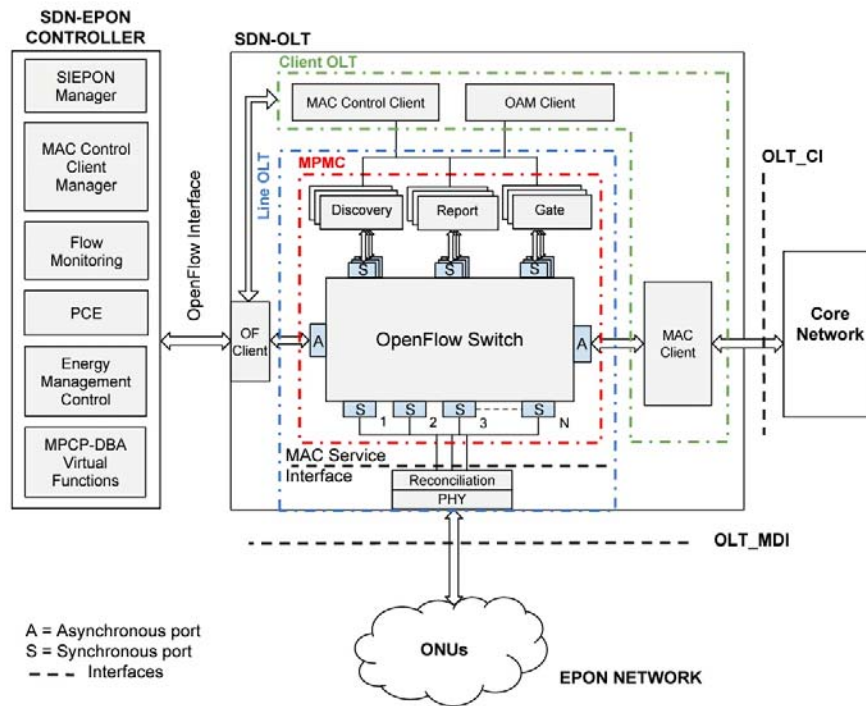


Figure 2. SDN-based SIEPON architecture.

MAC client remains in the SDN-OLT to be 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, 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 LLID, logical port status (including its speed, 1Gb/10Gb), and MAC address of the ONU in order to configure the flow tables of the switch. The virtual MPCP-DBA module is in charge of the general policy of the network and it allows the controller to dialog with the SDN-OLT in order to change the DBA parameters and policies, change the DBA priority for each ONU, or switching between multiple DBA algorithms.

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

2.1.2 Extend the 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 issued in specific times managed 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 made an extension of the Multipoint MAC Control sublayer of the OF-switch, that must keep the synchronous nature of the EPON network. Also, the forwarding of frames from the MAC control and the MAC client to the MAC service interface in the downstream direction must be synchronous.

The OF-switch synchronizes newly added ONU with the network by 1) assigning three synchronous logical ports for each connected instance of gate, report and discovery process, 2) assigning an input/output synchronous logical port with its respective LLID to link with the MAC address of the connected ONU, 3) timestamping out-going packets sent to the ONUs, 4) ranging the RTT of incoming packets from ONUs, 5) allocating bandwidth and transmission time slots to permit ONU to transmit data within a specified time slot in the upstream direction (packet forwarding strategy) with the cooperation of the SIEPON manager, and 6) 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 single outgoing packet at the logical synchronous port. The synchronization between SDN-OLT and ONUs is kept using a register and port recirculation (functionalities of OpenFlow v1.5), which interact with the DBA for calculating the RTT of incoming packets.

Also, the OF-switch synchronizes the Multipoint MAC Control instances with the DBA in order 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 allow 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 illustrates the extension of Multipoint MAC Control sublayer of the SDN-OLT.

2.1.3 Extend the 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

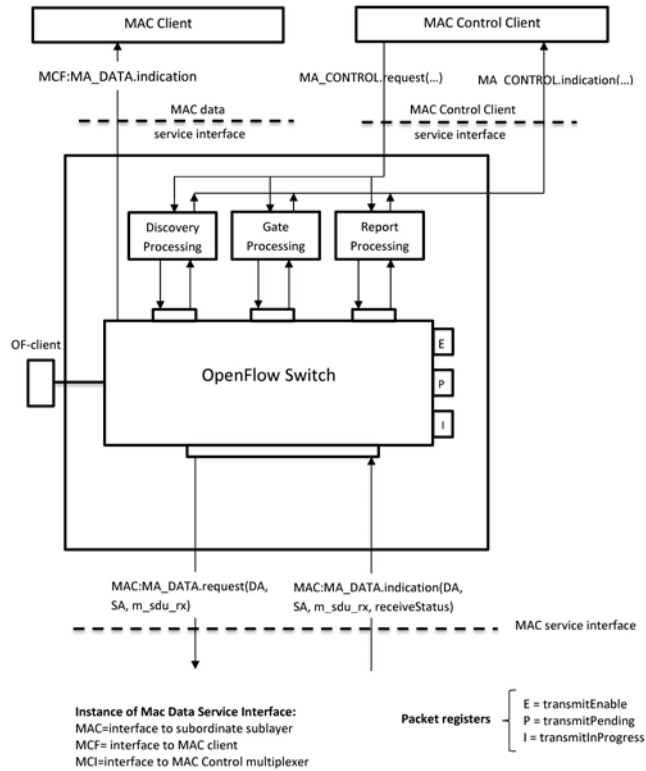


Figure 3. Multipoint MAC Control functional block, including the synchronous OF-switch and MAC interfaces.

query of controller whether the switch is a SDN-OLT switch or not. Table 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 a new extension for the OFPT_FLOW_MOD message with a new set of synchronous rules, matching fields and actions to perform the main functionalities and operations of legacy EPON and also to implement the ESP functionalities of SIEPON (explained in detail later).

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

- PDU type (45): it represents the type of incoming packet (whether it is a MAC control frame or a data frame).

```

/*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);

```

Table 1
Extension of OpenFlow 1.5 features reply structure.

```

/*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*/
};

```

Table 2
Extension of OpenFlow 1.5 match fields.

- 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 with 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.

In order to perform 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 registers² and port recirculation³. In this architecture we have implemented five packet registers for RTT calculation, writing timestamps, and the last three packet registers (P, I and E) are used to emulate the Multipoint Transmission Control. Table 3 shows the five new extended action sets of OpenFlow 1.5 (30-34). These extended action sets for EPON packets are explained as follows:

² Packet registers are used to store temporary values and informations with set-field and copy-field actions.

³ Port recirculation allows a logical port to be associated with another logical port to process packets (packets are recirculated between logical ports).

```

/*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*/
};

```

Table 3
Extension of OpenFlow 1.5 actions.

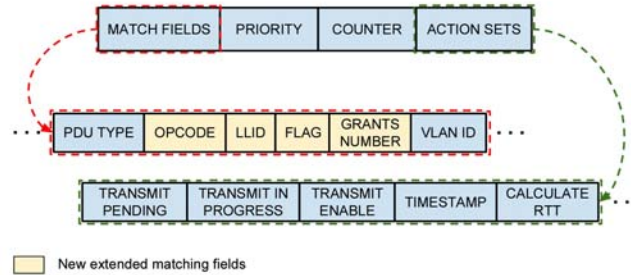


Figure 4. Example of some of the extended matching fields of the data layer and the extended action sets of the flow entry in the flow table.

- 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): 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 TransmitInProgress.
- TransmitEnable (32): 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 TransmitEnable.
- RTT calculation (33): it calculates the round trip time of incoming packets via port recirculation and the packet registers, interacting with the DBA.
- Timestamp (34): it writes the timestamp through the packet register, with the local time of the SDN-OLT once the packet transmission is enabled.

2.1.4 Implement the ESP functional block via OF messages

As we mentioned in the last subsection, our extension of the Flow-Mod message is also used to implement the ESP functional blocks of SIEPON in the SDN-OLT. Figure 4 shows an example of the OF flow entry with the extended and existing matching fields⁴. The main existing actions in our proposal which correspond to the ESP functional blocks are:

⁴ The existing match fields together with the extensions and priority are used to classify the incoming packets and determine which action must be executed.

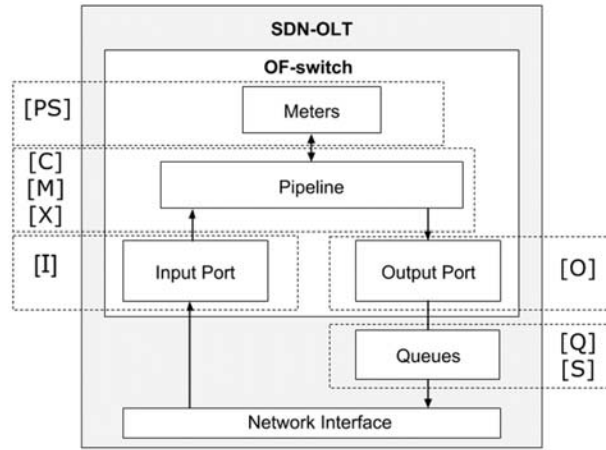


Figure 5. ESP functional block in the OF-switch.

- Set-Field: it corresponds to the *Modifier block* 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: it corresponds to the *Policer/Shaper block* and enables the measurement and control of the rate of packets (rate-limiter).
- Set-Queue: it corresponds to the *Queues block* that sets the `queue_id` for a packet when sent to the port. It is used for scheduling and forwarding the packet.
- Output port: it corresponds to the *Output block*. Packets are assigned to a physical port (controller) or a logical port (unicast port/broadcast port/an entity port of instances).

Note that in this scheme the *Scheduler block* 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. The ESP functional blocks of the OF-switch are shown in Figure 5.

2.1.5 Open the 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 controls the allocated

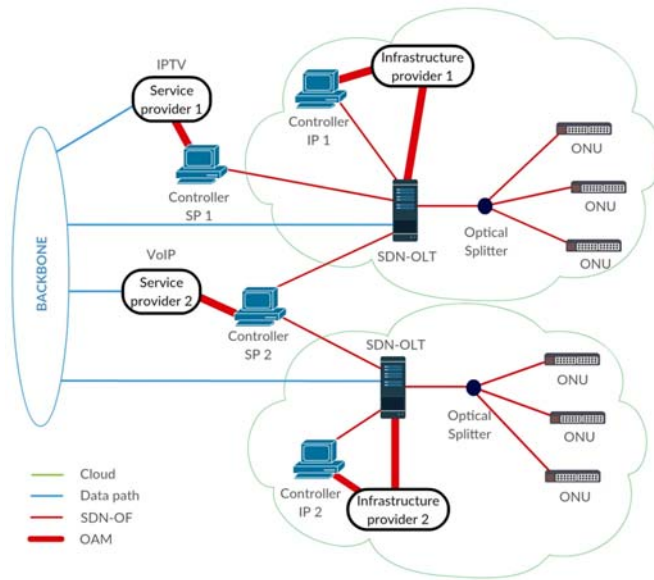


Figure 6. SDN-EPON scenario with multiple service providers.

virtual network resources. Figure 6 illustrates a scenario with multiple service and infrastructure providers with different SDN-EPON networks, which are managed and controlled by several controllers. The controllers of different service providers are able to offer various services on-demand to the users over an infrastructure that does not belong to a single infrastructure provider. For instance, service provider 1 may offer an IPTV service over the controller SP 1 while service provider 2 may offer a VoIP service over the controller SP 2.

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 end user/ONU. In order 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 aforementioned services to multiple users. So we implement those services with the meters of OF protocol to share common access with specific privileges to the software instance. OF enables meters to measure the rate of packets assigned to each flow and to apply QoS policies like rate limiting and DiffServ-like policing. Additionally, meters can assist with per port queues to schedule packets according to their priority on an output port to implement QoS. Therefore, with OpenFlow meters we can share the infrastructure among multiple tenants by defining different access and control privileges.

We now provide an example to illustrate the procedure for generating a new flow entry for a GATE message in the SDN-OLT. A GATE message indicates to a specific ONU when it must issue a frame (starting time) and transmit it, in response to a REPORT message.

When the connection is established between the SDN-EPON controller and SDN-OLT (at initial time), the controller sends the set of FlowMods with the rules to be uploaded in the flow table of the OF-switch (via the OF-client) in order to prevent the delay at the beginning of the dialogue between the SDN-OLT and the ONUs. The uploaded rules correspond to the GATE, REPORT, DISCOVERY GATE, REGISTER_REQ, REGISTER, and REGISTER_ACK messages. Later, if the controller receives a request for a GATE message from the SDN-OLT, it collects and analyzes the data of the incoming packet (source MAC address, destination MAC address, PDU type, opcode, flag, grants number, LLID and VLAN ID) in order to generate a rule with the set of required actions to apply (such as setting the number of grants and QoS policy, for example), and defines the priority for sending the packet to an output port. The SDN-EPON controller sends a FlowMod with the new rule via the OF-client to the OF-switch to be applied to the incoming GATE message. The incoming OF packet creates a new flow entry in the flow table through the asynchronous port of the OF-switch. Once the new flow entry is added to the flow table, the system waits for a MAC control client to initiate a GATE message request for the corresponding instance of a GATE entity. This instance generates a GATE message with the required fields (requested number of grants and policy to apply to the end user) and sends it through the corresponding synchronous logical port of the OF-switch. When a packet is received, a flow table lookup is performed by comparing the value of the packet fields against the matching fields of existing rules in order of priority. Once the packet is matched, the corresponding counter of the selected flow entry must be increased and the actions of the selected flow entry are applied immediately.

Once the packet is ready to be transmitted, the action of TransmitPending is executed on the packet by interacting with the register “P” in order to change the transmission state of the packet to “TransmitPending” (set 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” in order to change the transmission state of the packet to “TransmitInProgress” (set flag from 0 to 1 and change the flag of TransmitPending from 1 to 0). Then the action of TransmitEnable is executed when there is no any further packet to transmit by interacting with register “E”, so the transmission state of the packet is changed to enable (set

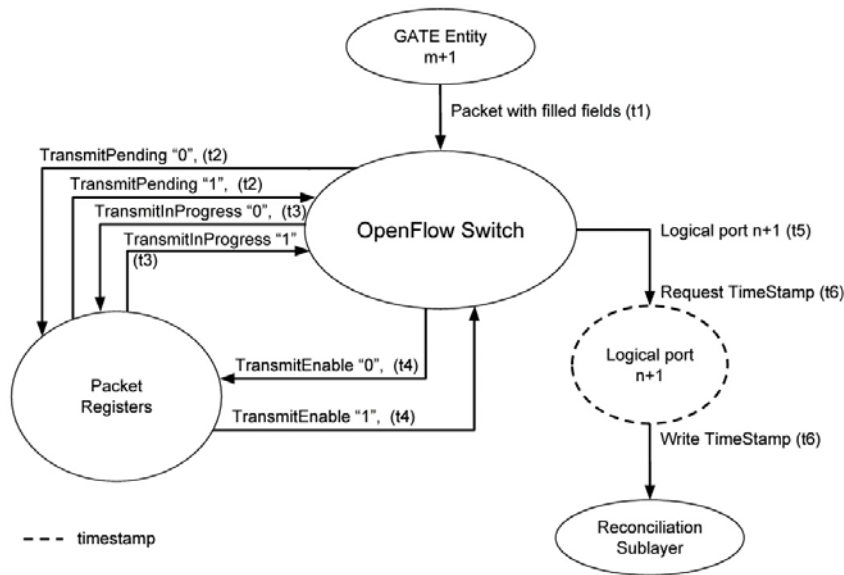


Figure 7. GATE related actions in OF-switch.

flag to 1 and change the flag of TransmitInProgress to 0). The next action is forwarding the packet to a certain logical output port. Finally the action of 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 7 shows the state diagram of the actions related to the GATE message, and Figure 8 illustrates the procedures followed for the GATE message.

3 Validation

To demonstrate the validity of our architecture, we built an scenario and ran some simulations to evaluate the performance of the proposed SDN-EPON architecture by using the OPNET Modeler package, following the work done in [16]. The result is compared with the legacy EPON case in terms of the OF-switch performance, delay and throughput. To this end, a network topology was created including one controller, one SDN-OLT, and 16 ONUs. OpenFlow is used as a standard interface between the SDN-EPON controller and the OF-switch. Physical distance between controller and SDN-OLT is set to 10 meters, while ONUs are at distances in the range of 16 to 18 Km. Users generate self-similar traffic with the packet sizes uniformly distributed between 64 bytes and 1518 bytes. Table 4 lists the parameters used in the simulation.

The validation comprises two parts:

- Low level SDN-OLT performance.

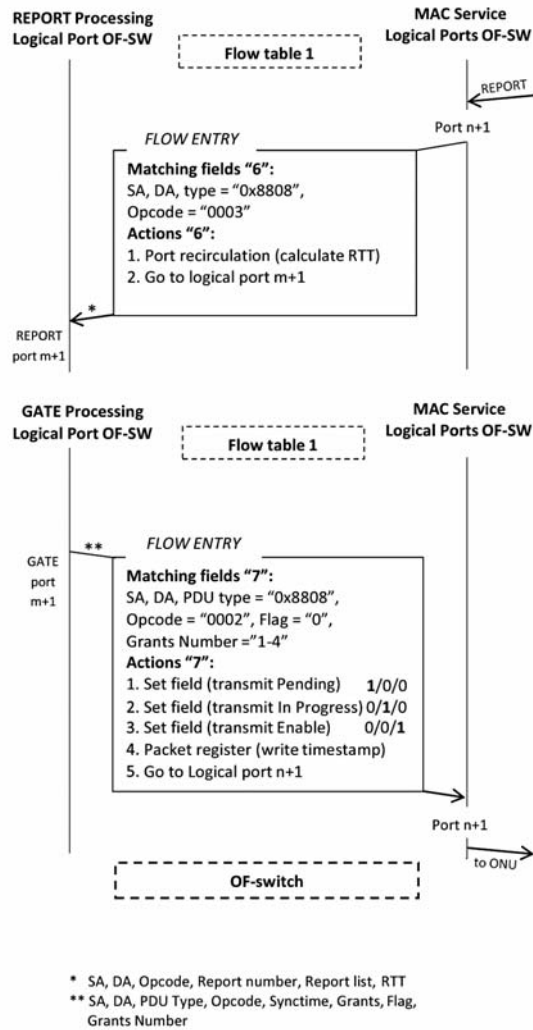


Figure 8. Procedures related to the processing of the GATE message in the OF-switch.

- High level considerations related to operations.

3.1 Low level SDN-OLT performance

In what follows we evaluate the proposed architecture in three specific aspects:

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

Figure 9 illustrates the measurement points for traffic delay (green flow), queuing delay and system throughput (orange flow).

Parameters	Value
Link speed	1 Gbps
Maximum cycle time	1ms
Guard time	1 μ s
Message processing delays in SDN-OLT/OLT	0.0164ms
DBA algorithm	IPACT
Hurst parameter	0.7
Network topology	Tree
OLT-ONU distance (d)	16 < d < 18 km

Table 4
Simulation parameters.

3.1.1 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 a PACKET-OUT message is $T_{controller-process} = 1.986\mu s$ with a 99% confidence interval of $\pm 0.0001\mu s$. The average delay⁵ 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 $T_{delay} = 5.243$ with a 99% confidence interval of $\pm 0.001\mu s$. Hence, the average response time of the controller for a packet is calculated as follows:

$$\begin{aligned}
 T_{controller-response} &= T_{controller-process} + 2 * (T_{delay}) \\
 &= 12.472\mu s
 \end{aligned}$$

3.1.2 Performance of the OF-switch

We now evaluate the performance of the SDN-OLT and compare it with the legacy OLT. We study two cases: 1) proactive system (presence of the rules-flow entries in the OpenFlow table) and 2) reactive system (absence of rules). In the proactive system, the controller has uploaded the set of rules such as GATE, REPORT and DISCOVERY GATE at the beginning of the connection, for all the ONUs. The delay measurement is performed from the instant

⁵ The average delay is composed of propagation time and transmission time.

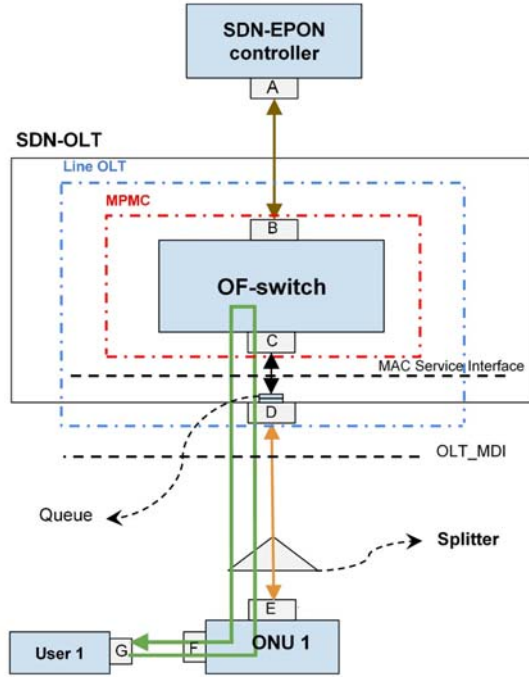


Figure 9. Measurement points in the evaluation.

when a packet (control frame/data frame) enters the OF-switch through a synchronous port, and includes the look-up in the flow table⁶ 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 $T_{p-OF-switch-process} = 0.488$ with a 99% confidence interval of $\pm 0.003\mu s$.

In the reactive system, in 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 OF interface and, based on that, the controller generates a FlowMod message and a PACKET-OUT message in order to set a rule in the flow table. The rest of actions of this case is 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 controller and vice versa $2 * T_{delay} = 10.486\mu s$, d) controller processing time $T_{controller-process} = 1.986\mu s$, and e) second time look-up and actions $T_{secondTime-lookup\&actions} = 0.488\mu s$.

$$T_{a-OF-switch-process} = T_{firstTime-lookup} + T_{transmission}$$

⁶ We defined one flow table and the length of flow entries are dynamic.

$$+T_{round-trip-delay} + T_{controller-process}$$

$$+T_{secondTime-lookup\&actions} = 13.353\mu s$$

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

Figure 10 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 back. This is not the real 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 globally. For the sake of brevity, we use the term "RTT" as a short name for the sum of the aforementioned delays. The figure 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). As shown (a green line) in Figure 9, 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). In order to interpret correctly the figures, it is important to note that blue and red data points do not coincide in the same x-axis in order to avoid the overlapping of the points. 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 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 (RTT increase at load 0.1 equal to $0.341ms$ to load 0.9 equal to $0.603ms$), while in the SDN scenario the RTT is almost constant and equal to $0.018ms$. The remarkably small confidence interval in the SDN scenario (99% confidence interval of $\pm 0.0005ms$) is due to the existence of pre-defined rules in the flow table of OF-switch, which makes its 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 units (one per instance) to receive and send packets, respectively, while in the SDN-EPON architecture the DBA only controls the OF-switch. Therefore the procedures of DBA related to the timing management and processing time of a packet inside the SDN-OLT are highly reduced.

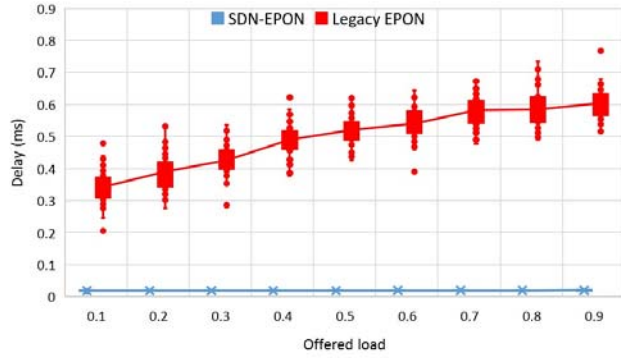


Figure 10. Comparison of RTT in SDN-EPON and legacy EPON scenarios under different offered load.

Figure 11 shows the downstream queuing delay versus the offered load at the OLT in both architectures. 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 it corresponds to measurements of the packet waiting times in the downstream queue. 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 11 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, (almost) independent of the offered load. 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. On the other hand, 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 (with the maximum of variation at the 0.4 load, possibly due to statistical randomness of the packet generator at this value).

Figure 12a and 12b 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 9 and the ONU port E. 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%. 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.

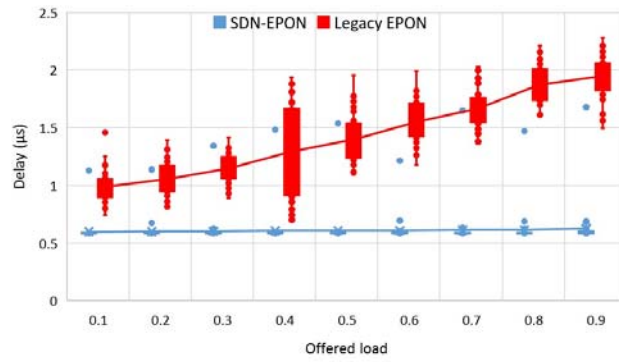
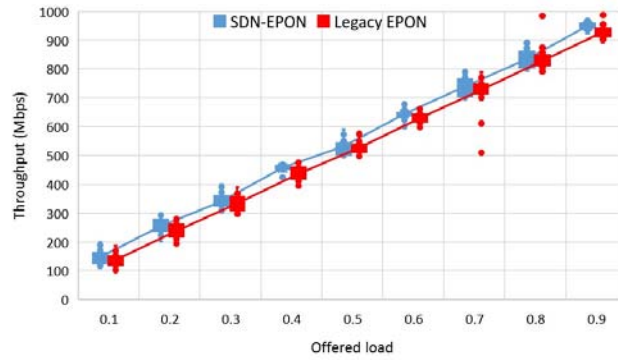
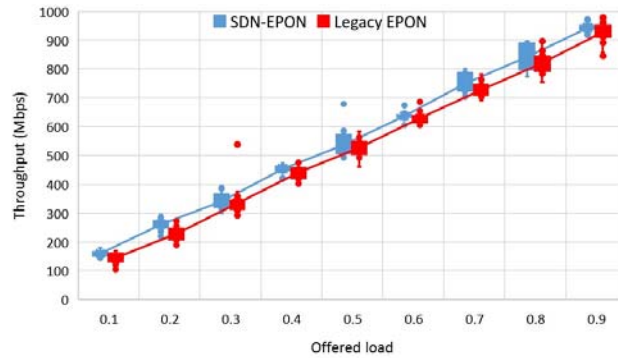


Figure 11. Comparison of queuing delays in SDN-EPON and legacy EPON scenarios under different offered load.



(a) Comparison of upstream throughput.



(b) Comparison of downstream throughput.

Figure 12. Comparison of throughput in SDN-EPON and legacy EPON scenarios under different offered load.

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

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 of 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 upload the specific rule to the OF-switch. The controller can generate a new FlowMod with actions corresponding to QoS and upload it in scale of 5 to 8 μ s in OF-switch, thus allowing very fast reconfigurations of the QoS profile assigned to each user.

4 Conclusion

This paper 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 of them in the SDN-OLT, 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 timescales can be migrated to the controller. By extending the messages and rules of OpenFlow v1.5 we were able to implement the synchronous operations required by EPON. Our design opens the way to multi-tenant and multi-provider PON networks, thanks to the flexibility provided by the virtualization, and allows the optimization of resource allocation, service availability, thus enhancing SLA and QoS requirements. We evaluated our approach with simulations, showing that the SDN-EPON approach is not only more flexible than the legacy EPON in terms of operations, but it also improves the average throughput of the ONUs in both the downstream and upstream channels and the average delay of data packets.

Acknowledgment

This work is partially supported and funded by the “Ministerio de Economía y Competitividad” of the Spanish Government through project “Redes troncales y de acceso inteligentes definidas por software” TEC2013-47960-C4-1-P.

References

- [1] IEEE P802.3av Standard for Information technology, Sept. 2009. 1.1
- [2] IEEE 1904.1 Standard for Service Interoperability in Ethernet Passive Optical Networks, IEEE Standard Association. Sept. 20, 2013, ISBN 9780738185101. 1.2
- [3] Glen Kramer, Lior Khernmsh, Fumio Daido, Alan Brown, Hosung Yoon, Ken-Ichi Suzuki, Wang Bo, "The IEEE 1904.1 Standard: SIEPON Architecture and Model", IEEE Communications Magazine, vol. 50 , pp. 98-108, Sept. 2012. 1.2, 1
- [4] Ivica Cale, Aida Salihovic, and Matija Ivekovic. "Gigabit passive optical network-GPON." In 2007 29th International Conference on Information Technology Interface, pp.679-684. IEEE, 2007. 1.2
- [5] Bruno Nunes, Manoel Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks." Communications Surveys & Tutorials, IEEE 16, no. 3, pp. 1617-1634, 2014. 1.3
- [6] Anu Mercian, Michael P. McGarry, Martin Reisslein, “Software Defined Optical Access Networks (SDOANs): A Comprehensive Survey”. IEEE Communications Surveys & Tutorials, vol. 18, no. 4, pp. 2738-2786, 2016. 1.4
- [7] Pawel Parol, Michel Pawlowski. "Towards networks of future: SDN paradigm introduction to PON networking for business applications." Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on, pp.829-836. IEEE, 2013. 1.4
- [8] Neda Cvijetic, "Software-defined optical access networks for multiple broadband access solutions." In OptoElectronics and Communications Conference and Photonics in Switching, p. TuP2_1. Optical Society of America, 2013. 1.4
- [9] M. Ruffini, F. Slyne, C. Bluemm, N. Kitsuwon, and S. McGettrick, "Software Defined Networking for Next Generation Converged Metro-Access Networks." Optical Fiber Technology, vol. 26, pp. 31-41, Dec. 2015. 1.4

- [10] Hui Yang, Jie Zhang, Yongli Zhao, Jialin Wu, Yuefeng Ji, Yi Lin, Jianrui Han, and Young Lee, "Experimental demonstration of remote unified control for OpenFlow-based software-defined optical access networks." *Photonic Network Communications*, 31(3), pp. 568-577, 2015. 1.4
- [11] Rentao Gu, Shizong Zhang, Yuefeng Ji, Tonglu Guo, and Xiaoxiong Wang, "Efficient software-defined passive optical network with network coding." *Photonic Network Communications*, 31(2), pp. 239-250, 2015. 1.4
- [12] Ahmed Amokrane, Jinho Hwang, Jin Xiao, and Nikos Anerousis, "Software defined enterprise passive optical network." In *10th International Conference on Network and Service Management (CNSM) and Workshop*, pp. 406-411. IEEE, 2014. 1.4
- [13] Hamzeh Khalili, David Rincón, and Sebastià Sallent, "Towards an Integrated SDN-NFV Architecture for EPON Networks." In *Advances in Communication Networking*, vol. 8846, pp. 74-84. Springer International Publishing, 2014. 1.4, 1.5, 2.1.1
- [14] Chengjun Li, Wei Guo, Wei Wang, Weisheng Hu, Ming Xia, "Programmable bandwidth management in software-defined EPON architecture." *Optics Communications*, vol. 370, pp. 43-48, 2016. 1.4
- [15] Toshihiko Kusano (Kawasaki, JP), Architecture for an Access Network System Management Protocol Control under Heterogeneous Network Management. US Patent Application 14/256,011, 2014. 1.4
- [16] Paola Garfias, "Resource Management Research in Ethernet Passive Optical Networks." PhD thesis, Universitat Politècnica de Catalunya UPC-BarcelonaTech, 2013.