ULPGC Universidad de Las Palmas de Gran Canaria | eite ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA

TRABAJO DE FIN DE GRADO

Red GPON experimental virtualizada (NFV) mediante gestión definida por software (SDN)

Titulación: Grado en Ingeniería en Tecnologías de la Telecomunicación

Mención: Telemática

Autor: GABRIEL ERNESTO LARES ASPERA

Tutor: CARLOS MIGUEL RAMÍREZ CASAÑAS

Fecha: Junio 2025

Resumen

Las redes ópticas pasivas con capacidad de gigabit (GPON) se han consolidado como la tecnología predominante para el despliegue de redes de fibra hasta el hogar (FTTH), gracias a su eficiencia y capacidad para ofrecer servicios de banda ancha. Sin embargo, su arquitectura tradicional presenta limitaciones en cuanto a flexibilidad, agilidad en la provisión de servicios y costes operativos derivados de una gestión dependiente del *hardware* y de configuraciones manuales. Para superar estos desafíos, paradigmas emergentes como la Virtualización de Funciones de Red (NFV) y las Redes Definidas por *Software* (SDN) ofrecen un enfoque innovador.

El presente Trabajo de Fin de Grado aborda el diseño y la implementación de una maqueta experimental que integra estas tecnologías, desarrollando una arquitectura de red GPON virtualizada y gestionada de forma centralizada mediante un controlador SDN. Los objetivos principales incluyen la implementación virtualizada de servicios de red sobre la infraestructura GPON, la posterior integración de gestión SDN centralizada, y demostrar la viabilidad de desacoplar el plano de control del plano de datos en un entorno de acceso óptico, logrando así una red más programable, automatizada y escalable.

Para lograrlo, se desarrolla un prototipo experimental que combina componentes físicos de una red GPON (Terminal de Línea Óptica, cableado de fibra óptica y Unidades de Red Óptica) con servicios virtualizados implementados en máquinas virtuales. Sobre esta infraestructura híbrida, se despliega el controlador SDN ONOS (*Open Network Operating System*) como cerebro de la red. La gestión y el control de los elementos de red se abstraen y centralizan en ONOS, permitiendo la configuración de la conectividad y las políticas de red a través de su interfaz de alto nivel.

El proyecto explora en detalle el funcionamiento de las redes GPON, así como las capacidades de gestión que ofrece el controlador, utilizando su API *REST* para la manipulación de políticas de conectividad, la imposición de reglas de calidad de servicio y la administración de flujos de datos en los dispositivos de red. Este enfoque definido por *software* no solo simplifica la operación de la red, sino que también abre la puerta a la automatización de la provisión de servicios y a una gestión del tráfico más dinámica y eficiente, sentando las bases para futuras redes de acceso más inteligentes y adaptables.

Abstract

Gigabit-capable Passive Optical Networks (GPON) have become the predominant technology for Fiber-to-the-Home (FTTH) network deployment, thanks to their efficiency and capacity to deliver broadband services. However, their traditional architecture presents limitations in terms of flexibility, service provisioning agility, and operational costs derived from hardware-dependent management and manual configurations. To overcome these challenges, emerging paradigms such as Network Function Virtualization (NFV) and Software-Defined Networking (SDN) offer an innovative approach.

This Bachelor's Thesis addresses the design and implementation of an experimental platform that integrates these technologies, developing a virtualized GPON network architecture managed centrally through an SDN controller. The main objectives include the virtualized implementation of network services over the GPON infrastructure, the subsequent integration of centralized SDN management, and demonstrating the feasibility of decoupling the control plane from the data plane in an optical access environment, achieving a more programmable, automated, and scalable network.

To accomplish this, an experimental prototype is developed that combines physical components of a GPON network (Optical Line Terminal, fiber optic cabling, and Optical Network Units) with virtualized services implemented in virtual machines. On top of this hybrid infrastructure, the SDN controller ONOS (Open Network Operating System) is deployed as the network's brain. The management and control of network elements are abstracted and centralized in ONOS, enabling connectivity configuration and network policy management through its high-level interface.

The project explores in detail the operation of GPON networks, as well as the management capabilities offered by the controller, using its REST API for connectivity policy manipulation, quality of service rule enforcement, and data flow administration in network devices. This software-defined approach not only simplifies network operation but also opens the door to service provisioning automation and more dynamic and efficient traffic management, laying the foundation for future more intelligent and adaptable access networks.

Agradecimientos

En este apartado quiero dedicar unas palabras a las personas que han estado a mi lado durante este camino. Empezando especialmente por mi familia, que siempre ha confiado en mí y me ha dado ese apoyo incondicional para cumplir todos mis objetivos; a mi pareja, que no ha dudado ni un segundo de mí, en las buenas y en las malas, siempre a mi lado; y a mis amigos, que también han sido un gran apoyo, demostrando ser personas increíbles y únicas.

En segundo lugar, también he de agradecer a profesores, tutores y personal de la ULPGC que ha formado parte de mi vida durante estos años, por lo que también se merecen un lugar en esta mención.

Y finalmente, agradecerme a mí mismo por siempre confiar en mi intuición y en que cualquier obstáculo que me ponga la vida por delante, seré capaz de superarlo.

Tabla de contenido

Índice de Tablas

Índice de Figuras

Lista de acrónimos

NAT *Network Address Translation*

AES-128 *Advanced Encryption Standard 128-bit*

API *Application Programming Interface*

ARP *Address Resolution Protocol*

BIP *Bit-Interleaved Parity*

BWmap *BandWidth Map*

CAPEX *Capital Expenditures*

CCTV *Closed Circuit Television*

CI/CD *Continuous Integration/Continuous Deployment*

CLI *Command Line Interface*

CORD *Central Office Re-architected as a Datacenter*

C-Tag *Customer Tag*

DBRu *Dynamic Bandwidth Report upstream*

DHCP *Dynamic Host Configuration Protocol*

DNS *Domain Name System*

ETSI *European Telecommunications Standards Institute*

FCS *Frame Check Sequence*

FEC *Forward Error Correction*

FTTB *Fiber To The Building*

FTTC *Fiber To The Curb*

FTTH *Fiber-to-the-Home*

FTTx *Fiber To The x*

GEM *GPON Encapsulation Method*

GPON *Gigabit-capable Passive Optical Networks*

GTC *GPON Transmission Convergence*

HEC *Header Error Control*

ICMP *Internet Control Message Protocol*

IGMP *Internet Group Management Protocol*

IGMPv3 *Internet Group Management Protocol version 3*

IP *Internet Protocol*

IPTV *Internet Protocol Television*

IPv4 *Internet Protocol version 4*

ITU-T *International Telecommunication Union Telecommunication Standardization Sector*

KVM *Kernel-based Virtual Machine*

LAN *Local Area Network*

LLDP *Link Layer Discovery Protocol*

LTS *Long Term Support*

MAC *Media Access Control*

MANO *Management and Orchestration*

NAPT *Network Address Port Translation*

NFV *Network Function Virtualization*

NFVI *Network Function Virtualization Infrastructure*

NFVO *NFV Orchestrator*

NVR *Network Video Recorder*

OAM *Operations, Administration, and Maintenance*

ODL *OpenDaylight*

OLT *Optical Line Terminal*

OMCI *ONU Management and Control Interface*

ONF *Open Networking Foundation*

ONOS *Open Network Operating System*

ONT *Optical Network Terminal*

ONU *Optical Network Unit*

ONU-ID *ONU Identifier*

OPEX *Operational Expenditures*

OvS *Open vSwitch*

OVSDB *Open vSwitch Database*

PBX *Private Branch Exchange*

PLI *Payload Length Indicator*

PLOAMu *Physical Layer OAM upstream*

PoE *Power over Ethernet*

POTS *Plain Old Telephone Service*

PPPoE *Point-to-Point Protocol over Ethernet*

PTI *Payload Type Indicator*

PTP *Physical Termination Point*

PTPID *Physical Termination Point Identifier*

QoS *Quality of Service*

RTP *Real-time Transport Protocol*

SC/APC *Subscriber Connector/Angled Physical Contact*

SDN *Software-Defined Networking*

SIP *Session Initiation Protocol*

S-Tag *Service Tag*

SSM *Single Source Multicast*

TGMS *Telnet GPON Management System*

User-Tag *User Tag*

VIM *Virtual Infrastructure Manager*

VLAN *Virtual Local Area Network*

vBNG *BroadBand Network Gateway virtual*

vFirewall *virtual Firewall*

VIM *Virtual Infrastructure Manager*

vOLT *virtual OLT*

VoIP *Voice over IP*

VPN *Virtual Private Network*

vRouter *virtual Router*

VMS *Video Management System*

VNFM *VNF Manager*

VNFs *Virtualized Network Functions*

WAN *Wide Area Network*

Capítulo 1. Introducción y Objetivos

1.1. Introducción

El presente Trabajo Fin de Grado (TFG) se centra en el estudio e implementación de una maqueta experimental de red GPON (*Gigabit Passive Optical Network*), integrando tecnologías de virtualización de funciones de red (NFV) y redes definidas por *software* (SDN). Este proyecto aborda la convergencia de estas tecnologías emergentes en el contexto de las redes de acceso óptico, explorando las posibilidades de crear arquitecturas de red más flexibles, eficientes y gestionables.

1.2. Antecedentes y estado del arte

Contexto actual

Las redes de acceso óptico pasivo (PON) se han convertido en la tecnología predominante para ofrecer conectividad FTTH a nivel global, con previsiones de superar 200 millones de hogares pasados en Europa para 2026 (frente a 88 millones en 2019) [1].

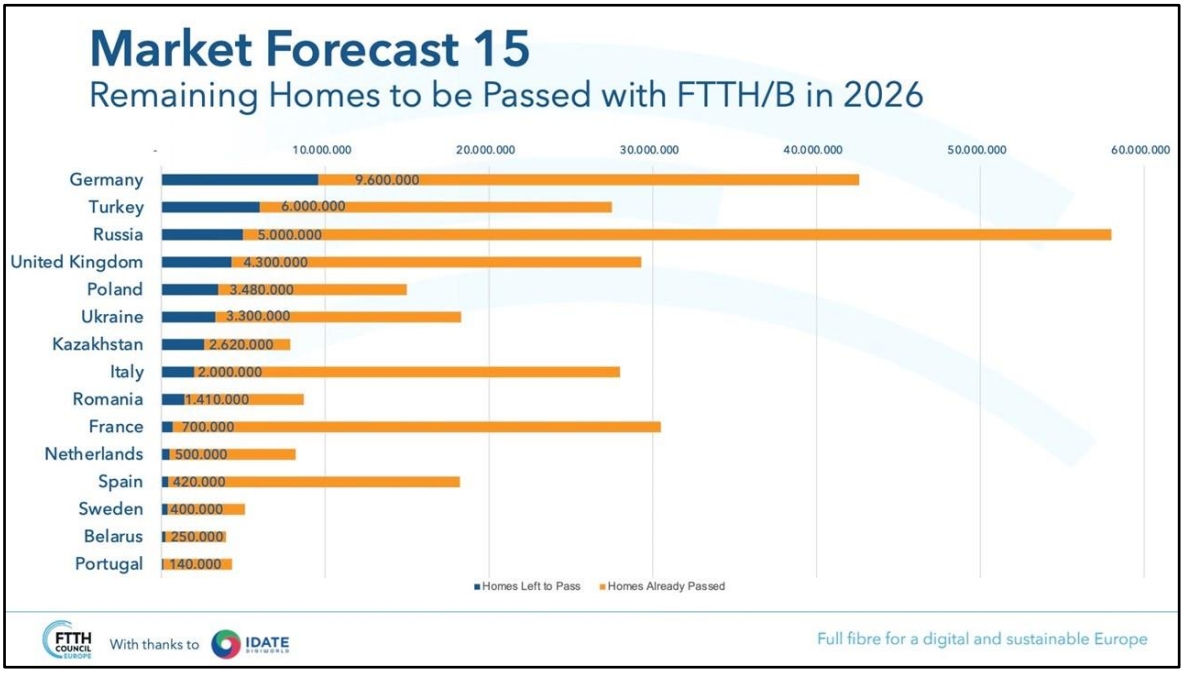


Figura 1. Previsión de viviendas con FTTH/B pendientes y ya pasadas en 2026 (Europa) [54].

No obstante, la arquitectura tradicional de GPON basada en equipos propietarios (OLT, *Optical Line Terminal*, y ONT's, *Optical Network Units*) con planos de control integrados presenta limitaciones en cuanto a flexibilidad, escalabilidad e interoperabilidad *multi-vendor* [2]. Estas limitaciones ralentizan la introducción de nuevos servicios y complican la gestión unificada de la red. Para abordar este problema, la industria ha adoptado los paradigmas de SDN (*Software Defined Networking*) y NFV (*Network Functions Virtualization*) en el acceso óptico, buscando redes más ágiles y programables.

En una arquitectura SDN el plano de control se desacopla del plano de datos de la red, delegándose las decisiones de encaminamiento a un controlador de *software* centralizado [1]. De este modo, los dispositivos de la red (p. ej., conmutadores u OLT) pasan a ser elementos programables gobernados mediante APIs (*Application Programming Interface*) y protocolos abiertos (*OpenFlow*, *NETCONF*, etc.) por aplicaciones de control externas [1]. Por su parte, NFV traslada funciones de red que antes requerían *hardware* dedicado hacia máquinas virtuales o contenedores sobre infraestructura genérica, facilitando su despliegue flexible en centros de datos [3]. La combinación de SDN y NFV permite entonces desagregar los equipos de acceso óptico: separar el *hardware* físico (p. ej., chasis OLT) de las funciones lógicas de control y servicio, que pueden ejecutarse en la nube. Esta tendencia ha motivado esfuerzos de estandarización como CloudCO de Broadband Forum, iniciado en 2016 para definir una "central" de acceso totalmente virtualizada mediante SDN/NFV [4]. CloudCO proporciona una arquitectura y especificaciones de interfaz que habilitan la migración de nodos de acceso tradicionales hacia nodos desagregados controlados por *software* [5]. En esencia, esto permite usar *hardware* genérico *white-box* en el acceso (en lugar de equipos cerrados), moviendo inteligencia de las OLT a plataformas *cloud*, con la promesa de reducir costes y agilizar la introducción de mejoras mediante *software* [5].

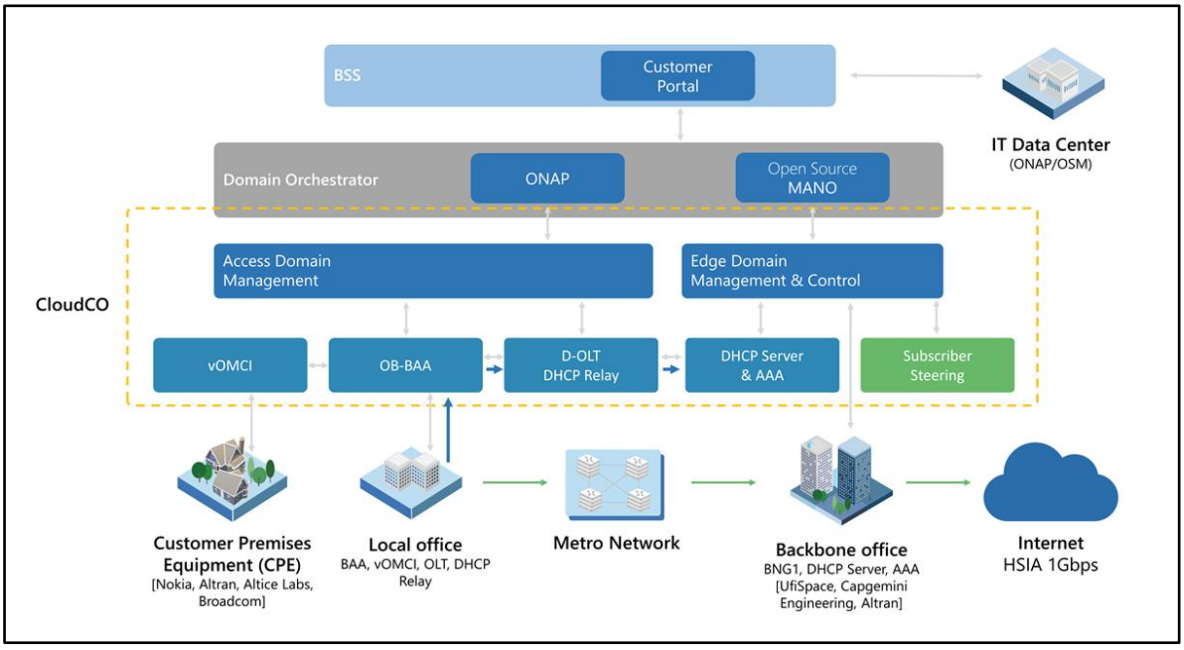


Figura 2. Arquitectura de referencia CloudCO: desagregación del acceso mediante SDN/NFV [55].

Uso de *Open vSwitch* (OvS) en redes de acceso virtualizadas

Open vSwitch (OvS) se ha consolidado como un componente fundamental en entornos de virtualización de funciones de red (NFV) y redes definidas por *software* (SDN). OvS es un conmutador virtual de propósito general, integrado en el *kernel* de Linux desde la versión 3.3, que soporta interfaces y protocolos estándar de gestión de red como OpenFlow [6]. Gracias a su naturaleza de código abierto y alto rendimiento, OvS se utiliza ampliamente tanto en entornos de Centro de Datos *Cloud* como en aplicaciones SDN de operadores, actuando como *vSwitch* para interconectar funciones virtualizadas [6]. Asimismo, con optimizaciones como DPDK, OvS puede alcanzar velocidades de conmutación del orden de 40 Gbps sobre *hardware* común, logrando mejoras de rendimiento de hasta 12× respecto a configuraciones estándar [6]. Estas capacidades hacen que OvS sea una solución idónea para virtualizar redes de acceso de alta velocidad, asegurando que un *switch software* pueda manejar el tráfico de tecnologías como GPON y evoluciones (por ejemplo, XGS-PON a 10 Gbps).

En el contexto de las redes de acceso ópticas, la adopción de SDN ha dado lugar a arquitecturas donde la red GPON se abstrae como un único conmutador virtual controlable centralmente. Diversos trabajos han demostrado que es posible dotar a un sistema GPON comercial de capacidades SDN, de forma que toda la red GPON se comporta como un *switch OpenFlow* manejado por un controlador SDN [1]. Por ejemplo, autores como Lee *et*

al. implementaron un OLT GPON habilitado para SDN mediante la incorporación de *switches* virtuales *OpenFlow* en el propio OLT, emulando una capa de conmutación programable sobre el equipo tradicional [1]. De manera análoga, Rouskas *et al.* exploraron la inclusión de conmutadores virtuales dentro de las ONUs en arquitecturas PON de nueva generación (TWDM-PON) para posibilitar la selección dinámica de canales OFDM en el tramo de abonado [1]. En todas estas propuestas, OvS (u otras implementaciones equivalentes de *software switching*) actúa como elemento de datos programable, permitiendo que las instrucciones del controlador SDN se apliquen en la red de acceso.

Un caso ilustrativo es la arquitectura presentada por Lee *et al.* [7]. En su diseño, dado que el OLT GPON carecía de soporte nativo para protocolos SDN, se integró un agente *OpenFlow* dentro del OLT encargado de traducir y ejecutar las órdenes del controlador SDN [7]. Adicionalmente, se añadió un *switch OpenFlow* auxiliar -basado en OvS- para el reenvío de paquetes entre las distintas ONUs, así como para implementar funcionalidades avanzadas de red que el conmutador *Ethernet* interno del OLT no ofrecía [7]. Este *switch* abierto proporcionó capacidades de encaminamiento y aislamiento de tráfico entre ONUs, recolección de estadísticas, etiquetado VLAN por usuario y *metering* (control de ancho de banda) de manera flexible [7]. Gracias a OvS, la red GPON pudo gestionarse como un conmutador unificado, interconectando múltiples sitios distribuidos y aplicando políticas de tráfico programables, algo inviable con la electrónica GPON convencional. En suma, esta integración de OvS solventó limitaciones del *hardware* legado, como la imposibilidad de conmutar tráfico directamente entre puertos ONU en un OLT tradicional, a la vez que habilitó un control más fino del ancho de banda por abonado [7].

Asimismo, en el ámbito académico y de plataformas abiertas se encuentran ejemplos donde OvS es pieza clave para virtualizar la capa de acceso. Merayo *et al.* describen una implementación SDN/NFV sobre un banco de pruebas GPON *legacy* en la cual se despliega OvS tanto a nivel de OLT como en las ubicaciones de las ONTs, todos bajo el control de un controlador *OpenDaylight* (ODL) [1]. En esta plataforma, un controlador SDN central gestiona, vía *OpenFlow*, las instancias de OvS colocadas junto a los equipos GPON, logrando reconfigurar en tiempo real parámetros de servicio (p. ej., garantías de QoS por usuario) de acuerdo con las demandas de tráfico [1]. Dado que los OLT/ONT comerciales no soportan SDN de forma nativa, los autores insertan OvS en computadores adyacentes (un OVS central conectado al OLT y pequeños OVS residenciales en cada ONU) para emular la funcionalidad SDN en ambos extremos de la red de acceso [1]. Esta aproximación confirmó la viabilidad de introducir SDN en redes GPON reales, permitiendo políticas de gestión dinámicas sobre equipamiento antiguo y facilitando nuevos modelos

de negocio en redes de banda ancha residencial [1]. De hecho, los resultados experimentales muestran que es posible ajustar de forma ágil los perfiles de servicio de los abonados (ancho de banda garantizado, prioridades, etc.) mediante flujos *OpenFlow* aplicados en OvS, todo ello sin modificar el *hardware* PON subyacente [1].

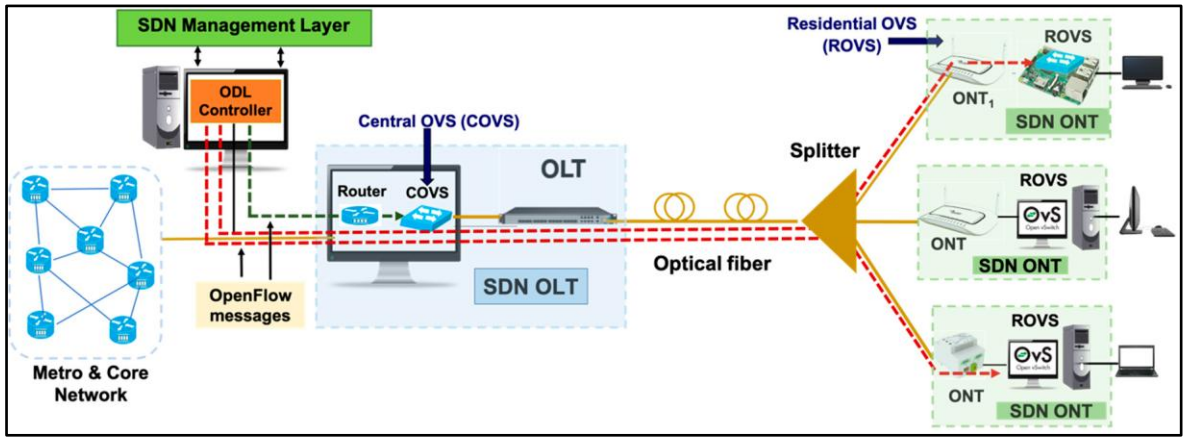


Figura 3. Plataforma experimental GPON heredada gestionada por SDN con OpenDaylight y Open vSwitch (COVS/ROVS) [1].

Iniciativas industriales de código abierto también han adoptado enfoques similares. Por ejemplo, la plataforma CORD/SEBA (de la ONF) incluye el proyecto VOLTHA (*Virtual OLT Hardware Abstraction*), que propone abstraer los sistemas PON como *switches Ethernet* programables controlados por un controlador SDN [1]. En VOLTHA, las OLT y ONT físicas quedan envueltas por una capa de *software* (frecuentemente soportada por OvS u otros elementos de conmutación virtual) que expone interfaces estándares (*OpenFlow/NETCONF*) hacia el orquestador SDN, permitiendo gestionar la configuración de la red de acceso de forma unificada [1]. Esta filosofía sigue la tendencia de llevar los principios de las redes definidas por *software* y la virtualización a la "última milla" de la red, reduciendo la dependencia de *hardware* propietario y facilitando la implementación de funciones de acceso como VNFs (*Virtual Network Functions*). Estudios recientes incluso plantean que la virtualización SDN/NFV en el acceso posibilitará escenarios de *multi-tenant* u *open access*, donde múltiples operadores comparten la misma infraestructura PON de manera flexible [8]. Gracias a OvS y tecnologías afines, es factible lograr un *unbundling* virtual de la red de fibra, asignando particiones lógicas o *slices* de ancho de banda a distintos proveedores sobre un mismo despliegue físico, con aislamiento y control de calidad garantizados [8].

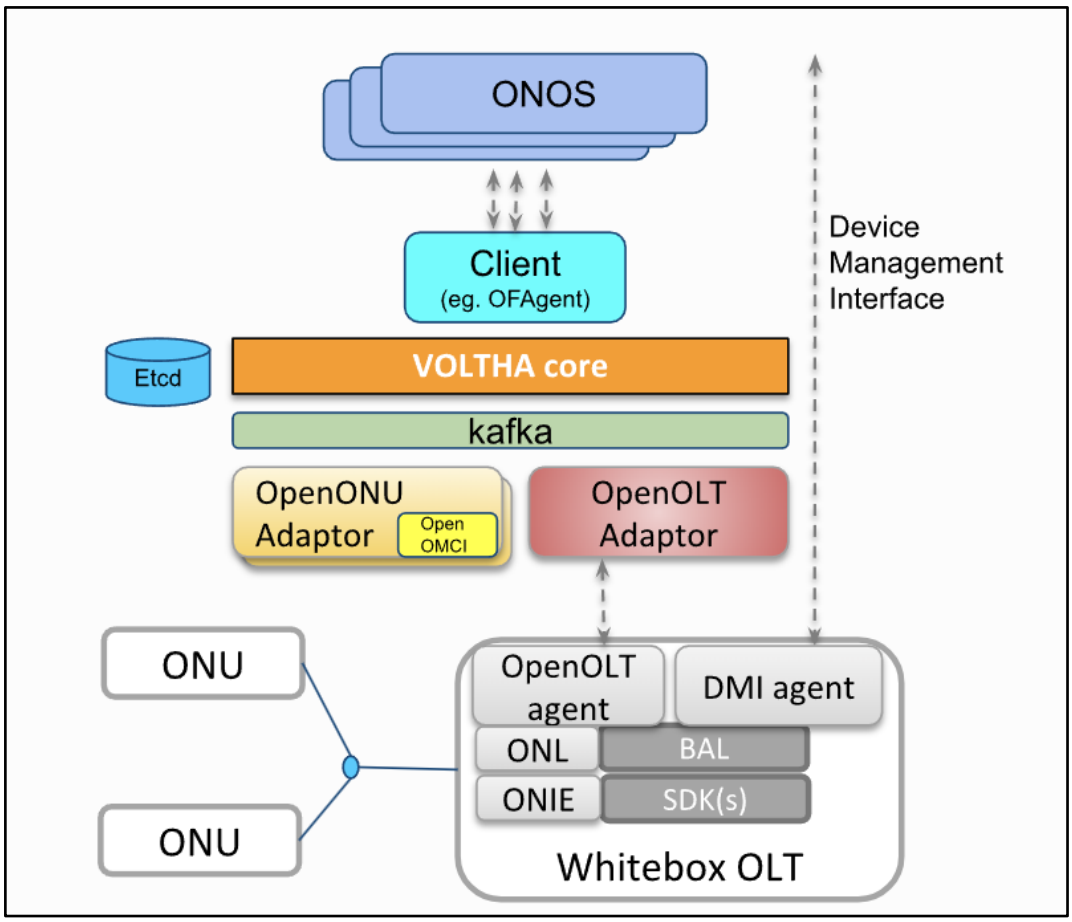


Figura 4. Pila VOLTHA y OLT de caja blanca controlados por ONOS en SEBA [56].

En resumen, OvS se ha convertido en una solución común en la virtualización de redes de acceso debido a varias ventajas clave. En primer lugar, su compatibilidad con protocolos SDN estándar (*OpenFlow*, OVSDB) y su amplia adopción garantizan la interoperabilidad con controladores y orquestadores de distintos fabricantes [6]. En segundo lugar, al ser *software open-source* ejecutable sobre *hardware commodity*, OvS reduce costes y evita la dependencia de equipos específicos, a la vez que puede escalar en rendimiento mediante optimizaciones *software* (p.ej., uso de DPDK para acelerar el plano de datos) [6]. Además, OvS ofrece una gran flexibilidad para implementar funcionalidades avanzadas en la red de acceso desde segmentación de tráfico por VLAN o por usuario, hasta control de ancho de banda, recopilación de métricas y aplicación de políticas de QoS todo ello mediante reglas programables que el controlador SDN instala dinámicamente en los switches virtuales [7]. Estas capacidades no solo mejoran la agilidad y programabilidad de las redes de acceso, sino que habilitan nuevos modelos operativos como la coexistencia de múltiples servicios y operadores sobre la misma infraestructura, gracias al aislamiento lógico que brinda la

virtualización [8]. Por las razones anteriores, numerosos proyectos académicos y plataformas experimentales han incorporado OvS como elemento de datos en sus arquitecturas SDN de acceso, validando su desempeño y beneficios en escenarios GPON virtualizados [1]. En consonancia con este estado del arte, la propuesta implementada en este TFG emplea OvS como conmutador virtual SDN dentro de la red GPON experimental, aprovechando su flexibilidad y control centralizado para gestionar el acceso de banda ancha de manera eficiente y alineada con las soluciones investigadas previamente.

1.3. Objetivos

Este TFG tiene como objetivo general el diseñar e implementar una red GPON experimental que integre funciones virtualizadas (NFV), gestionada mediante un controlador SDN y soportada sobre un entorno de virtualización, a fin de demostrar la viabilidad y las ventajas de esta aproximación frente a las redes tradicionales.

El objetivo general anteriormente descrito se llevará a cabo mediante el alcance de los siguientes objetivos específicos:

O1. Estudio e implementación de una red GPON estándar.

O2. Estudio e implementación de servicios virtualizados (NVF) en la red GPON.

O3. Estudio e implementación de un controlador SDN para la gestión centralizada de la red GPON.

O4. Integración, puesta en funcionamiento y verificación de la red GPON virtualizada.

1.4. Contenidos

La presente memoria se estructura en seis capítulos, además de la bibliografía, el pliego de condiciones, el presupuesto y los anexos correspondientes. A continuación, se describe brevemente el contenido de cada capítulo:

Capítulo 1: Introducción

Presenta el contexto del trabajo desarrollado, estableciendo los antecedentes basados en proyectos previos y definiendo los objetivos que se pretenden alcanzar a lo largo del desarrollo del TFG.

Capítulo 2: Fundamentos teóricos: SDN, NFV

Establece las bases teóricas de las tecnologías SDN (*Software Defined Networking*) y NFV (*Network Function Virtualization*), proporcionando el marco conceptual necesario para la comprensión del proyecto.

Capítulo 3: Fundamentos teóricos GPON

Desarrolla los conceptos fundamentales de la tecnología GPON (*Gigabit Passive Optical Network*), abordando su arquitectura, componentes y funcionamiento.

Capítulo 4: Servicios y aplicaciones NFV

Describe las funciones de red virtualizadas y los servicios empleados en la implementación del sistema, detallando su configuración y aplicación específica en este TFG.

Capítulo 5: Implementación del Escenario GPON-SDN

Explica detalladamente el proceso de diseño e implementación del escenario experimental GPON-SDN, documentando paso a paso la metodología seguida.

Capítulo 6: Resultados finales y conclusiones

Presenta las pruebas de funcionamiento realizadas sobre el escenario desarrollado, incluyendo los resultados obtenidos, su análisis y las conclusiones de estos.

Capítulo 2. Fundamentos teóricos: SDN, NFV

2.1. Introducción

Este capítulo resume los fundamentos tecnológicos que sustentan la red GPON virtualizada del TFG. En primer lugar, se describen los principios clave de las redes definidas por *software* (SDN) y la separación entre planos de datos, control y aplicación. Se profundiza en el controlador SDN de *ONOS*, destacando su arquitectura distribuida y las razones de su elección frente a otras opciones (*OpenDaylight, Ryu, Floodlight*). Luego se presenta el protocolo *OpenFlow* como interfaz de control dominante entre ONOS y los conmutadores. A continuación, se introduce la virtualización de funciones de red (NFV) siguiendo la arquitectura ETSI (NFVI–VNFs–MANO) y sus beneficios de flexibilidad y ahorro de costes. El capítulo concluye analizando la sinergia SDN + NFV para redes ópticas de acceso, base sobre la que se construye la maqueta GPON experimental descrita en los capítulos siguientes.

2.2. SDN

2.2.1. Definición

Las redes definidas por *software* (SDN, *Software-Defined Networking*) representan un paradigma de redes en el que se desacoplan los mecanismos de control de red de los dispositivos de comunicaciones. Es decir, se separa el plano de control (*control plane*) del plano de datos (*data plane*) de la infraestructura subyacente [9]. En lugar de que cada conmutador o enrutador tome decisiones de encaminamiento de forma autónoma (como ocurre en las redes tradicionales), en SDN dichas decisiones se trasladan a un *software* de control centralizado, denominado controlador SDN, que gestiona de forma programable el comportamiento de la red completa. Este enfoque proporciona una arquitectura de red más flexible y programable, ya que las lógicas de control se implementan por medio *software* externo y pueden adaptarse o reemplazarse con mayor facilidad que en *hardware* especifico.

2.2.2. Arquitectura

En términos arquitectónicos, SDN se suele conceptualizar en tres planos o capas principales [9]:

- ***Plano de datos (data plane):*** formado por los dispositivos de conmutación o encaminamiento (conmutadores, *routers*, etc.) que se encargan del reenvío de paquetes. Estos dispositivos ejecutan acciones simples sobre el tráfico (reenviar, filtrar o modificar paquetes) siguiendo las reglas que les indica el plano de control. El plano de datos corresponde al *hardware* de red (físico o virtual) y también es referido como plano de encaminamiento o reenvío. Su funcionalidad se limita a cumplir las instrucciones que recibe, típicamente consultando tablas de flujo para cada paquete que procesa.

- ***Plano de control (control plane):*** es el "cerebro" de la red [9]. Se ocupa de calcular y decidir cómo deben manejarse los flujos de tráfico en el plano de datos, determinando las rutas, políticas y reglas de reenvío. En SDN, el plano de control está centralizado lógicamente en uno o varios controladores *software*, en lugar de distribuido en cada nodo de red. Este controlador tiene una visión global de la topología y estado de la red, y mediante protocolos especializados comunica sus decisiones a los elementos del plano de datos. Gracias a esta centralización, se facilita la optimización global, la implementación de políticas coherentes y una gestión más sencilla de la red.

- ***Plano de aplicación (application plane):*** capa superior donde residen las aplicaciones y servicios de red que aprovechan la abstracción proporcionada por el controlador [9]. Estas aplicaciones (por ejemplo, sistemas de orquestación de tráfico, balanceo de carga, *firewalls* definidos por *software*, sistemas de detección de intrusos, etc.) interactúan con el controlador a través de interfaces norte (*Northbound APIs*). De este modo, pueden programar el comportamiento de la red sin tener que preocuparse por los detalles de bajo nivel del *hardware*. El plano de aplicación introduce innovación y flexibilidad, permitiendo a desarrolladores y operadores implementar nuevas funcionalidades de red de manera ágil sobre la infraestructura SDN.

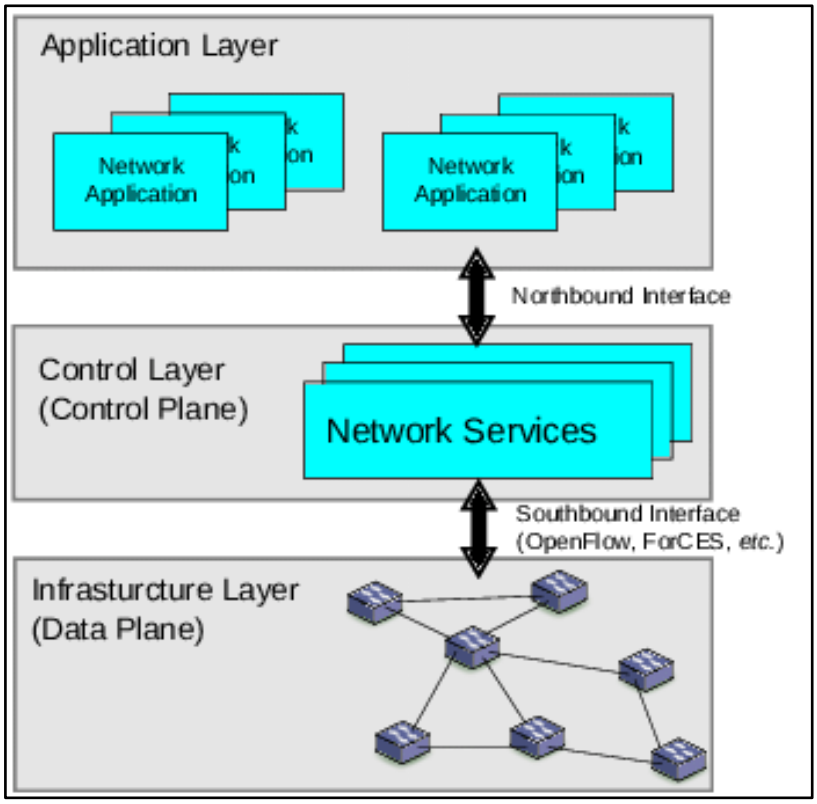


Figura 5. Arquitectura SDN [57].

Es importante destacar que algunos modelos también consideran un plano de gestión (o de orquestación) adicional, encargado de la configuración general y la coordinación de la red. No obstante, en el contexto de este trabajo nos centraremos en los tres planos fundamentales mencionados: datos, control y aplicación. Estos planos se comunican mediante interfaces bien definidas: típicamente, el controlador SDN expone API hacia las aplicaciones (interfaz norte) y utiliza protocolos de control hacia los dispositivos (interfaz sur). Esta separación de tareas hace posible una administración centralizada y programable de la red. Diversos estudios señalan que la arquitectura SDN aporta dinamismo, flexibilidad y capacidad de gestión superiores a las redes tradicionales [9], al separar las funciones estratégicas de control (*software*) de la función de reenvío de datos (*hardware*).

2.2.3. El controlador SDN (ONOS)

Uno de los componentes clave en cualquier despliegue SDN es el controlador. Existen múltiples controladores SDN de código abierto en la actualidad; en este trabajo se ha optado por ONOS (*Open Network Operating System*) como plataforma de control. ONOS es un sistema operativo de red diseñado específicamente para entornos SDN de nivel

operador (*carrier-grade*), enfocado en ofrecer alta escalabilidad, disponibilidad y rendimiento en redes de proveedores de servicios [10]. El proyecto *ONOS* fue iniciado por ON.Lab (*Open Networking Laboratory*) y tuvo su primera publicación abierta en 2014 [10], pasando posteriormente a la órbita de la Fundación Linux. A continuación, se detallan las características de ONOS, su arquitectura modular, ventajas y casos de uso, así como las razones de su elección frente a otras opciones como *OpenDaylight, Ryu o Floodlight*.

Características destacables y arquitectura de ONOS:

ONOS se concibe como un controlador distribuido y tolerante a fallos. A diferencia de controladores más sencillos que operan en una única instancia, ONOS soporta despliegues en *cluster* (con múltiples instancias cooperando) para lograr alta disponibilidad y resiliencia. Su núcleo está desarrollado en Java y corre sobre un contenedor OSGi (Apache Karaf), lo que le brinda una arquitectura modular y extensible similar a la de *OpenDaylight* [11]. ONOS está diseñado para que varias instancias trabajen juntas de forma coordinada: conforman un único plano de control lógico, aunque estén físicamente distribuidas en distintos servidores. Si uno de los nodos del *cluster* falla o necesita ser actualizado, los otros toman el relevo sin interrumpir el servicio, garantizando un control de red continuo [11]. Esta arquitectura distribuida y lógicamente centralizada permite escalar horizontalmente el controlador para manejar un gran número de dispositivos y eventos en la red.

Otra característica destacada de *ONOS* es su enfoque en la modularidad. Cuenta con un núcleo ligero al que se agregan módulos o aplicaciones para funciones específicas (enrutamiento, gestión de topología, cortafuegos, etc.). Mantiene separadas las llamadas “flujos norte-sur” de comunicación (entre aplicaciones y controlador, y entre controlador y dispositivos) de los flujos “este-oeste” (comunicación interna entre instancias del controlador), facilitando la personalización y extensión del sistema sin afectar su estabilidad [10]. En el plano de aplicación, *ONOS* ofrece una API norte abstracta, incluyendo un *framework* de intenciones (*Intent Framework*) que permite a las aplicaciones expresar requerimientos de alto nivel (intenciones de conectividad, ancho de banda, aislamiento, etc.), delegando en el controlador la traducción automática de esas intenciones en reglas concretas de flujo [10]. En el plano de datos, *ONOS* soporta múltiples protocolos sur para comunicarse con los dispositivos: principalmente *OpenFlow*, pero también NETCONF, gNMI, BGP-LS, PCEP, OVSDB, entre otros, e incluso lenguajes como P4Runtime, gracias a su diseño abstracto de proveedor de dispositivos [10][11]. Esta capacidad multi-protocolo le otorga flexibilidad para integrarse en diversos entornos y controlar equipamiento heterogéneo.

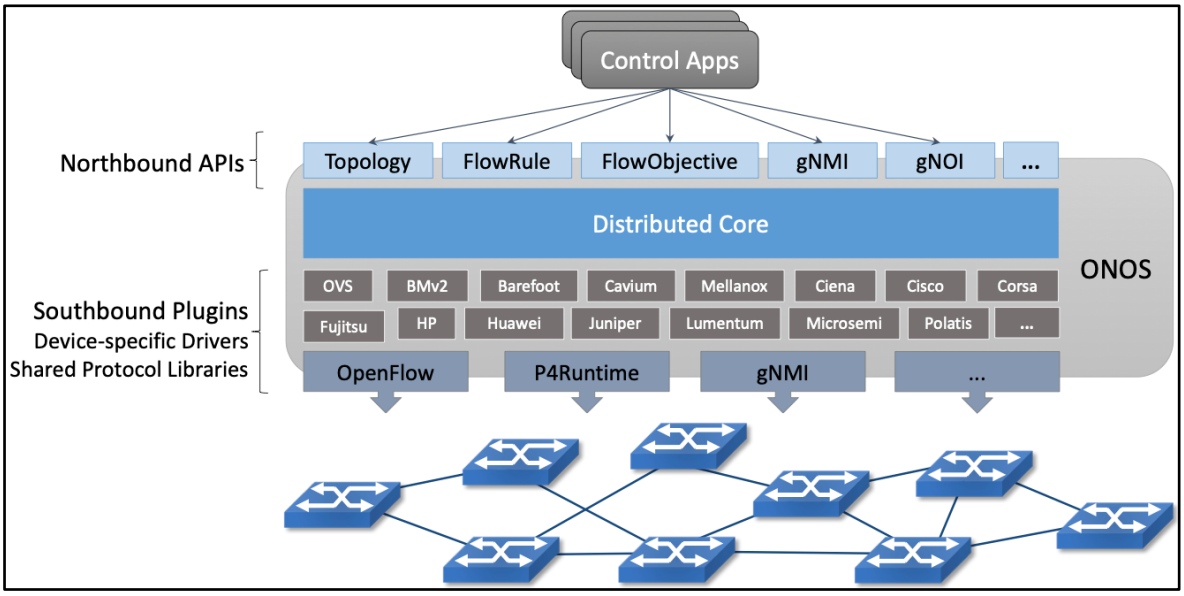


Figura 6. Arquitectura SDN con la integración del controlador ONOS [58].

Casos de uso y adopción de ONOS:

ONOS ha sido utilizado en múltiples iniciativas de investigación e implementaciones piloto. Un ejemplo representativo es CORD (*Central Office Re-architected as a Datacenter*), un proyecto impulsado por ON.Lab en colaboración con operadores como AT&T. CORD propone transformar las centrales telefónicas tradicionales en centros de datos abiertos, combinando SDN (con ONOS como controlador) y virtualización de funciones (NFV) para implementar de forma ágil funciones de acceso y servicios de abonado [10]. En particular, en CORD se virtualiza la función de terminación óptica (vOLT) en redes GPON, controlada centralmente por ONOS. Este caso de uso demuestra la capacidad de ONOS para manejar redes de acceso de gran escala, integrando elementos ópticos y proporcionando a la operadora una plataforma programable sobre la cual desplegar nuevos servicios de banda ancha. Otros casos de uso de ONOS reportados incluyen redes metropolitanas ópticas disgregadas, entornos multi-capa (paquete/óptico), redes 5G en la *fronthaul/backhaul*, y escenarios de *multi-tenant networking*, en los cuales ONOS actúa como pieza central del control definido por *software*.

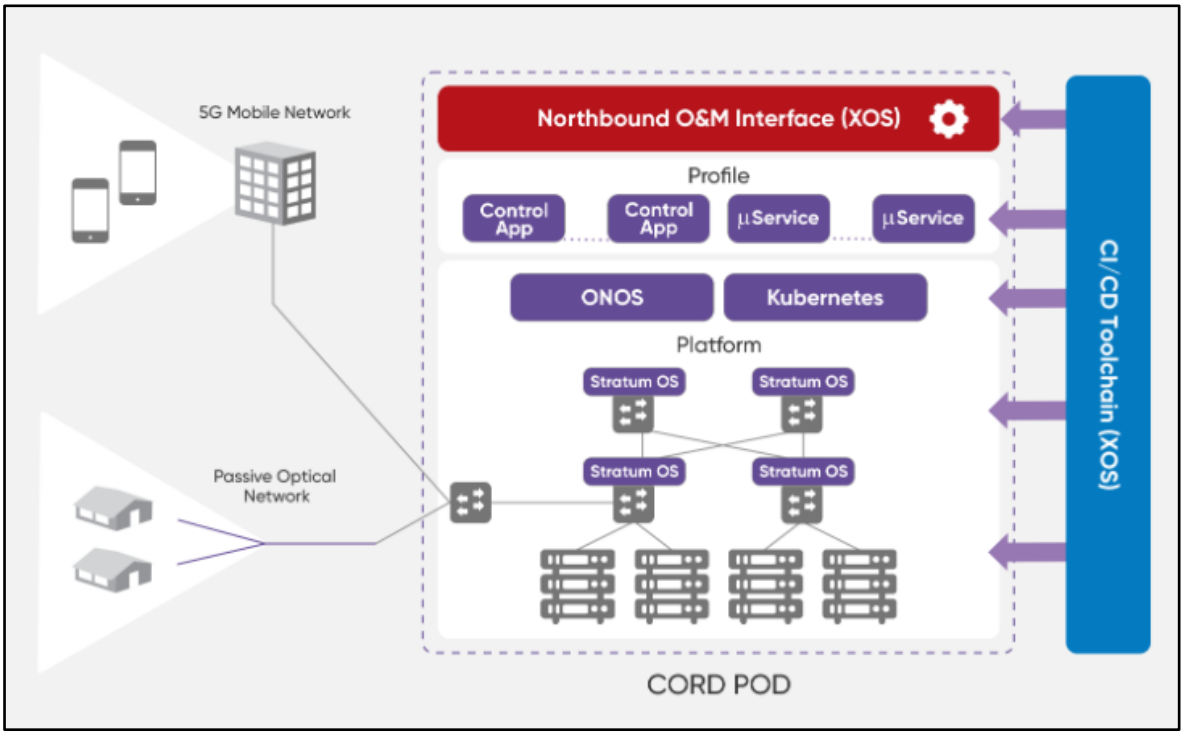


Figura 7. Arquitectura del proyecto CORD [55].

2.2.4. Comparativa con otros controladores SDN

La elección de ONOS frente a otros controladores populares se sustenta en varias consideraciones técnicas. En la Tabla 1 se presenta una comparación resumida entre ONOS y tres plataformas *open-source* destacadas: *OpenDaylight* (ODL), *Ryu y Floodlight*. Se contrastan aspectos como el origen del proyecto, año de lanzamiento, lenguaje de implementación, arquitectura y enfoque principal.

Tabla 1. comparación de controladores SDN.

Controlador	Proyecto/Desarrollador	Lanzamiento	Lenguaje	Arquitectura	Enfoque principal
ONOS	ON.Lab / ONF (Linux Foundation)	2014	Java	Distribuido (*cluster* lógico)	Operadores (*carrier-grade*), redes WAN y ópticas
OpenDaylight	Linux Foundation (Comunidad)	2013	Java	Modular (OSGi), HA opcional	Centros de datos, integración legada (multi-protocolo)

Ryu	NTT (comunidad *open-source*)	2012	Python	Centralizado (monolítico)	Investigación y pequeñas redes (prototipado rápido)
Floodlight	Big Switch Networks (ahora Intel)	2012	Java	Centralizado (modular básico)	Redes empresariales, OpenFlow puro (OF 1.0-1.3)

Como se observa, tanto *ONOS* como *OpenDaylight* están escritos en Java y cuentan con arquitecturas modulares extensibles. Sin embargo, *ONOS* se distingue por su diseño nativamente distribuido y orientado a entornos de operador, mientras que *OpenDaylight* nació como una plataforma más genérica enfocada inicialmente en redes de centro de datos y en facilitar la transición desde redes *legacy* hacia SDN [10]. De hecho, *OpenDaylight* opera bajo una filosofía de *framework*: proporciona un conjunto muy amplio de APIs, protocolos y *plugins* para adaptarse a múltiples casos (soporta OpenFlow, BGP-LS, NETCONF/YANG, PCEP, etc.), pero esto conlleva mayor complejidad y consumo de recursos. *ONOS*, en cambio, prioriza el desempeño y la consistencia en entornos de gran escala, manteniendo un núcleo más compacto optimizado para operación distribuida.

Por otro lado, *Ryu* y *Floodlight* son controladores más ligeros. *Ryu* (desarrollado en *Python*) es valorado por su simplicidad y facilidad de uso en entornos de investigación o despliegues de menor tamaño; diversas evaluaciones señalan que Ryu ofrece un conjunto suficiente de funcionalidades para escenarios a pequeña escala, mientras que controladores como ODL y ONOS proveen la mayor riqueza funcional para implementaciones de gran envergadura [12]. *Ryu* no soporta *clustering* ni alta disponibilidad de forma nativa, operando típicamente como un único proceso que controla la red. *Floodlight*, por su parte, es una evolución del controlador inicial de Stanford (el proyecto Beacon) mantenida por Big Switch. Es una plataforma Java modular, pero concebida para ejecutarse de forma centralizada, enfocada principalmente en controlar *switches OpenFlow*. *Floodlight* fue de los primeros controladores SDN ampliamente disponibles y se empleó mucho en laboratorios y pruebas de concepto, pero carece de las capacidades de escalado horizontal que ofrecen *ONOS* u ODL.

En términos de prestaciones, estudios recientes han comparado el desempeño de estos controladores bajo distintas métricas (latencia de control, *throughput* de flujo, tiempo de convergencia, etc.). De manera interesante, se ha observado que *Ryu* puede lograr latencias muy bajas en topologías simples gracias a su ligereza, pero *ONOS* y ODL

sobresalen en robustez y riqueza de características cuando se enfrentan a escenarios complejos [12]. En particular, *ONOS* ha sido identificado como uno de los controladores más seguros**,** fiables**,** robustos y escalables en evaluaciones globales [13]. Entre las razones se encuentran su arquitectura altamente flexible y personalizable (es más sencillo implementar extensiones o aplicaciones a medida en *ONOS* que en otros controladores) y su capacidad de escalar para manejar un gran número de dispositivos y flujos, lo que lo hace idóneo para redes de operadores [13]. Asimismo, *ONOS* ofrece soporte para múltiples tecnologías de red (no se limita a *OpenFlow*) y cuenta con una comunidad activa que impulsa mejoras frecuentes del *software* [13]. Estas ventajas inclinaron la balanza a favor de *ONOS* para nuestra implementación, ya que nuestra red GPON virtualizada experimental requiere un control central eficiente, con soporte multi-protocolo y potencial de crecimiento en escala, propiedades en las que ONOS destaca frente a alternativas como ODL, *Ryu* o *Floodlight*.

2.3. El protocolo *OpenFlow*

2.3.1 Definición

Para que un controlador SDN pueda gobernar los dispositivos de red en el plano de datos, es necesario un protocolo de comunicación estándar entre el plano de control centralizado y cada conmutador/enrutador. OpenFlow surgió precisamente para satisfacer esta necesidad, convirtiéndose en el primer estándar ampliamente adoptado como interfaz abierta entre el controlador SDN y el plano de reenvío [14]. Propuesto inicialmente por investigadores de Stanford y la Universidad de California en Berkeley en 2008, OpenFlow fue posteriormente promovido y estandarizado por la Open Networking Foundation (ONF) a partir de 2011. De acuerdo con la ONF, OpenFlow define una interfaz de control que proporciona acceso directo y manipulación de bajo nivel del plano de reenvío de dispositivos de red (*switches* y *routers*), sean físicos o virtuales [14]. Esto rompe con el carácter cerrado y propietario de los equipos tradicionales, permitiendo que un *software* de control externo configure las tablas de flujo y dicte el comportamiento de los *switches* de forma consistente entre múltiples fabricantes.

2.3.2 Funcionamiento básico

En una red SDN basada en *OpenFlow*, los *switches* (conocidos como *OpenFlow switches*) mantienen una o varias tablas de flujo que contienen reglas de coincidencia (*match rules*) y acciones asociadas. Cada regla especifica un conjunto de campos de cabecera (por ejemplo, dirección MAC, IP, puerto TCP, etc.) y la acción a aplicar a los paquetes que

coincidan (por ejemplo, reenviar por cierto puerto, modificar un campo, o descartar). El controlador SDN, a través del protocolo *OpenFlow*, puede añadir**,** modificar o eliminar entradas en estas tablas de flujo de los *switches* de manera dinámica [14]. Cuando llega un paquete a un conmutador *OpenFlow*, este intenta hacer coincidir los campos del paquete con alguna entrada en su tabla de flujo: si encuentra una coincidencia, aplica la acción definida (p. ej. reenviarlo por un puerto específico) localmente y a velocidad de línea; si no existe una regla para ese flujo, el paquete (o al menos su cabecera) es enviado al controlador mediante un mensaje *Packet-In*. El controlador, al recibir este evento, puede decidir cómo manejar ese tráfico: por ejemplo, instalar una nueva regla de flujo en el *switch* para que trate ese tipo de paquetes en adelante, o reenviar el paquete por otro camino. De este modo, las decisiones de encaminamiento se toman de forma centralizada en el controlador, pero una vez establecidas las reglas, el tráfico de datos fluye directamente por el *hardware* con alto rendimiento. Este equilibrio permite combinar la flexibilidad del control centralizado con la eficiencia del reenvío distribuido en los dispositivos [14].

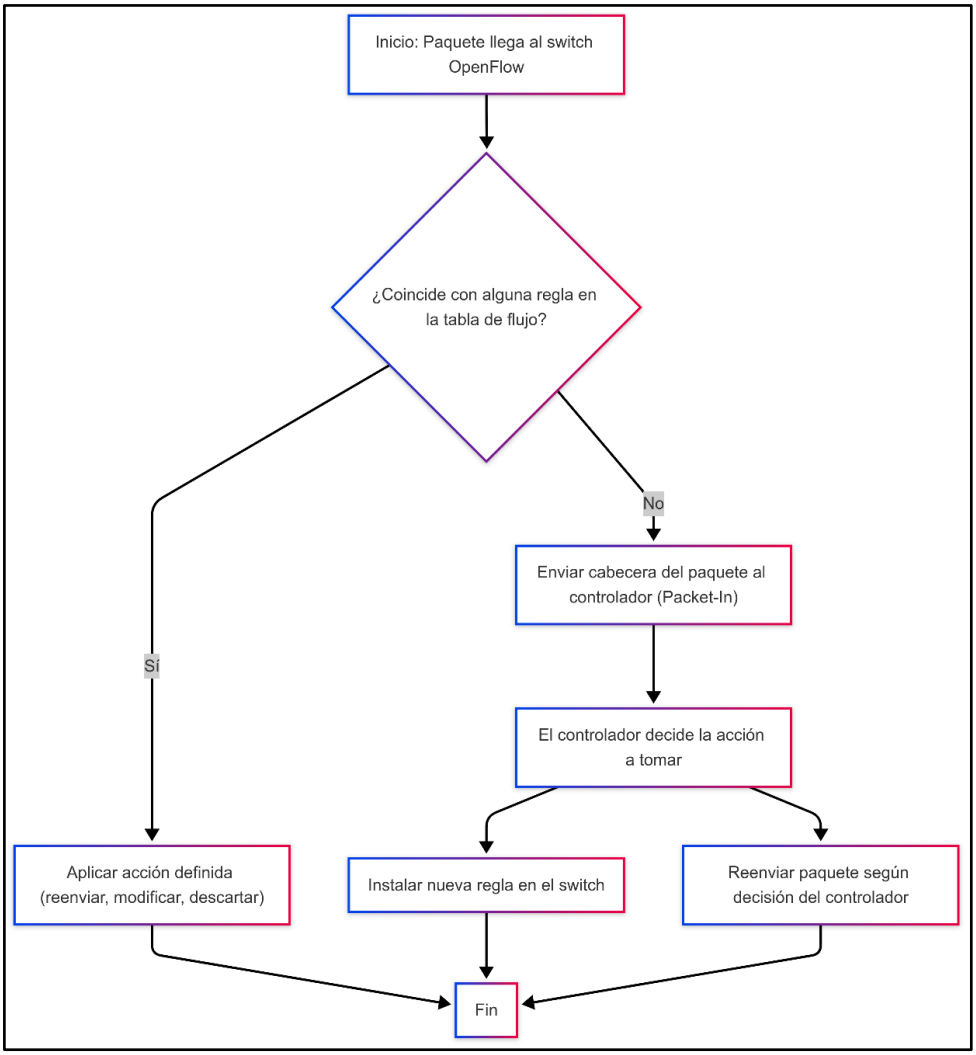


Figura 8. Diagrama de flujo del procesamiento de un paquete en un switch OpenFlow. Elaboración propia.

OpenFlow define además un canal de comunicación seguro entre cada *switch* y el controlador, conocido como canal de control o *secure channel*. Usualmente se establece sobre TCP utilizando TLS para cifrado; el controlador por defecto escucha conexiones de *switches* en el puerto TCP 6653 (en versiones anteriores se usaba el 6633) [14]. A través de este canal, el controlador envía mensajes de administración de flujo (*Flow Mod*) para manipular las tablas de los *switches*, y recibe de los *switches* eventos y estadísticas (p. ej., mensajes *Packet-In* para paquetes nuevos, *Port Status* cuando cambia el estado de un puerto, o *Flow Removed* cuando expira una regla).

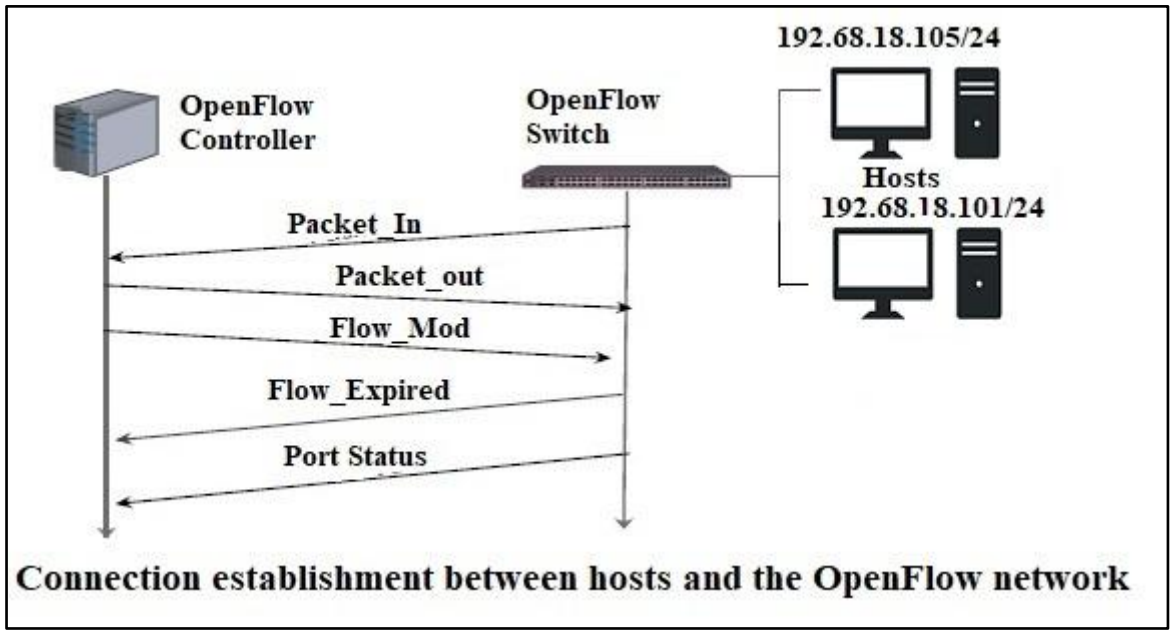


Figura 9. Secuencia de mensajes de control para el establecimiento de conexión entre hosts y una red OpenFlow [59].

Entre los componentes clave de una arquitectura SDN basada en *OpenFlow* se incluyen entonces: (1) el controlador *OpenFlow*, *software* central que orquesta la red; (2) los *switches OpenFlow*, que soportan el protocolo y ejecutan las reglas de flujo; y (3) el canal de control seguro entre ellos, que garantiza la comunicación fiable y aislada del tráfico de datos.

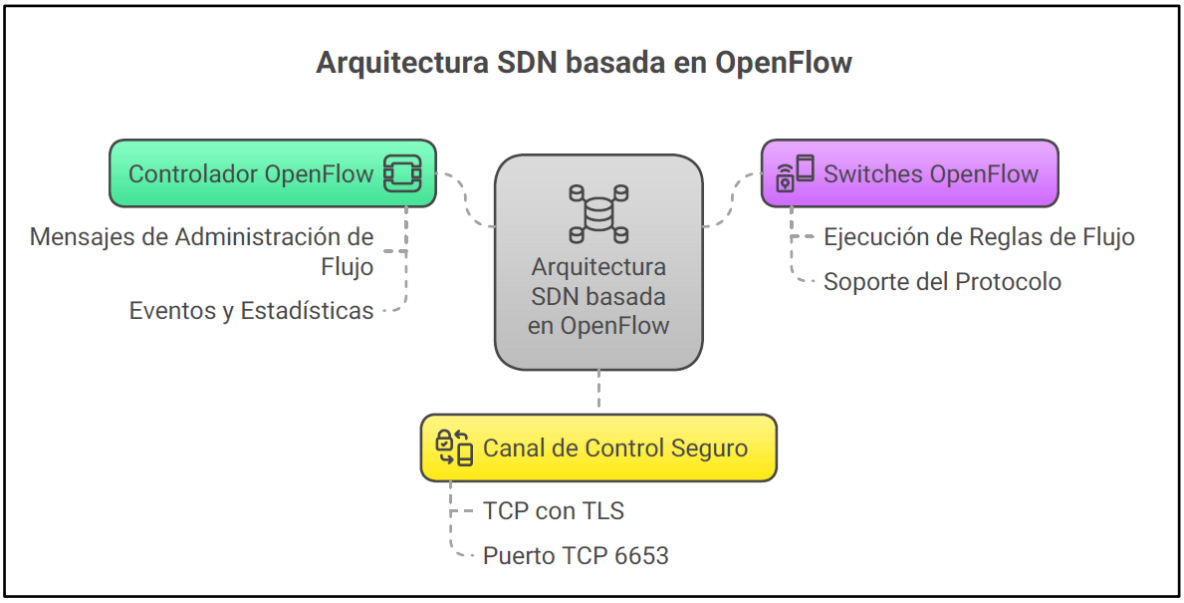


Figura 10. Componentes principales de una arquitectura SDN basada en OpenFlow. Elaboración propia.

2.3.3 Versiones de *OpenFlow*

Desde su introducción, el estándar *OpenFlow* ha evolucionado incorporando nuevas capacidades. La versión 1.0 (publicada en 2009) fue la primera en ser implementada ampliamente, permitiendo una única tabla de flujo y un conjunto básico de acciones. Versiones posteriores añadieron funcionalidades importantes: *OpenFlow* 1.1 introdujo múltiples tablas y grupos; la 1.2 y 1.3 (2012) agregaron mejoras como soporte IPv6, medidores, y sobre todo *OpenFlow* 1.3 extendió significativamente las capacidades (fue adoptada como base en muchos controladores SDN actuales, incluyendo ONOS y ODL). *OpenFlow* 1.4 y 1.5 añadieron características adicionales (p. ej. control de tablas múltiples más flexible, nuevas acciones y tipos de coincidencia, etc.), alcanzando la versión 1.5.1 en 2015 como la última especificación pública [14]. (Existe un *OpenFlow* 1.6 de acceso restringido dentro de ONF, orientado a experimentación). En general, *OpenFlow* 1.3 se considera una versión relevante y estable, soportada por la mayoría de los controladores y *switches OpenFlow* actuales, mientras que versiones anteriores como la 1.0 han caído en

desuso y las posteriores se usan principalmente en entornos específicos que requieran sus extensiones.

OpenFlow ha sido fundamental en el auge de SDN, ya que proporcionó por primera vez un lenguaje común para que controladores y dispositivos de distintos fabricantes interactúen en el plano de control. Si bien con el tiempo han surgido otros protocolos e interfaces sur (por ejemplo, NETCONF/YANG para configuración, gRPC/gNMI, P4Runtime para conmutadores programables, etc.), *OpenFlow* sigue siendo un pilar histórico y conceptual de SDN. ONF lo reconoce como "la primera interfaz estándar entre el plano de control y el de reenvío en una arquitectura SDN" [14], y su presencia es amplia tanto en entornos de investigación como en algunas implementaciones productivas. En este trabajo, la red GPON virtualizada empleará *OpenFlow* (versión 1.3) como protocolo de control entre *ONOS* y los elementos de la red de acceso, dada la compatibilidad de *ONOS* y del equipamiento virtual con dicho estándar.

Tabla 2. Comparación entre las versiones del protocolo OpenFlow.

Versión	Año de publicación	Caracteristicas principales	Notas de adopción
1.0	Diciembre 2009	Una única tabla de flujo; conjunto básico de acciones (forward, drop, set VLAN, modificación de cabeceras simples). Sin soporte IPv6, sin grupos ni medidores.	Primera versión adoptada ampliamente en entornos de investigación; actualmente obsoleta.
1.1	Febrero 2011	Introducción de múltiples tablas de flujo y la tabla de grupos (all, select, indirect, fast-failover); pipeline más flexible; nuevas acciones de copia TTL y manipulación MPLS.	Poca adopción en producción; muchos fabricantes avanzaron directamente a 1.3.
1.2	Febrero 2012	Compatibilidad completa con IPv6 gracias al formato OXM extensible; nuevos campos de coincidencia y acciones asociadas.	Uso limitado; sirvió de transición tecnológica hacia OpenFlow 1.3.

1.3	Junio 2012	Incorporación de la tabla de medidores para QoS, set-field para modificar cabeceras, mejoras de IPv6, cookie de 64 bits, tabla-miss y contadores ampliados.	Versión de referencia; ampliamente soportada por controladores (ONOS, ODL) y switches (OVS, white-box).
1.4	Octubre 2013	Mensajes "bundle" para operaciones atómicas, monitorización asíncrona y extensiones de notificación, mejoras en contadores y nuevas acciones/oxm.	Soporte experimental en algunos controladores; adopción limitada.
1.5.1	Marzo 2015	Tablas de egreso, acción copy-field, push/pop PBB, nuevos motivos de packet-in, extensiones de OAM y campos OXM adicionales.	Última especificación pública (1.6 es restringida); usada principalmente en investigación y laboratorios.

2.4 NFV

2.4.1 Definición

En paralelo al avance de SDN, surgió en el ámbito de las telecomunicaciones el concepto de virtualización de funciones de red (NFV, *Network Functions Virtualization*). La idea central de NFV es desligar las funciones de red (cortafuegos, encaminadores, balanceadores de carga, sistemas de autenticación, etc.) del equipamiento especializado propietario, implementándolas en *software* que se ejecuta sobre *hardware* de propósito general (servidores estándar, plataformas *cloud*, etc) [15]. Este enfoque, propuesto formalmente por un grupo de operadores a través de ETSI en 2012, busca trasladar a las redes los beneficios comprobados de la virtualización en TI: mayor flexibilidad, reducción de costes y agilidad en el despliegue de nuevos servicios.

2.4.2 Arquitectura NFV (ETSI)

El marco de referencia definido por el *European Telecommunications Standards Institute* (ETSI) para NFV consta de tres componentes principales:

- ***Infraestructura de Virtualización de Funciones de Red (NFVI):*** comprende los recursos de *hardware* y *software* sobre los que se ejecutan las funciones virtualizadas. Incluye los servidores, almacenamiento y equipos de red físicos, así como la capa de virtualización (hipervisores, contenedores) que abstrae esos recursos físicos en recursos virtuales. La NFVI expone así CPUs virtuales, redes y almacenamiento virtuales a las funciones, permitiendo su ubicación y escalado flexible [16]. Puede estar distribuida en múltiples centros de datos o ubicaciones de la red.

- ***Funciones de Red Virtualizadas (VNFs):*** son las instancias *software* de las funciones de red tradicionales ejecutándose sobre la NFVI. Cada VNF realiza una función específica (por ejemplo, un *vFirewall*, un *vRouter* o un *vBNG* - BroadBand Network Gateway virtual). Las VNFs pueden descomponerse en componentes más pequeños (VCUs, VDUs) y escalarse horizontal o verticalmente según la carga. Múltiples VNFs pueden desplegarse en la misma infraestructura compartida, aisladas unas de otras mediante la capa de virtualización.

- ***Gestión y Orquestación (MANO):*** es el conjunto de sistemas encargados de orquestar el despliegue y ciclo de vida de las VNFs, así como de gestionar los recursos de la NFVI. Dentro de MANO se definen entidades como el NFV *Orchestrator* (NFVO), responsable de coordinar la implementación de servicios de red compuestos por VNFs; el VNF Manager (VNFM), encargado de la configuración y control específico de cada VNF (por ejemplo, escalar instancias de una función); y el Gestor de Infraestructura Virtual (VIM), que administra los recursos virtuales en la NFVI (por ejemplo, un VIM típico es OpenStack para controlar máquinas virtuales, redes virtuales y almacenamiento). A través de MANO, se automatizan tareas como la instalación de una nueva función virtual, su escalamiento bajo demanda, la monitorización de desempeño, y la conexión entre múltiples VNFs para conformar *service chains* (cadenas de servicio). Esta capa es crítica para materializar la promesa de NFV de gestión ágil y dinámica.

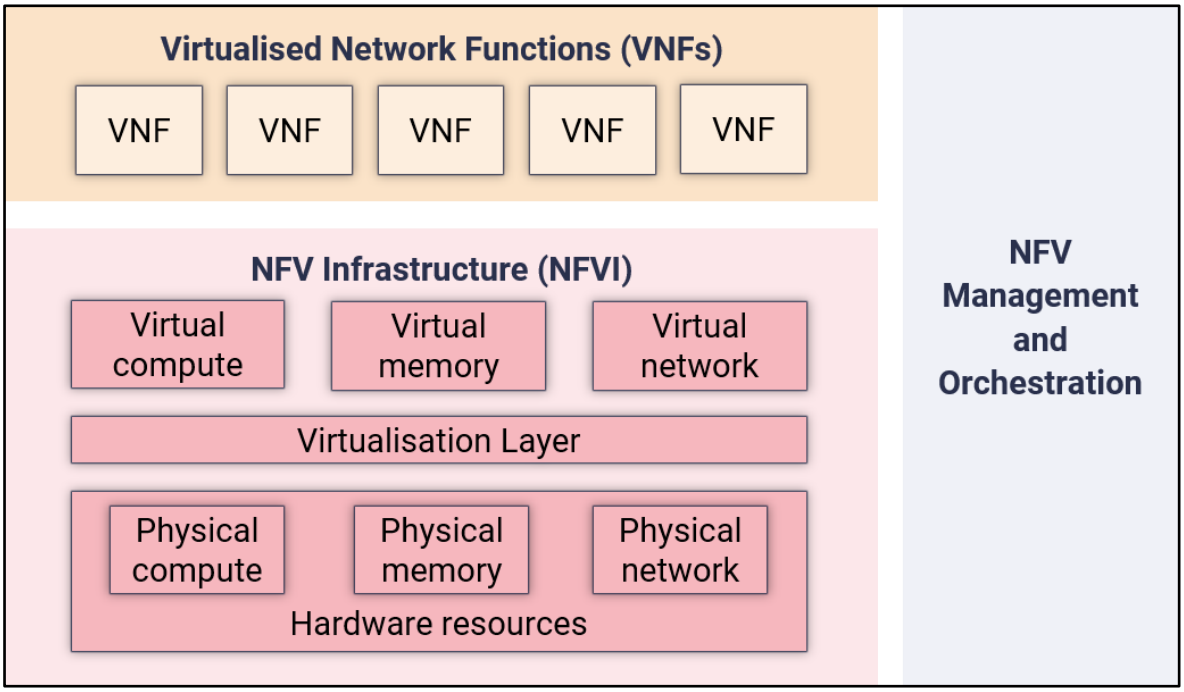


Figura 11. Arquitectura NFV según ETSI: VNFs, NFVI y Gestión-Orquestación (MANO) [60].

En esencia, NFV traslada funciones que antes requerían *appliances* dedicados (cajas físicas con *software* embebido) a entornos virtualizados estándar. Por ejemplo, en vez de instalar un nuevo *hardware* propietario para implementar un *firewall* en una red operadora, NFV permite desplegar un *firewall* virtual como *software* en un servidor x86 común, posiblemente conviviendo con otras funciones en la misma máquina física. Esto conlleva ahorros de *capital expenditures* (CAPEX) y de *operational expenditures* (OPEX): se reduce la inversión en equipos especializados (reemplazados por servidores comerciales de alto volumen) y se aprovechan economías de escala en centros de datos; además, la provisión de nuevas funciones es más rápida y automatizable (reduciendo labores manuales de instalación y configuración) [17]. La infraestructura se vuelve más dinámica, pudiendo escalar funciones hacia arriba o abajo mediante *software* según la demanda (por ejemplo, instanciar más VNFs de balanceo en horas pico y liberarlas luego), en contraste con la rigidez de las redes tradicionales donde el dimensionamiento depende de *hardware* fijo. Estudios recientes enfatizan que NFV habilita una infraestructura de red flexible y programable, al sustituir funciones en *hardware* por *software* sobre servidores estándar, a la vez que mantiene especificaciones de nivel *carrier* (alta disponibilidad, confiabilidad) mediante la elasticidad del *software* [15].

2.4.3 Ventajas y beneficios

Entre los principales aportes de NFV se cuentan:

***Reducción de costos y eficiencia operativa**:* Al usar *hardware* genérico en lugar de dispositivos propietarios costosos, los operadores pueden reducir gastos de capital. Asimismo, la automatización mediante orquestación *software* disminuye los costos operativos al necesitar menos intervención manual [17]. Por ejemplo, consolidar varias funciones virtuales en un mismo servidor incrementa la utilización de recursos y reduce la cantidad de equipos físicos a mantener.

***Agilidad en el despliegue de servicios**:* NFV acelera el tiempo de salida al mercado de nuevos servicios de red. Donde antes desplegar una nueva función podía tomar semanas o meses (adquisición de *hardware*, instalación en sitio, configuración), con NFV es posible instanciar VNFs en minutos mediante *software*. Esta agilidad permite a los proveedores responder más rápido a demandas cambiantes, desplegar funciones temporales para pruebas, o actualizar *software* frecuentemente para introducir mejoras o parches.

***Escalabilidad y elasticidad**:* Las VNFs pueden escalar horizontalmente (lanzando instancias adicionales en momentos de alta demanda) y luego liberarlas cuando baja el tráfico, algo difícil de lograr con equipos físicos fijos. Esta elasticidad asegura que la capacidad de la red se ajuste dinámicamente al requerimiento, mejorando la eficiencia. Además, NFV facilita adaptar la infraestructura a distintos tamaños de red sin rediseñar *hardware* específico, simplemente asignando más recursos de la NFVI a las funciones críticas.

***Innovación y ecosistema abierto**:* Al basarse en *software*, NFV abre la puerta a que nuevas funciones sean desarrolladas por terceros o la comunidad *open-source*, fomentando la innovación más allá de los ciclos tradicionales de *hardware*. También reduce la dependencia de un único fabricante: en una plataforma NFV, distintas VNFs, incluso de proveedores distintos, pueden convivir sobre la misma infraestructura estandarizada siempre que cumplan las especificaciones de ETSI, aumentando la interoperabilidad.

Capítulo 3. Fundamentos teóricos GPON

3.1. Introducción

Este capítulo tiene como objetivo exponer los fundamentos básicos de la tecnología GPON (*Gigabit-capable Passive Optical Network*), empleada en redes de acceso de fibra óptica. Se describen los componentes fundamentales que componen este tipo de red, concretamente la OLT (*Optical Line Terminal*) y la ONT (*Optical Network Terminal*), así como el funcionamiento detallado de las tramas que gestionan la comunicación en los canales ascendentes y descendentes.

La estructura de este capítulo es la siguiente:

- En primer lugar, se describirá el equipo central OLT, explicando sus características técnicas, funcionalidades y métodos de gestión empleados en este proyecto.
- A continuación, se abordará la descripción detallada del dispositivo ONT, detallando sus características técnicas y funciones específicas en el entorno del usuario final.
- Finalmente, se explicará en profundidad la estructura y funcionalidad de las tramas empleadas en la comunicación GPON, tanto en el sentido ascendente como en el descendente.

3.2. *Optical Line Terminal* (OLT)

3.2.1 Descripción

La OLT es el elemento central de la red GPON ubicado en la cabecera o central del operador. Su función es concentrar todo el tráfico ascendente proveniente de las ONT y multiplexar el tráfico descendente hacia ellas, a la vez que se encarga de la gestión y control de la red de acceso.

En esencia, la OLT actúa como puerta de enlace entre la red óptica pasiva y la red de agregación del operador, proporcionando interfaces hacia la red troncal (habitualmente puertos Ethernet o similares) y controlando el reparto del ancho de banda en la PON. Cada puerto GPON de la OLT puede atender decenas de ONT's en paralelo; en el caso de la plataforma empleada en esta red experimental, un solo equipo OLT con cuatro puertos

PON es capaz de servir hasta 256 abonados (64 ONT's por puerto) compartiendo el mismo enlace óptico [18].

3.2.2. Características de la *SmartOLT Serie 240*

En la red experimental del TFG se utiliza una OLT modelo *SmartOLT Serie 240* del fabricante Telnet. Este equipo implementa todas las funcionalidades del estándar GPON G.984 y es apto para despliegues FTTx (*Fiber To The x*) en distintas modalidades FTTH (*Fiber To The Home*), FTTB (*Fiber To The Building*), FTTC (*Fiber To The Curb*), entre otros [18].

Especificaciones técnicas principales:

- Soporte hasta 64 ONT's por puerto PON.
- Total de 512 Alloc-ID (*Allocation Identifier*) disponibles.
- Capacidad para servir hasta 256 hogares con un solo dispositivo.
- 4 puertos Gigabit Ethernet utilizados como interfaces PON (mediante módulos SFP GPON).
- Capacidad de 1 Gbps simétrico por fibra [18].

Funcionalidades de capa 2:

La OLT admite funciones de puenteado 802.1D, incluyendo *snooping* y filtrado IGMP (*Internet Group Management Protocol*) para optimizar el tráfico *multicast/broadcast*. También incorpora soporte de VLAN (*Virtual Local Area Network*) IEEE 802.1Q (VLAN simple) y QinQ (Q-in-Q) IEEE 802.1ad (doble encapsulado) para la segmentación de servicios.

Adicionalmente, la *SmartOLT 240* gestiona tramas con prioridad 802.1p y puede transportar o modificar etiquetas VLAN (*Virtual Local Area Network*) en el tráfico de usuario, lo que la hace flexible para integrarse en diversas configuraciones de red [18].

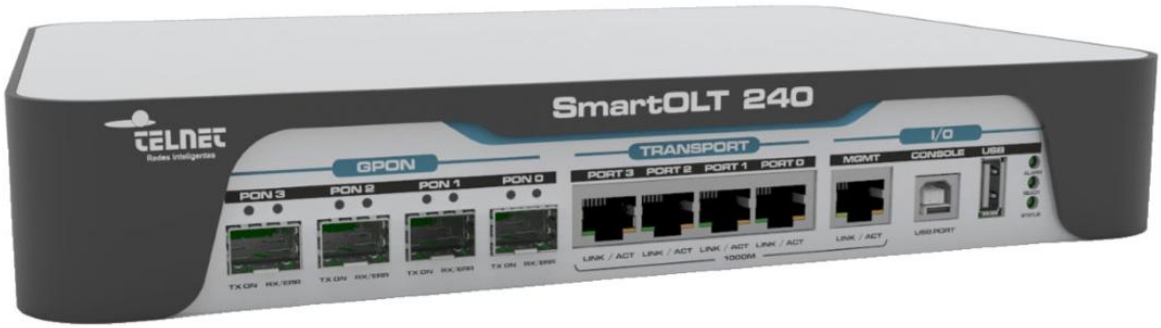


Figura 12. OLT GPON SmartOLT 240 de Telnet RI [61].

3.2.3. Gestión mediante TGMS

La *SmartOLT Serie 240* cuenta con un sistema de gestión web integrado denominado TGMS (*Telnet GPON Management System*), orientado a simplificar la configuración y administración del equipo.

A través de una interfaz gráfica accesible vía navegador, TGMS abstrae al operador de la complejidad del protocolo GPON y de los comandos de bajo nivel, automatizando tareas de provisión de ONT, asignación de perfiles de servicio, monitoreo de la red, entre otras funciones.

De este modo se reduce la curva de aprendizaje en comparación con OLT's tradicionales que requieren configuraciones manuales extensas, disminuyendo el riesgo de errores de configuración y agilizando el despliegue. El TGMS tanto la configuración inicial de la OLT como la gestión en operación, integrándose también con protocolos de gestión estándar para facilitar la supervisión remota [18].

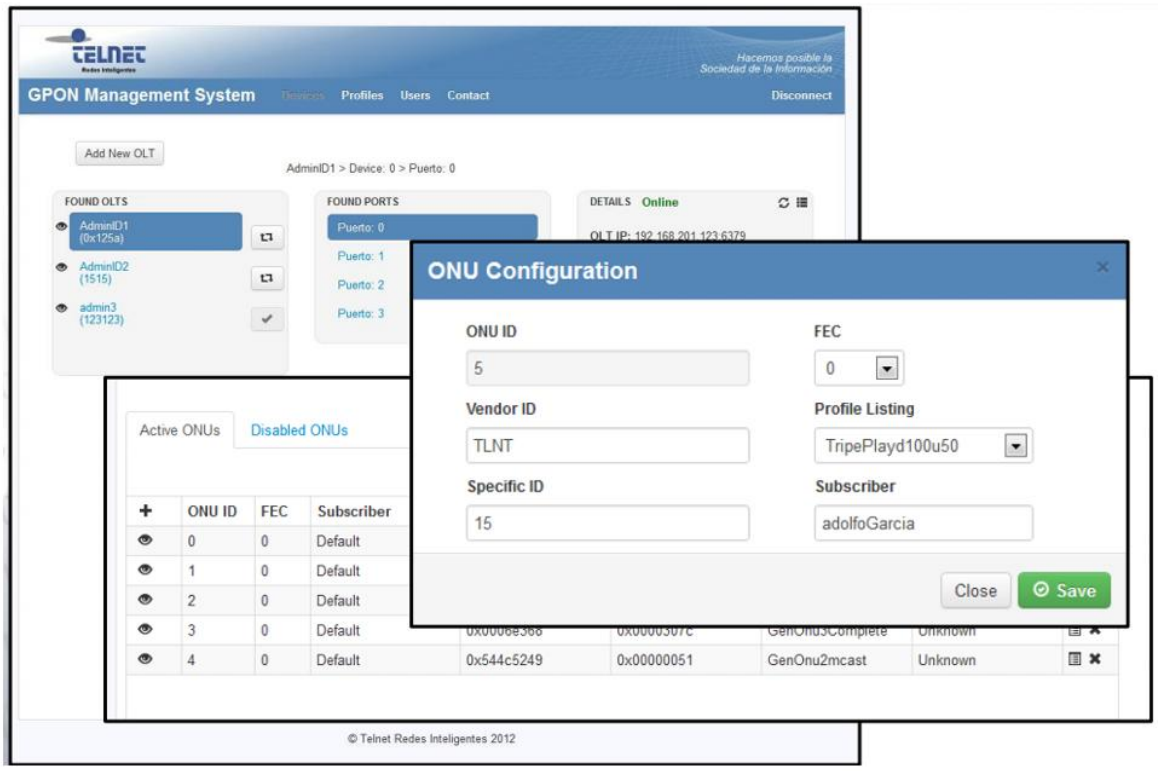


Figura 13. Sistema de gestión GPON TGMS de Telnet RI [62].

3.3. *Optical Network Terminal* (ONT)

3.3.1 Descripción

La ONT, también conocida como ONU (*Optical Network Unit*), es el dispositivo terminal del lado del abonado que convierte la señal óptica de la fibra GPON en interfaces de usuario (datos, voz, WiFi, etc.). En la red implementada se emplea el modelo de ONT Telnet *WaveAccess 4520* [19], un equipo GPON residencial que cumple con las especificaciones estándar ITU-T (*International Telecommunication Union Telecommunication Standardization Sector*) G.984.x (GPON) [20] y G.988 (OMCI) [21], así como con el perfil de interoperabilidad Broadband Forum TR-156 [22].

Características de seguridad y protocolo:

- Soporte del proceso de activación (*auto-discovery*) mediante registro del número de serie y contraseña según G.984.3 [23].
- Cifrado AES-128 (*Advanced Encryption Standard*) para proteger el flujo descendente [24].

- Corrección de errores FEC (*Forward Error Correction*) tanto en *downstream* como *upstream* [19].
- Mecanismos de detección de ONT maliciosas (*rogue* ONTs) para asegurar la integridad de la red PON [25].

Figura 14. ONT-L3 doméstica GPON de Telnet RI – vista superior [63].

3.3.2 Interfaces físicas

Puerto óptico:

El *WaveAccess 4520* dispone de un puerto óptico GPON con conector SC/APC (*Subscriber Connector/Angled Physical Contact*) de fibra monomodo, utilizando óptica de clase B+ capaz de alcanzar distancias de enlace de hasta 20 km. Opera en las longitudes de onda estándar de GPON (≈1310 nm en *uplink* y 1490 nm en *downlink*, con filtrado de 1550 nm para servicios de vídeo RF) [26].

Interfaces de red cableada:

La ONT incluye 4 puertos *Ethernet Gigabit* (10/100/1000 Base-T RJ-45) para brindar conectividad LAN (*Local Area Network)* cableada, con soporte de [19]:

- VLAN (*Virtual Local Area Network*) por puerto.
- Traducción y apilado de VLAN (Q-in-Q).

- Marcado de prioridad 802.1p en el tráfico.
- *Snooping* IGMP (*Internet Group Management Protocol*) (compatible con IGMPv1/v2/v3) y filtrado de multidifusión.

Interfaces de voz:

Para servicios de telefonía, el equipo ofrece 2 puertos POTS (*Plain Old Telephone Service*) (RJ-11) que permiten conectar teléfonos analógicos. Estos puertos pueden configurarse en modo *single* o *dual* (una o dos líneas) y soportan múltiples *códecs* de voz (p. ej., G.711 A/μ-law, G.729a, G.722) [19].

Figura 15. ONT-L3 doméstica GPON de Telnet RI – vista posterior [63].

3.3.3 Funcionalidades de ONT-*ROUTER*

La ONT actúa como *router* residencial completo, soportando:

- Funcionalidad de cliente PPPoE (*Point-to-Point Protocol over Ethernet*) para autenticación en redes de banda ancha [27].
- NAT/NAPT (*Network Address Translation/Network Address Port Translation*) para compartir la conexión entre múltiples dispositivos [28].
- Servidor DHCP (*Dynamic Host Configuration Protocol*) (además de cliente DHCP en la WAN) para asignación automática de direcciones IP (*Internet Protocol*) [29].

Conectividad inalámbrica:

En el apartado inalámbrico, incorpora conectividad Wi-Fi de doble banda:

- IEEE 802.11b/g/n (2,4 GHz, MIMO 2×2).
- IEEE 802.11ac (5 GHz, MIMO 3×3).

- Antenas internas de alta ganancia.

Gracias a estas prestaciones, una sola ONT *WaveAccess 4520* puede proporcionar acceso de alta velocidad por cable e inalámbrico, y servicios de voz sobre IP (*Internet Protocol*), integrando en un solo equipo la funcionalidad de ONT y home *router*. Cabe destacar que este modelo ha sido verificado para ser interoperable con OLT's de los principales fabricantes del mercado, asegurando su compatibilidad en escenarios *multi-vendor* [19].

3.4. Descripción de las tramas GPON

El estándar GPON define un esquema de transmisión basado en tramas periódicas de duración fija, encargadas de transportar tanto la información de usuario (datos) como los mensajes de control necesarios para el funcionamiento de la red.

A nivel de la capa de convergencia GPON GTC (GPON *Transmission Convergence*), la OLT envía tramas descendentes regulares de 125 µs, mientras que las ONT transmiten en intervalos ascendentes sincronizados de igual duración, según la asignación otorgada por la OLT [30].

3.4.1 Trama descendente (*Downstream*)

En el enlace descendente, la OLT emite de forma continua una trama GTC (GPON *Transmission Convergence*) cada 125 µs, lo que corresponde a un tamaño fijo de 38.880 *bytes* a la velocidad de línea de 2,488 Gbps [30]. En la siguiente figura podemos ver la estructura de la trama en cuestión.

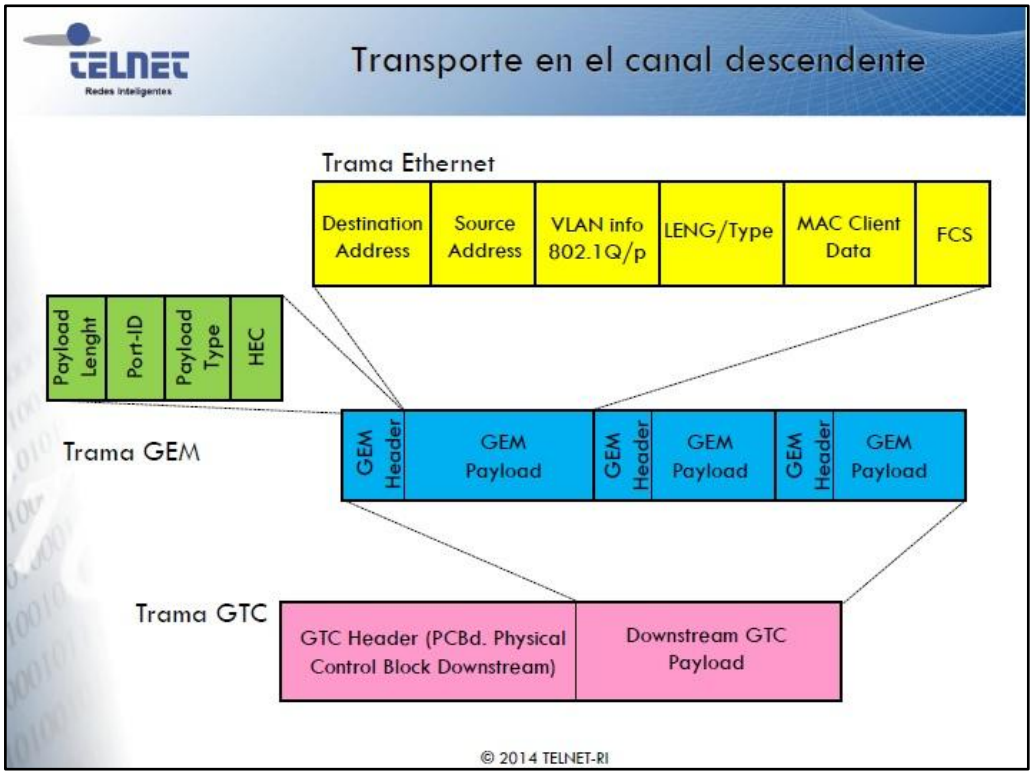


Figura 16. Estructura de trama en el canal descendente de la GPON [65].

En la figura superior observamos, que las tramas involucradas se componen de los siguientes campos:

Trama *ethernet*:

En la trama *Ethernet* encontramos la unidad clásica de nivel 2 que transporta los datos de usuario de manera transparente para la red GPON. Su función primordial es ofrecer un contenedor universal, reconocido por prácticamente todos los equipos de conmutación y encaminamiento, que incluya las direcciones MAC de origen y destino, los identificadores de VLAN (802.1Q/p) y el campo *Length/Type* que anuncia el protocolo de capa 3 o la longitud del campo de datos. El FCS (CRC-32) garantiza la detección de errores [32] sobre el enlace punto a punto lógico que se construye entre el OLT y cada ONU; así, la red óptica puede entregar tráfico IP, MPLS o cualquier otra carga encapsulada sin necesidad de modificar los equipos terminales del cliente.

Tabla 3. Campos de la trama Ethernet (canal descendente).

Orden	Campo	Tamaño (bytes)	Función principal
1	Destination Address (DA)	6	Dirección MAC *unicast*, *multicast* o *broadcast* de destino.
2	Source Address (SA)	6	Dirección MAC del emisor.
3	Etiqueta 802.1Q/p	4	Presente solo si la trama va etiquetada. Incluye TPID, PCP, DEI y VID.
4	Length / Type	2	Longitud del *payload* (≤1 500) o *EtherType* (≥1 536) que indica el protocolo de capa 3.
5	MAC Client Data (Payload)	46 – 1500	Datos de usuario; se rellena con *padding* si es menor que el mínimo.
6	Frame Check Sequence (FCS)	4	CRC-32 para detección de errores.

Trama GEM:

La trama GEM (GPON *Encapsulation Method*) actúa como capa de adaptación entre el mundo *Ethernet* y la infraestructura GPON. Mediante un encabezado compacto de 5 bytes formado por el *Payload Length Indicator*, el Port-ID, el *Payload Type Indicator* y un HEC de protección, GEM permite multiplexar flujos lógicos (servicios) pertenecientes a distintos T-CONT, fragmentar tramas grandes, etiquetar paquetes OAM o de mantenimiento y, en general, dotar de flexibilidad y control de calidad de servicio a la red [30] [31]. Su diseño ligero aporta todo lo necesario para distinguir y gestionar el tráfico sin añadir la complejidad de protocolos de nivel 3, manteniendo al mismo tiempo la transparencia completa para las tramas *Ethernet* que transporta como carga útil.

Tabla 4. Campos de la trama GEM (canal descendente).

Campo (Encabezado GEM)	Longitud	Descripción
Payload Length Indicator (PLI)	12 bits	Indica la longitud en bytes del payload (0 – 4095).
Port-ID	12 bits	Identificador lógico del puerto GEM; permite multiplexar servicios.
Payload Type Indicator (PTI)	3 bits	Clasifica el contenido y la fragmentación (usuario, OAM, fragmento, inactivo).
Reservado	0 – 1 bit	Reservado para alineación.
Header Error Control (HEC)	13 bits	CRC que protege los 27 bits anteriores.

Trama GTC:

En el sentido descendente de una red GPON, la trama GTC (GPON *Transmission Convergence*) constituye la unidad de transmisión periódica que el OLT difunde cada 125 µs. Su principal cometido es ofrecer un marco temporal común y continuo a todas las ONUs, integrando en un único bloque la referencia de sincronización que alinea los relojes ópticos, la información de operación y mantenimiento que sustenta la gestión remota, el control de errores que salvaguarda la integridad del canal y (la planificación del ancho de banda ascendente mediante el mapa de asignación (BWmap) [30] [32]. En la sección de carga útil, la GTC encapsula secuencialmente las tramas GEM que transportan los flujos de usuario, garantizando así que los datos lleguen a destino con la calidad de servicio prevista.

Tabla 5. Campos de la trama GTC (Canal descendente).

Campo	Longitud (bytes)	Propósito
Psync	4	Patrón de sincronización que delimita el inicio de la trama.
Ident	4	Contador de supertrama; interviene en cifrado y sincronismo.
PLOAMd	13	Canal OAM descendente para gestión y alarmas.
BIP	1	Bit-Interleaved Parity calculado sobre la trama para supervisión de errores.
Plend	4	Longitud del campo US BWmap (enviado dos veces para redundancia).
US BWmap	Variable	Mapa de asignación de ancho de banda ascendente: define los *time-slots* para cada T-CONT.

3.5.2 Descripción de las tramas: canal ascendente

En el sentido ascendente, el medio es compartido y las ONT transmiten en ráfagas temporizadas dentro de cada ciclo de 125 µs, según el esquema definido por el BWmap (*BandWidth Map*) previamente enviado por el OLT. La suma de todas las ráfagas de las ONT (*Optical Line Terminal*) conforma una trama ascendente GTC (GPON *Transmission Convergence)* de 125 µs. Para un enlace GPON (*Gigabit Passive Optical Network*) de 1,244 Gbps, cada trama ascendente tiene un tamaño agregado de 19.440 *bytes* distribuidos entre los múltiples *bursts* de las ONT (*Optical Line Terminal*) activas [30] [32].

En la figura sigueinte podemos observar la estructura de la trama en cuestión:

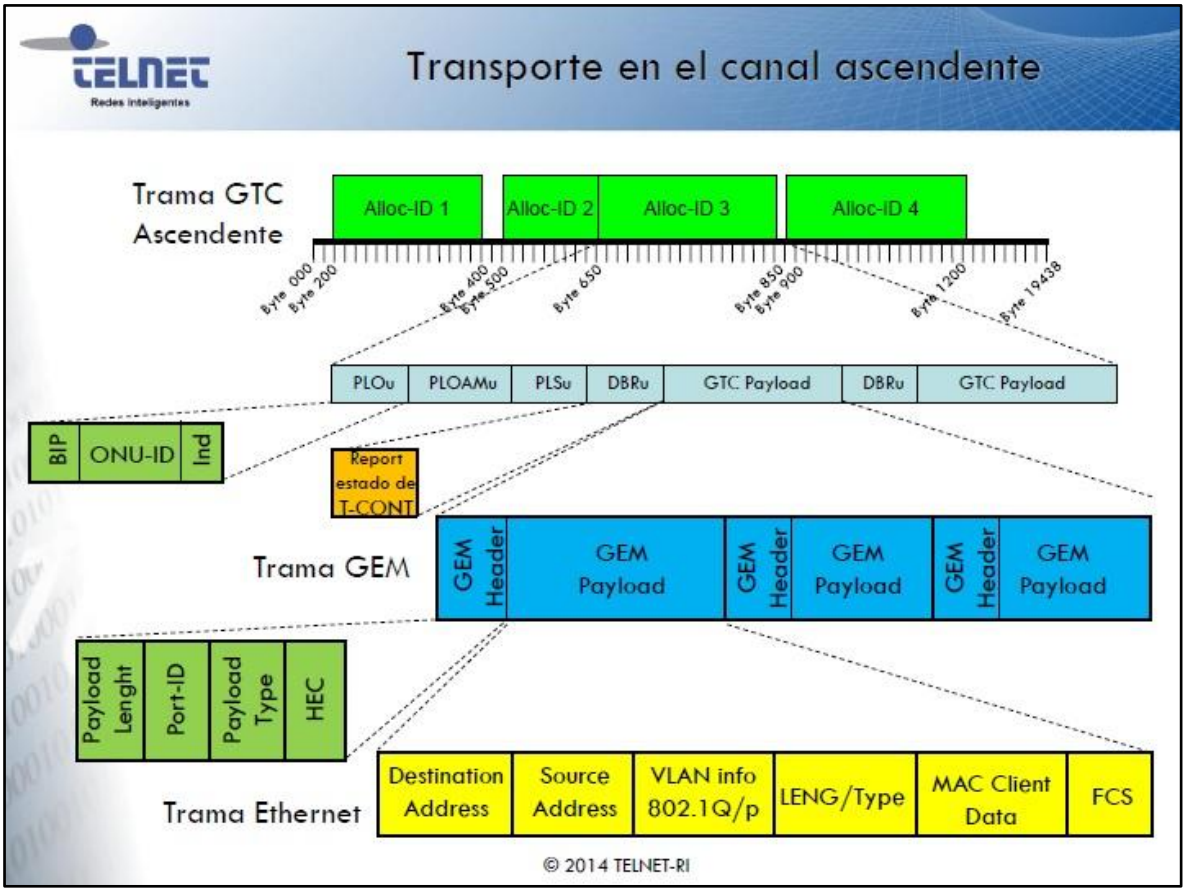


Figura 17.Estructura de trama en el canal ascendente de la GPON [65].

En términos funcionales, Ethernet y GEM conservan exactamente el mismo cometido tanto en el sentido descendente como en el ascendente: la trama Ethernet sigue siendo la unidad de servicio que porta la información del usuario sin modificación, y la trama GEM mantiene su encabezado de 5 bytes para identificar el flujo lógico, permitir la fragmentación y proteger la cabecera, independientemente de la dirección de transmisión [30] [31].

La única variación significativa aparece en la trama GTC.

- ***Descendente***: se emite como una secuencia continua y periódica de 125 µs que incluye el bloque de control PCBd (sincronismo, OAM, BIP y BWmap) seguido de un gran bloque de payload.

- ***Ascendente***: adopta la forma de ráfagas (*bursts*) temporizadas. Cada ráfaga comienza con un *Burst Header* (BIP, ONU-ID, indicadores de estado) y, a continuación, presenta secciones opcionales de control (PLOu, PLOAMu, DBRu)

antes de los datos. La longitud y el momento de cada ráfaga vienen determinados por el BWmap descendente [30] [32].

En consecuencia, los campos de la trama GTC quedan de la siguiente forma:

Encabezado de ráfaga (*Burst Header*):

Tabla 6. Campos del encabezado de ráfaga (canal ascendente).

Campo	Longitud	Función
BIP	8 bits	Bit-Interleaved Parity sobre todos los bytes transmitidos desde el último BIP; permite medir la tasa de errores.
ONU-ID	8 bits	Identificador único de la ONU que envía la ráfaga; asignado durante el proceso de *ranging*.
Ind	8 bits	Indicaciones de estado en tiempo real: bit 7 → PLOAM urgente, bit 6 → FEC ON/OFF, bit 5 → RDI, bits 4-0 reservados.

Secciones de control ascendente:

Tabla 7. Campos de la sección de control (canal ascendente).

Sección	Longitud típica	Descripción
PLOu	Preamb. + Delim. + 3 bytes	Overhead físico para sincronización, alineación y entrega del encabezado de ráfaga.
PLOAMu	13 bytes	Canal OAM ascendente: mensajes de activación, alarmas y mantenimiento.
PLSu	120 bytes (opcional)	Secuencia de nivelación de potencia; característica

		obsoleta usada para ajustar la potencia láser.
DBRu	4 bytes	Informe dinámico de ancho de banda: la ONU reporta bytes en cola por Alloc-ID para el algoritmo DBA.

Capítulo 4. Servicios y aplicaciones NFV

4.1. Introducción

La Virtualización de Funciones de Red (*Network Functions Virtualization*, NFV) es una tecnología promovida por el Instituto Europeo de Normas de Telecomunicaciones (ETSI) que permite desacoplar las funciones de red del *hardware* dedicado, ejecutándolas como *software* sobre servidores estándar [33]. De este modo, funciones como la conmutación, el enrutamiento y la seguridad, que anteriormente requerían equipos especializados, pueden desplegarse con mayor flexibilidad, reduciendo costes al utilizar infraestructuras comercialmente disponibles.

Al combinarse con el paradigma de las Redes Definidas por *Software* (*Software Defined Networks*, SDN), la NFV permite una gestión centralizada y programable de los servicios de red. Esto mejora significativamente la agilidad y escalabilidad de la infraestructura.

En nuestra red experimental, basada en una infraestructura GPON virtualizada se han implementado múltiples servicios y aplicaciones utilizando plataformas NFV gestionadas mediante SDN. Este capítulo describe los componentes virtualizados de la red y los servicios implementados, entre ellos:

- ***Open vSwitch (OvS)****:* un conmutador virtual encargado de la conmutación y control dinámico del tráfico.
- Servicios desplegados sobre la infraestructura óptica GPON:
    - IPTV (*Internet Protocol Television*).
    - VoIP (*Voice over IP*).
    - CCTV (*Closed Circuit Television*).
    - VyOS *Universal Router*.

Cada apartado detalla la tecnología empleada, explica brevemente su funcionamiento y describe cómo se integra específicamente en nuestra arquitectura GPON gestionada por SDN, enfatizando los beneficios obtenidos en eficiencia operativa, flexibilidad y gestión unificada.

4.2. *Open vSwitch*

4.2.1 Definición

Open vSwitch (OvS) es un conmutador virtual de código abierto, de calidad de producción y multinivel, diseñado para habilitar la virtualización de redes y arquitecturas basadas en SDN [34]. OvS se utiliza ampliamente en entornos *cloud* y de virtualización, integrándose de forma nativa con hipervisores como KVM (*Kernel-based Virtual Machine*), Xen o VirtualBox. Permite una gestión centralizada del tráfico mediante interfaces programables (p. ej., protocolos *OpenFlow* y OVSDB para control y configuración dinámica) [34].

En nuestro proyecto, OvS actúa como el *switch* virtual principal de la red GPON, brindando la capacidad de manejar el tráfico de forma centralizada desde el controlador SDN y de interconectar las funciones virtuales desplegadas.

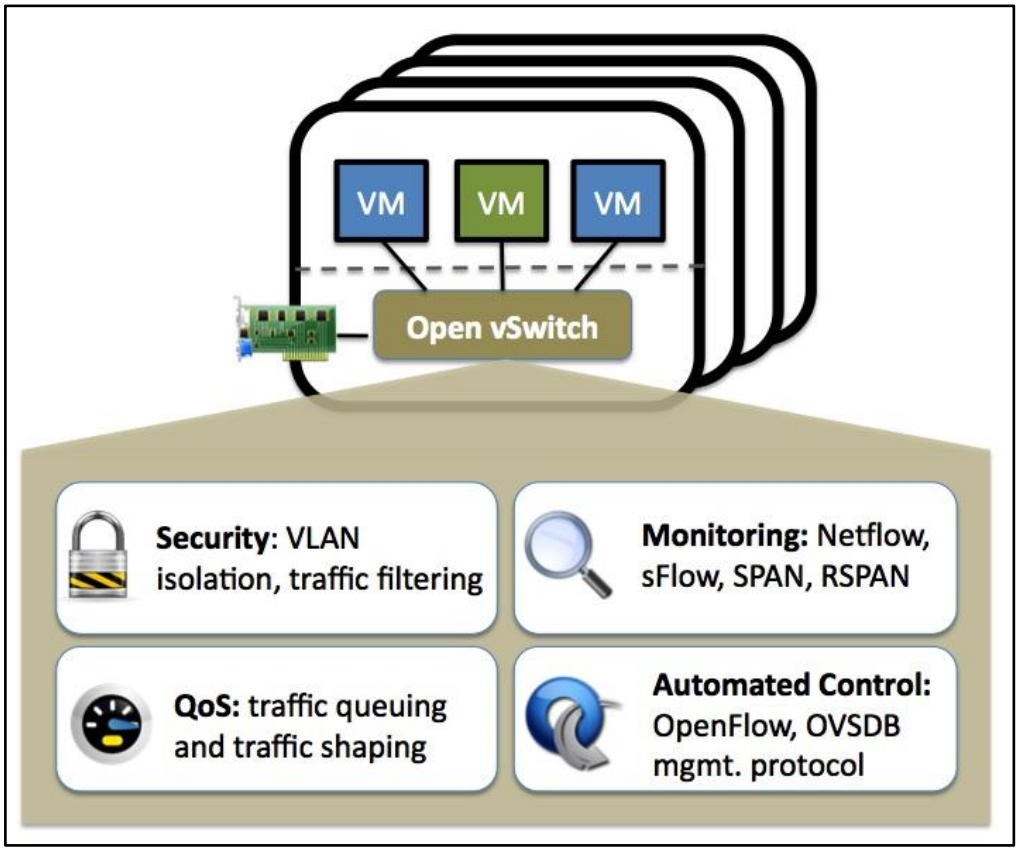


Figura 18. Esquema y características generales del OvS [66].

4.2.2. Arquitectura y funcionamiento

OvS se compone de varios módulos que trabajan en conjunto para ofrecer un rendimiento óptimo:

***Kernel module**:* Proporciona el plano de datos de alto rendimiento dentro del *kernel* de Linux y mantiene una tabla de flujos (incluyendo *"megaflows"*) para el encaminamiento rápido de paquetes. Cuando llega el primer paquete de un flujo nuevo, éste se envía al espacio de usuario (*deamon* ovs-vswitchd) para ser procesado según las reglas configuradas; una vez determinada la acción, se instala una entrada en caché en el módulo *kernel*. De este modo, los paquetes subsiguientes que coincidan con ese flujo pueden ser conmutados directamente en el *kernel*, sin necesidad de pasar de nuevo por el plano de control en espacio de usuario [35]. Este mecanismo de *fast-path* reduce significativamente la latencia y carga de procesamiento, ya que solo el primer paquete de cada flujo incurre en la consulta al demonio de usuario.

***ovs-vswitchd**:* Es el *daemon* en espacio de usuario que administra la lógica de conmutación. Controla la creación y configuración de los *bridges* virtuales, las reglas de flujo y la aplicación de políticas. Se comunica de forma continua con el módulo de base de datos (OVSDB) para recibir y aplicar cambios de configuración, asegurando una adaptación dinámica del *switch* a las políticas definidas [36].

***ovsdb-server**:* Es el proceso que maneja la base de datos de configuración de OvS. En dicha base de datos se almacenan los parámetros y objetos de la conmutación virtual: *bridges*, puertos, túneles, VLAN, reglas, etc. Su diseño transaccional garantiza la consistencia de la configuración, y permite persistir los cambios incluso tras reinicios del sistema. El protocolo OVSDB, estándar de facto, facilita que controladores externos (como un controlador SDN) puedan gestionar la configuración de OvS de forma remota mediante transacciones bien definidas.

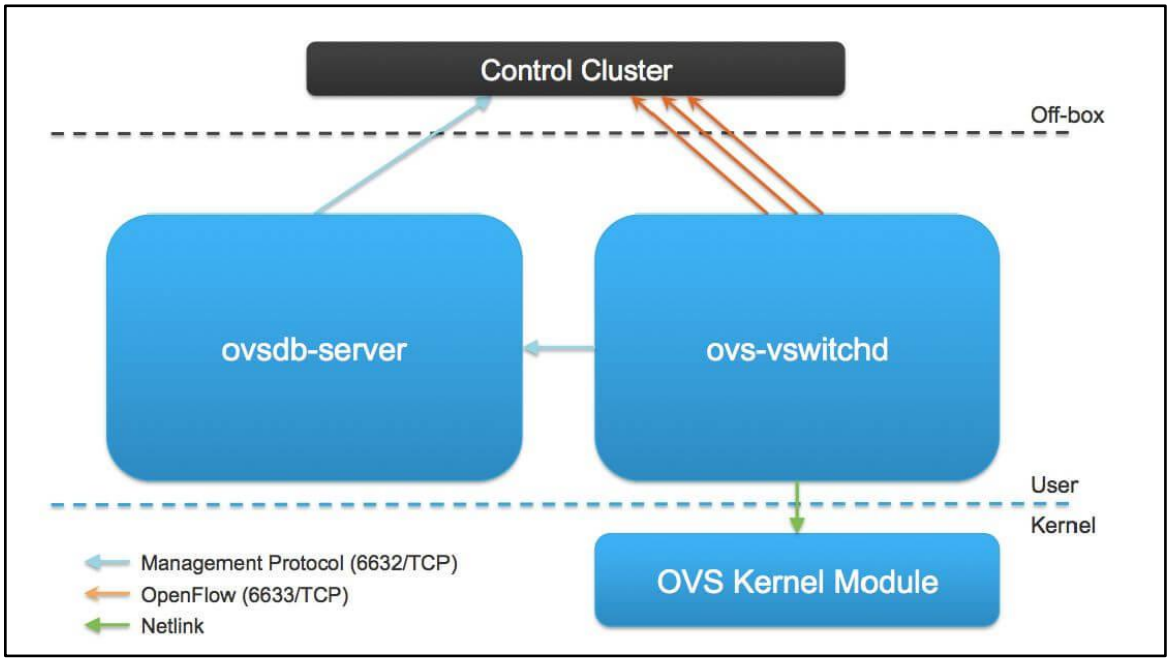


Figura 19. Arquitectura lógica del OvS [67].

El proceso interno de encaminamiento en OvS puede dividirse en dos fases esenciales [35]:

- ***Primer paquete (miss)**:* El primer paquete de un flujo desconocido es enviado desde el *kernel* al demonio *ovs-vswitchd* en espacio de usuario (evento de *flow miss*). El demonio determina la acción a aplicar según las reglas de flujo configuradas (por ejemplo, reenviar por cierto puerto, descartar, modificar encabezados, etc.).

- ***Paquetes posteriores (hit)**:* Tras la decisión inicial, *ovs-vswitchd* instala una entrada de flujo en la caché del módulo *kernel*, de forma que los siguientes paquetes del mismo flujo se conmutan directamente en el *kernel (fast-path)* sin intervención del demonio. Esto permite procesar el resto del tráfico de ese flujo con latencia mínima y alto rendimiento.

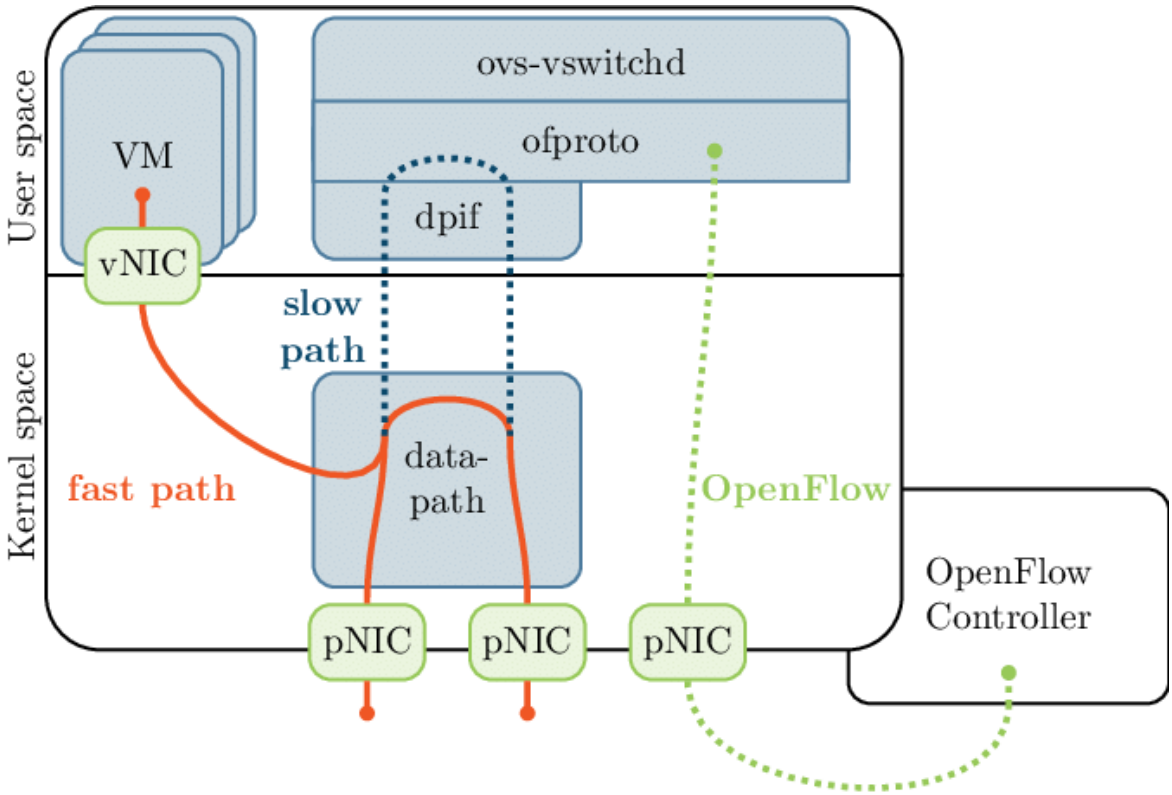


Figura 20. Arquitectura interna con rutas de decisión del OvS [68].

La figura anterior muestra con claridad las dos rutas internas (encaminamiento) de *Open vSwitch* (OvS):

- ***Slow-path (miss)**:* La flecha punteada azul; el primer paquete que no coincide con ninguna entrada en caché se envía del módulo *kernel* a *ovs-vswitchd* en espacio de usuario, donde se decide la acción correspondiente.

- ***Fast-path (hit)**:* La flecha continua naranja; una vez instalada la entrada de flujo en el *kernel*, los paquetes siguientes se procesan íntegramente en el *datapath*, evitando el paso por usuario y reduciendo la latencia.

4.2.3. Integración en redes GPON

Aunque OvS está orientado principalmente a entornos de centros de datos y virtualización, su arquitectura modular y compatibilidad con protocolos SDN lo hacen adaptable a otros escenarios, como lo son las GPON. En nuestra red experimental, OvS se despliega junto a la OLT para gestionar de forma centralizada la distribución y segmentación del tráfico hacia las ONT. Algunas ventajas de integrar OvS en la red GPON son:

- ***Segmentación dinámica de tráfico**:* Mediante técnicas de virtualización de red como la asignación de VLAN, OvS permite separar el tráfico de distintos servicios o clientes de forma flexible. Por ejemplo, se pueden aislar flujos de IPTV, VoIP y datos en diferentes redes lógicas, manteniendo la convergencia sobre el mismo medio físico.

- ***Control centralizado vía SDN**:* Gracias a protocolos abiertos como *OpenFlow*, un controlador SDN puede programar en tiempo real las reglas de encaminamiento en OvS. Esto posibilita optimizar el uso del ancho de banda GPON asignando prioridades o anchos de banda garantizados a ciertos servicios, y adaptando las políticas instantáneamente ante cambios en la demanda.

- ***Interoperabilidad y estándar abierto**:* OvS, al ser *software* abierto y ampliamente soportado, puede interactuar con múltiples equipos y tecnologías. Esto facilita la convergencia de la solución GPON tradicional con la infraestructura virtual; por ejemplo, el controlador puede comunicarse con OvS para coordinar el filtrado *multicast*, QoS o reconfiguración de topologías sin necesidad de *hardware* propietario específico.

En resumen, la inclusión de *Open vSwitch* como núcleo de conmutación virtual en la red GPON aporta la flexibilidad de una red definida por *software* al segmento de acceso, permitiendo ajustar y escalar el comportamiento de la red óptica mediante *software*.

4.2.4. Comandos básicos

La administración de *Open vSwitch* se realiza mediante una serie de herramientas de línea de comandos (CLI) que facilitan desde la creación de *bridges* hasta el control de flujos. A continuación, se muestran algunos comandos esenciales con ejemplos [36]:

1. ***Crear un Bridge***

   - ***Comando*:**
     ovs-vsctl add-br br0

   - ***Explicación*:** Crea un *bridge* virtual denominado “br0” que actuará como el punto central de conmutación para agrupar interfaces (físicas o virtuales).

2. ***Listar Bridges Configurados***

   - ***Comando*:**
     ovs-vsctl list-br

   - ***Explicación*:** Muestra la lista de todos los *bridges* configurados en el sistema, facilitando la verificación de la infraestructura implementada.

3. ***Agregar una Interfaz al Bridge***

   - ***Comando*:**
     ovs-vsctl add-port br0 eth1

   - ***Explicación*:** Añade la interfaz “eth1” al *bridge* “br0”, integrándola en la gestión de tráfico de OvS.

4. ***Configurar una Interfaz en Modo Acceso (Asignar Tag VLAN)***

   - ***Comando*:**
     ovs-vsctl set port eth1 tag=10

   - ***Explicación*:** Configura la interfaz “eth1” en modo acceso, asignándole el *tag* VLAN 10 para que todo su tráfico pertenezca a esa VLAN.

5. ***Configurar una Interfaz en Modo Trunk (Permitir Múltiples VLANs)***

   - ***Comando*:**
     ovs-vsctl set port eth2 trunks=10,20,30

   - ***Explicación*:**

     - El primer comando pone a “eth2” en modo *trunk*.

     - El segundo define la lista de VLANs (10, 20 y 30) que se permitirán a través de ese puerto, permitiendo el paso de tráfico etiquetado de diversas VLANs.

6. ***Listar Puertos Asociados a un Bridge***

   - ***Comando*:**
     ovs-vsctl list-ports br0

   - ***Explicación*:** Muestra las interfaces actualmente asociadas al *bridge* “br0”, permitiendo confirmar qué puertos están configurados.

7. ***Ver la Configuración Detallada de una Interfaz***

   - ***Comando*:**
     ovs-vsctl list port eth1

   - ***Explicación*:** Retorna todos los parámetros y opciones asociados a la interfaz “eth1” (incluyendo el estado del *tag* y el modo VLAN), facilitando la revisión detallada de la configuración.

8. ***Eliminar una Interfaz del Bridge***

   - ***Comando*:**
     ovs-vsctl del-port br0 eth1

   - ***Explicación*:** Remueve la interfaz “eth1” del *bridge* “br0”, desvinculándola de la gestión de OvS y permitiendo su reconfiguración o retiro.

9. ***Ver Flujos Activos en el Bridge***

   - ***Comando*:**
     ovs-ofctl dump-flows br0

   - ***Explicación*:** Muestra la tabla de flujos actualmente instalada en “br0”, proporcionando detalles como coincidencias, acciones, y contadores de paquetes y *bytes*.

10. ***Agregar un Flujo de OpenFlow (Ejemplo: Permitir Tráfico HTTP)***

    - ***Comando*:**
      ovs-ofctl add-flow br0 "tcp,tp_dst=80, actions=normal"

    - ***Explicación*:** Inserta una regla en la tabla de flujos del *bridge* “br0” para reenviar tráfico TCP cuyo destino sea el puerto 80, permitiendo así el acceso a servicios HTTP.

11. ***Eliminar Todos los Flujos del Bridge***

- ***Comando*:**
  ovs-ofctl del-flows br0

- ***Explicación*:** Borra todas las entradas de la tabla de flujos en “br0”. Esta acción es útil para reiniciar la configuración de flujo o durante procesos de depuración.

12. ***Mostrar Estadísticas del Datapath***

- ***Comando*:**
  ovs-dpctl show

- ***Explicación*:** Proporciona información sobre el estado y rendimiento del *datapath* de OvS, incluyendo datos sobre puertos, número de flujos instalados y estadísticas de aciertos en caché.

13. ***Reconfigurar Opciones de un Puerto (Ejemplo: Bonding)***

- ***Comando*:**
  ovs-vsctl set port eth1 other_config:bond_mode=active

- ***Explicación*:** Permite modificar opciones avanzadas de un puerto, en este caso para configurar el modo de enlace (*bonding*) a “*active*”, lo que puede ser útil en implementaciones con enlaces redundantes.

14. ***Consultar el Estado del Demonio de OvS***

- ***Comando*:**
  ovs-appctl info

- ***Explicación*:** Ofrece información detallada del estado del demonio *ovs-vswitchd*, incluyendo métricas de rendimiento y otros datos relevantes para la supervisión en tiempo real.

4.3. Virtual Router VyOS

4.3.1 Definición

VyOS es una plataforma de *routing* de código abierto basada en GNU/*Linux* que integra múltiples funciones de red en una única imagen de *software*. Gracias a su arquitectura

modular y flexible, VyOS puede implementar servicios esenciales como enrutamiento dinámico, cortafuegos (*firewall*), VPN (*Virtual Private Network*), NAT (*Network Address Traslation*), entre otros, todo gestionado a través de una interfaz de configuración unificada. Se trata de un sistema operativo de red de nivel empresarial que compite con soluciones comerciales tradicionales, pero que puede ejecutarse en *hardware* estándar x86 o en entornos virtualizados y en la nube [37]. Al ser independiente de la plataforma en la que se ejecute, VyOS es ideal para entornos NFV, ya que puede desplegarse tanto sobre servidores *bare-metal* como sobre máquinas virtuales, adaptándose a diversas topologías.

4.3.2 Arquitectura y funcionamiento

VyOS se caracteriza por una arquitectura modular que unifica varias funciones de red bajo una misma interfaz de administración. Entre los aspectos más destacados de su funcionamiento, podemos mencionar:

- ***CLI unificada con gestión transaccional**:* VyOS provee una interfaz de línea de comandos consistente para configurar todos sus servicios. La sintaxis es declarativa, permitiendo editar una configuración candidata que luego se confirma (*commit*) o de se vuelve atrás (*rollback)* en conjunto. Este enfoque garantiza que cualquier cambio pueda revisarse y revertirse si es necesario, proporcionando seguridad y control en la administración de la red.

- ***Funciones de red integradas**:* Dentro de una instancia VyOS coexisten funcionalidades de *router* IP (*routing* estático y dinámico mediante protocolos como OSPF, BGP), cortafuegos de inspección de paquetes (basado en *iptables/nftables*), VPN (soporta *IPsec*, *OpenVPN*, *WireGuard*, etc.), NAT (*Source NAT, Destination NAT*) y otros servicios (DHCP, QoS, etc.). Esta integración elimina la necesidad de múltiples dispositivos especializados, ya que una sola máquina VyOS puede asumir el rol de varios equipos de red tradicionales.

- ***Flexibilidad de despliegue**:* VyOS puede operar en una amplia gama de plataformas gracias a su naturaleza de *software*. Puede instalarse en *hardware* dedicado (*appliances*, PC estándar) o como una máquina virtual en KVM, *VMware*, *Hyper-V*, *VirtualBox*, e incluso desplegarse en instancias de nube pública. Esta portabilidad permite que lo utilicemos tanto en laboratorios virtualizados como en producción. En nuestro escenario, VyOS corre como máquina virtual dentro del entorno NFV, facilitando su conexión con OvS y otros componentes de la red.

- ***Actualizaciones e imagen inmutable**:* VyOS se distribuye como una imagen de sistema (ISO/OVA) que incluye todo el sistema operativo y las aplicaciones de red. Las actualizaciones se realizan mediante instalación de una nueva imagen en paralelo, lo que aporta estabilidad (se puede retroceder a la imagen anterior en caso de fallo). Esta filosofía, similar a la de enrutadores comerciales, aporta confiabilidad al introducir cambios de *software*.

En conjunto, estos elementos hacen de VyOS una solución muy robusta y manejable para ejercer de *router* virtual en entornos NFV [37].

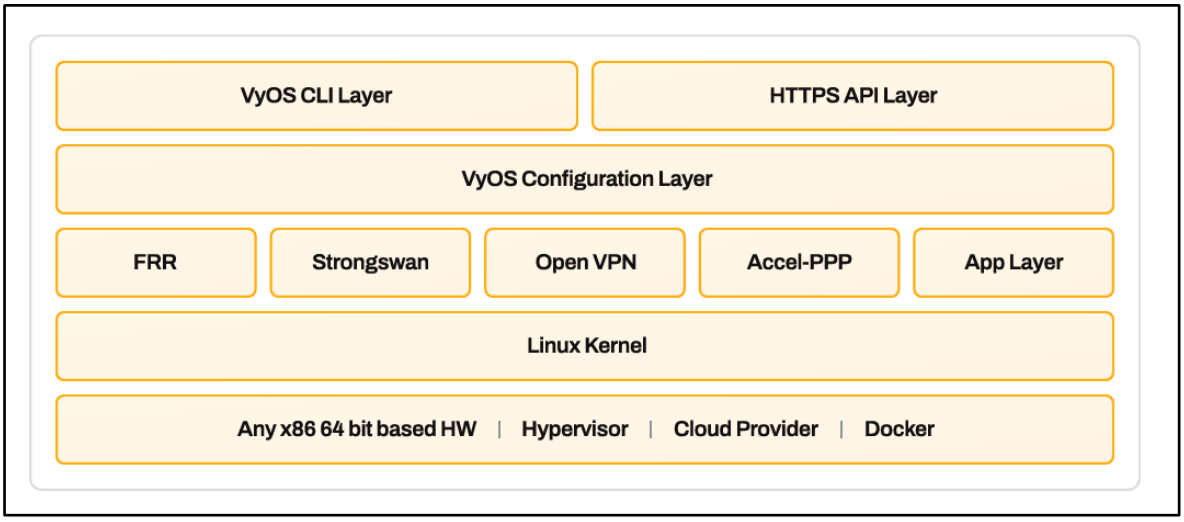


Figura 21. Arquitectura modular de VyOS [69].

4.3.3 Integración en entornos GPON

Esta sección detalla la integración del *router* VyOS para la provisión del servicio de datos mediante una VLAN dedicada.

Flujo de Datos y Conectividad del Servicio

El flujo de datos desde las ONT hacia Internet sigue un camino específico a través de la VLAN 100. La terminal de línea óptica (OLT) encapsula el tráfico generado por cada ONT en un GEM-Port específico, etiquetándolo con la VLAN 100 para identificarlo como tráfico de datos.

El puerto *Ethernet* de la OLT opera en modo *trunk* y transporta esta VLAN etiquetada hacia el *Open vSwitch* (OVS) ejecutándose sobre el Ordenador/Servidor central. Para que los abonados puedan acceder a Internet, es fundamental que el tráfico de la VLAN 100 llegue al OVS, donde se establece la conexión con el puerto virtual correspondiente que aloja el servidor/*router* VyOS. Este servidor/*router* virtual actúa como la puerta de enlace hacia Internet, procesando todo el tráfico de datos de los abonados conectados a través de la VLAN 100

Topología de interfaces VyOS:

Tabla 8. Relación de interfaces del router VyOS.

Interface VyOS	Conexión lógica	Descripción
eth0.100	VLAN 100 (datos)	Puerta de enlace predeterminada para todas las ONT
eth1	Red troncal	Conectividad hacia el núcleo IP del proveedor e Internet
loopback0	Gestión	Dirección de gestión y monitorización (SNMP, NetFlow)

El tráfico generado por las ONT se dirige desde la interfaz eth0.100 hacia eth1, donde se implementa NAT *Masquerade*. Esta configuración permite que múltiples abonados compartan un conjunto limitado de direcciones IP públicas, ocultando así su esquema de direccionamiento interno.

En síntesis, tenemos la siguiente secuencia que sigue nuestra red para el servicio de datos:

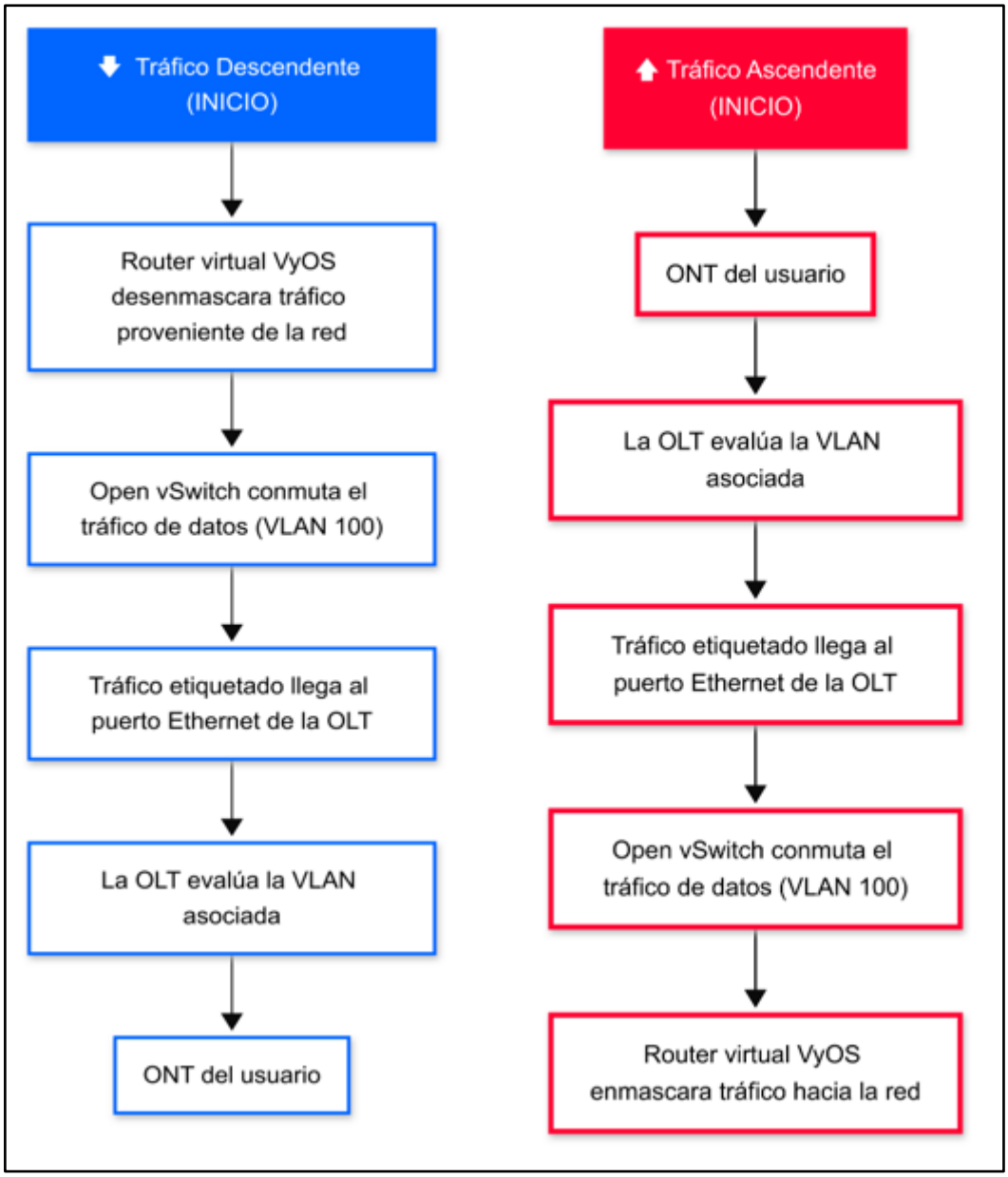


Figura 22. Secuencia del tráfico de datos ascendente y descendente. Elaboración propia.

4.4. Servicio IPTV

4.4.1 Definición

La Televisión por IP (IPTV) es un sistema que entrega los servicios televisivos utilizando el protocolo IP (*Internet Protocol*) sobre una red de banda ancha, en lugar de las señales tradicionales de cable o satélite [38]. Esto permite ofrecer al usuario contenido audiovisual (canales de TV en vivo, video bajo demanda, etc.) a través de su conexión de datos, aprovechando técnicas de *streaming* y *multicast*, e integrando la TV dentro de los servicios convergentes de telecomunicaciones.

4.4.2 Funcionamiento del protocolo IGMPv3

El protocolo de *Internet Group Management* (IGMP) es la pieza clave para la gestión eficiente de transmisiones *multicast* en redes. En nuestro servicio IPTV, usamos IGMPv3 (versión 3) para que los clientes puedan unirse o salir dinámicamente del grupo *multicast* del canal de TV, recibiendo únicamente las emisiones que les interesan. A continuación, se describe el proceso:

- ***Emisión multicast desde el servidor IPTV**:* El servidor de vídeo (VLC) envía el *stream* de TV en vivo a la dirección *multicast* 239.1.1.1 (dirección IP utilizada en la implementación) viajando una sola vez hasta la OLT.

- ***Distribución en la red GPON**:* La OLT encapsula el flujo *multicast* en tramas GEM sobre la fibra y, a través del *splitter* óptico, lo hace llegar a todas las ONT de los abonados simultáneamente. Inicialmente este tráfico *multicast* es "*broadcast*" en el segmento GPON, pero gracias a IGMP *Snooping* (escucha de IGMP) en los *switches*/OLT, se puede controlar su distribución.

- ***Solicitud de unión por el cliente (IGMP Join)***: Cuando un usuario final selecciona el canal en su cliente IPTV, el dispositivo envía un mensaje IGMPv3 *Membership Report* anunciando su intención de unirse al grupo *multicast* 239.1.1.1. IGMPv3 permite que este mensaje incluya información sobre las fuentes deseadas del flujo. En nuestro caso de *Single Source Multicast* (SSM), el reporte IGMPv3 indicará en modo *Include* que acepta tráfico únicamente del servidor IPTV (filtrando cualquier otra posible fuente).

- ***Propagación y filtrado**:* Los equipos de la red (la OLT o un *switch* virtual con función de IGMP *Snooping*, como puede ser OvS) interceptan los mensajes IGMP.

IGMP *Snooping* construye tablas de suscripción: solo mantiene el puerto (o la ONT) del usuario que solicitó el canal como asociado al grupo 239.1.1.1 [39]. De este modo, cuando la OLT reciba el flujo *multicast*, reenviará ese tráfico solamente hacia la(s) ONT correspondientes a usuarios suscritos, en lugar de inundarlo a todos [39]. Esto optimiza el uso de ancho de banda en la GPON y evita saturar de datos a quienes no los han solicitado.

- ***Recepción del canal**:* Finalmente, la ONT del usuario que se unió pasa el flujo *multicast* a su red doméstica, y el cliente IPTV reproduce el canal en tiempo real. Si otro usuario en la misma red GPON desea el canal, la red no necesita una nueva transmisión: simplemente su ONT se añadirá al grupo existente y recibirá el mismo flujo distribuido.

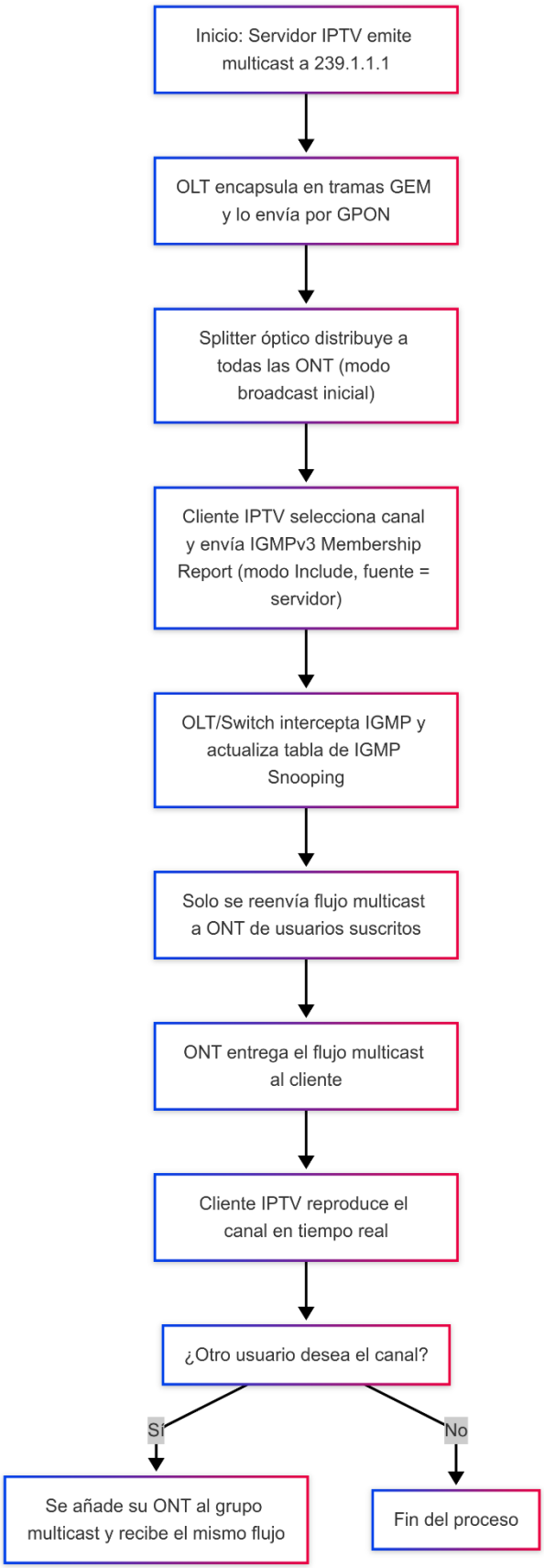


Figura 23. Secuencia del tráfico de IPTV. Elaboración propia.

4.4.3 Integración de IPTV en la red GPON

Una vez descrito el funcionamiento a nivel de protocolo de red, pasamos a ver como este servicio se ofrece.

En nuestro esquema de red GPON virtualizada se ha desplegado un servicio IPTV como parte de la oferta. El servidor central de IPTV utiliza *software* VLC para emitir en tiempo real contenido multimedia como flujo *multicast* hacia la red (dirección de grupo 239.1.1.1). Gracias a la arquitectura GPON, la OLT replica este flujo *multicast* hacia todos los ONT de los usuarios a través del divisor óptico, sin duplicar el tráfico por usuario (una única emisión es compartida por todos). Cuando un usuario desea sintonizar el canal IPTV transmitido, su cliente IPTV realiza un mecanismo de suscripción a ese flujo *multicast* empleando el protocolo IGMP.

Para poder llevar a cabo este servicio en la red implementada en este TFG, hay que llevar a cabo las siguientes configuraciones.

Lado del servidor

Del lado del servidor, hemos de configurar los siguientes parámetros para poder emitir correctamente en la dirección *multicast* establecida para que los usuarios finales de la red GPON puedan unirse al grupo y visualizar la transmisión correspondiente.

En primer lugar, tenemos que seleccionar la opción de “emitir” en el VLC servidor y nos saldrá el siguiente menú:

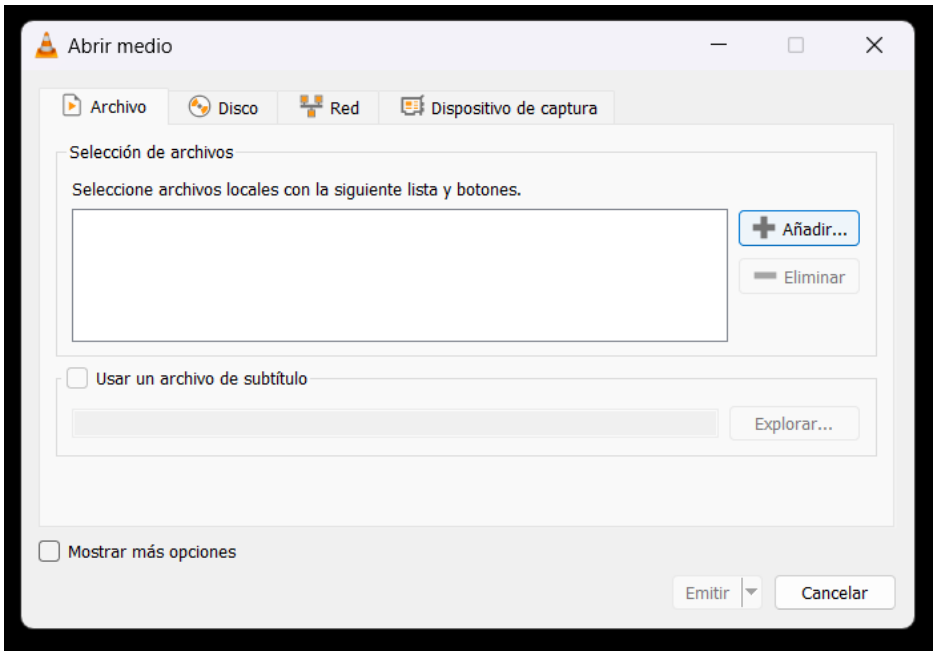


Figura 24. Configuración servidor IPTV 1.

Estando en el menú representado por la figura 24, añadimos un archivo multimedia, que será el ofrecido a los usuarios finales que estén suscritos a este servicio. Posterior pasaremos al siguiente menú que se muestra en la figura 25.

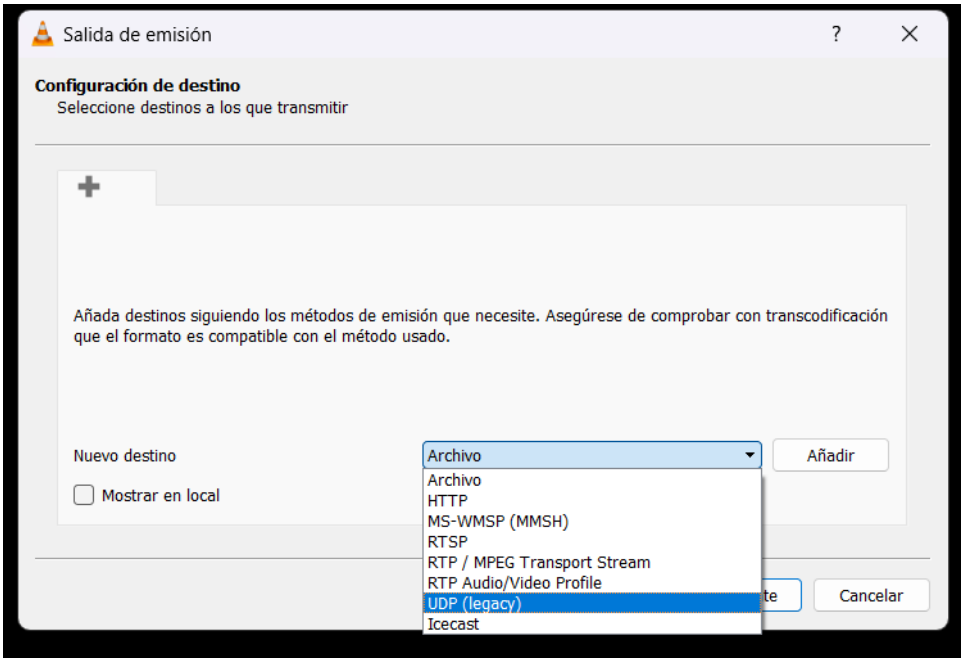


Figura 25. Configuración servidor IPTV 2.

En la figura 25 se elige el protocolo deseado para la transmisión de video. Seleccionamos el que consideremos, en este caso UDP (legacy) y pasamos a configurar la dirección *multicast* y el puerto en el que se va a emitir la transmisión, como se observa en la figura 26.

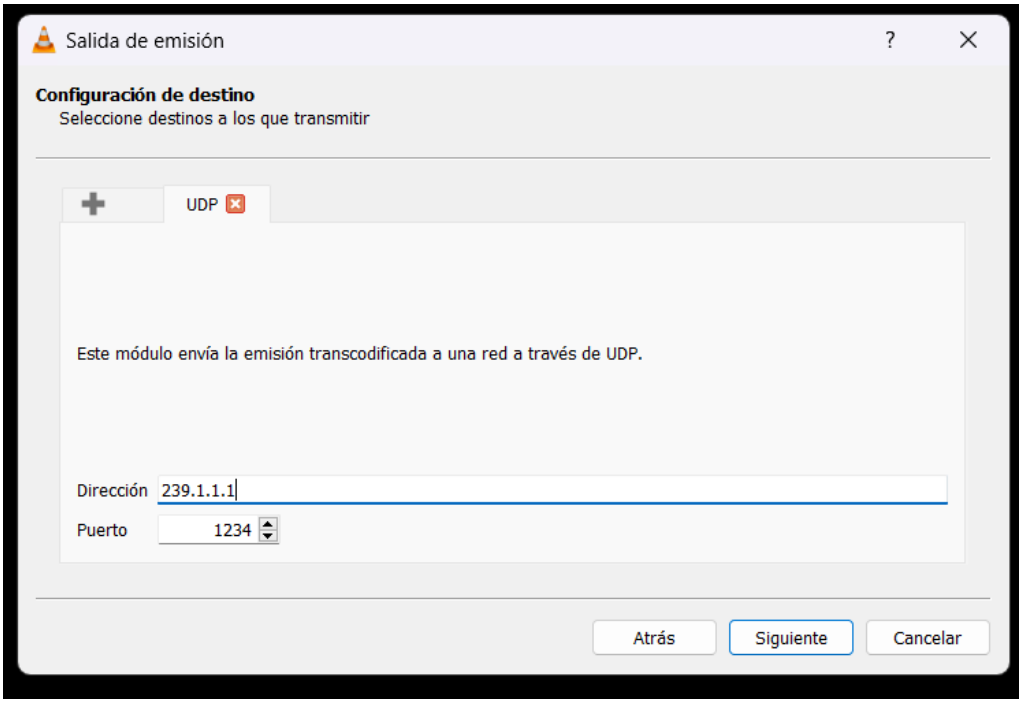


Figura 26. Configuración servidor IPTV 3.

Si todo ha ido bien, se emitirá el archivo multimedia seleccionado en la dirección *multicast*: 239.1.1.1 y el puerto 1234.

Lado del cliente

En el otro extremo, se encuentra el usuario final, el cual cuenta con un cliente IPTV para unirse al grupo *multicast* que este emitiendo dentro de la red.

Para ello, el cliente tiene que seleccionar la opción de reproducir dentro del del VLC en cuestión y establecer la dirección *multicast* a la que se quiere unir como se muestra en la figura siguiente:

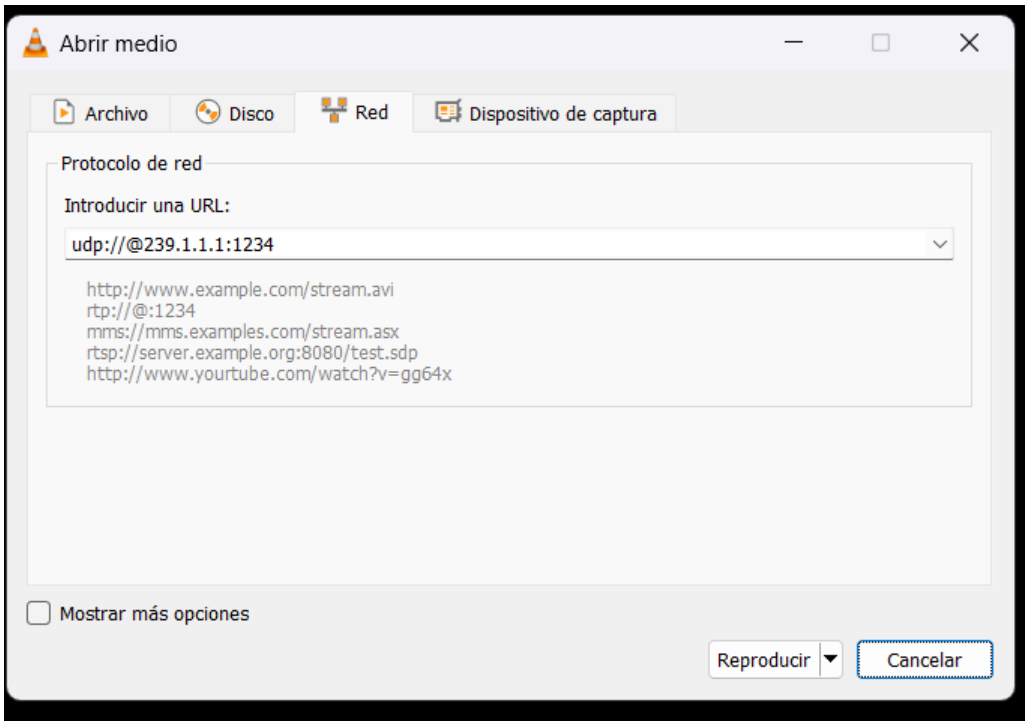


Figura 27. Configuración servidor IPTV 4.

4.5. Servicio VoIP

4.5.1 Definición

En la red GPON desarrollada, se implementa un servicio de Voz sobre Protocolo de Internet (VoIP), el cual permite comunicaciones de voz utilizando la infraestructura de datos óptica de la red. VoIP es la tecnología que digitaliza la voz en paquetes de datos IP y la transporta a través de la red en lugar de emplear la telefonía conmutada tradicional [40]. En esencia, la voz se envía por Internet de igual forma que otros datos, aprovechando la capacidad de la red GPON para integrar servicios. Este servicio forma parte de la solución convergente de nuestra red, orientada a ofrecer comunicaciones de alta calidad mediante la priorización y gestión centralizada del tráfico de voz.

4.5.2 Protocolos de señalización y transporte

Protocolo de inicialización de sesión (SIP)

Para el establecimiento y control de las llamadas VoIP se utiliza el Protocolo de Iniciación de Sesión (SIP), definido por IETF como estándar para señalización de sesiones multimedia [40]. SIP se encarga de la señalización: es decir, del registro de terminales, del

marcado o invitación a una llamada, la negociación de parámetros (*codecs*, direcciones) y la terminación de la sesión.

Protocolo de transporte en tiempo real (RTP)

Una vez establecida la llamada a través de SIP, el audio se transmite mediante el Protocolo de Transporte en Tiempo Real (RTP), el cual opera sobre UDP proporcionando el envío de paquetes de voz con los requisitos de baja latencia y mínima pérdida necesarios para mantener la calidad de la comunicación [41]. En VoIP, RTP lleva la voz codificada (p. ej., con códecs G.711, OPUS, etc.), incluyendo mecanismos para numerar paquetes y detectar pérdidas o *jitter*.

Integración SIP+RTP

La combinación SIP+RTP conforma la base de cualquier servicio VoIP moderno: SIP instala la llamada (es decir, quién habla con quién, cuándo y cómo), y RTP mueve la voz en tiempo real entre los interlocutores.

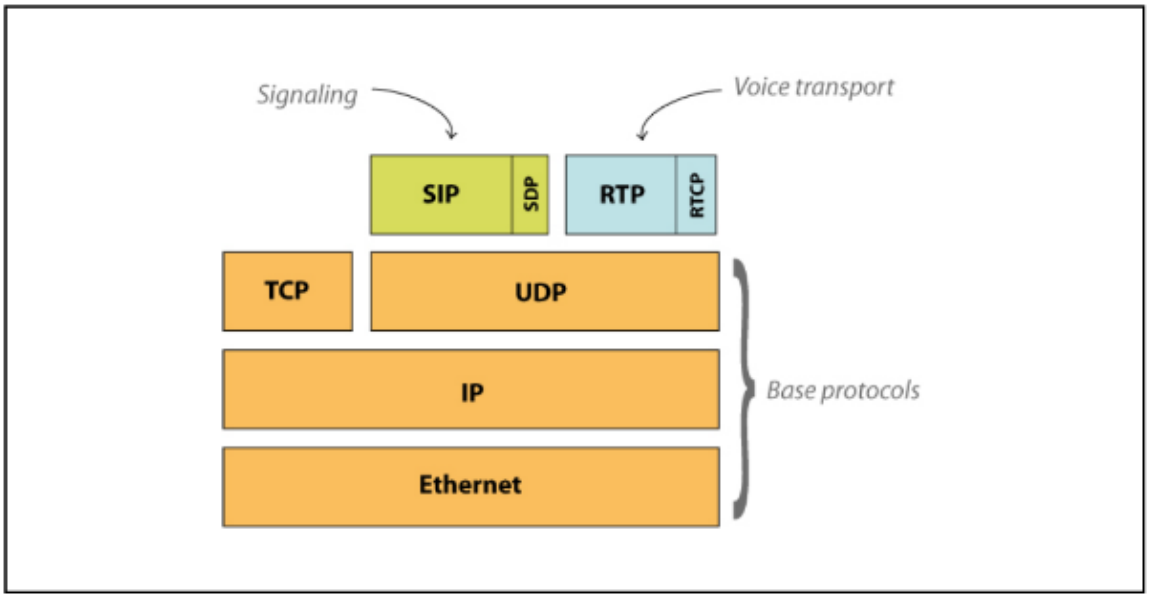


Figura 28. Pila de protocolos utilizada en comunicaciones VoIP.

4.5.3 Integración de VoIP a la red GPON

Para integrar este servicio en la red desarrollada en este TFG, es necesario implementar un componente clave en la infraestructura de comunicaciones. Este es la centralita IP PBX (*Private Branch Exchange*), que actúa como servidor central de comunicaciones VoIP.

Figura 29. Servidor PBX Asterisk.

Nuestra PBX virtualizada administra las llamadas dentro de la red GPON, desempeñando varias funciones:

- ***Establecimiento y enrutamiento de llamadas**:* La PBX recibe las solicitudes SIP de inicio de llamada desde los teléfonos IP o adaptadores de los usuarios. Según el número marcado, decide si la llamada es interna (otro abonado en la misma red) o externa. En llamadas internas, la PBX conecta directamente a los dos usuarios a través de la red IP local. En llamadas a la red telefónica tradicional, la PBX enruta la llamada hacia un *gateway* SIP-Trunk o PSTN de salida. En ambos casos, la PBX gestiona la señalización SIP necesaria para *tirar* (hacer sonar) el otro extremo y conecta los flujos RTP una vez aceptada la llamada [43].

- ***Gestión de servicios avanzados**:* La centralita proporciona características de valor añadido propias de sistemas empresariales. Por ejemplo: buzón de voz (almacena mensajes cuando un usuario no contesta), desvío de llamadas (redirige llamadas entrantes a otro número según horario u ocupación), conferencias de voz multiusuario, locuciones automáticas (IVR) y colas de espera con música/enunciados. Estos servicios enriquecen la experiencia del usuario y

mejoran la eficiencia en comunicaciones corporativas, siendo todos administrados por la lógica de la PBX [43].

- ***Integración en la infraestructura convergente***: Al estar alojada como función virtual en la red GPON, la PBX puede interactuar con mecanismos de red para garantizar la calidad de voz. Por ejemplo, la PBX puede marcar los paquetes RTP con ciertas etiquetas de prioridad (DiffServ/DSCP) para que la OLT y los *switches* den trato preferente a la voz. Asimismo, la PBX se beneficia del alto ancho de banda de la fibra óptica, que permite soportar múltiples llamadas simultáneas de alta calidad (incluso Voz HD), y de la baja latencia de la red GPON. En nuestro despliegue, la PBX está configurada para trabajar juntamente con las VLAN y políticas QoS de la OLT, priorizando el tráfico de voz sobre otros flujos de datos y garantizando así una comunicación clara e ininterrumpida [44].

La implementación de VoIP en una red GPON requiere la configuración adecuada de mecanismos de QoS para asegurar que el tráfico de voz tenga la prioridad necesaria, disminuyendo la latencia, el *jitter* y, de esta forma, garantizando la calidad en la comunicación. Con esta integración, el servicio de VoIP no solo permite una comunicación clara y eficiente, sino que también facilita la gestión centralizada de las llamadas mediante la PBX, lo que es crítico para entornos empresariales y de alta demanda.

En resumen, el servicio VoIP sobre GPON se configura mediante:

- ***Protocolo SIP:*** Para la señalización y establecimiento de llamadas.
- ***Protocolo RTP:*** Para el transporte en tiempo real de audio.
- ***Centralita PBX:*** Como servidor de comunicaciones que gestiona y enruta las llamadas, integrándose con la infraestructura convergente y aplicando políticas QoS que aseguran la calidad del servicio.

De esta manera, la forma de funcionamiento que sigue este servicio es como se muestra en la figura 30

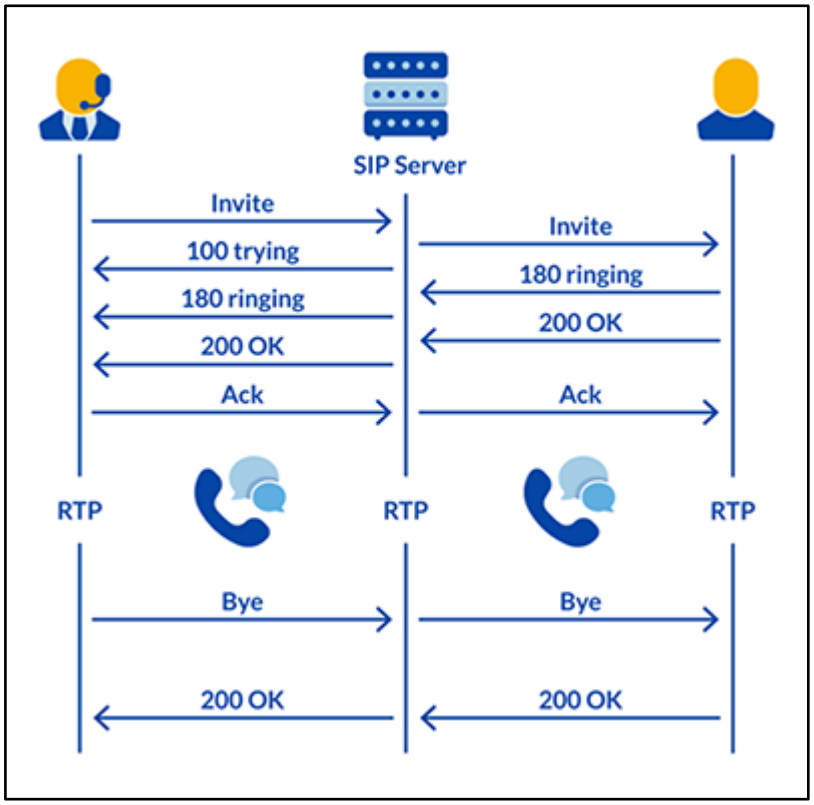


Figura 30. Flujo de comunicación VoIP [70].

Esta solución permite ofrecer un servicio de comunicaciones VoIP robusto y escalable en la red GPON, garantizando tanto la calidad de las llamadas como la eficiencia en el manejo del tráfico de voz.

4.6. Servicio CCTV

4.6.1 Definición

Un Circuito Cerrado de Televisión (CCTV) es un sistema de videovigilancia en el que las señales de vídeo permanecen dentro de un dominio privado y no se difunden públicamente [45] [46]. Sus elementos básicos incluyen cámaras IP, dispositivos de almacenamiento en red (*Network Video Recorder* – NVR) y un sistema de gestión de vídeo (*Video Management System* – VMS) para supervisión y análisis [47].

Cuando este servicio se soporta sobre una red GPON, la alta capacidad de la fibra óptica y su baja latencia permiten transportar flujos de vídeo de alta definición (HD) o ultra alta definición (4K) en tiempo real sin degradación perceptible [48].

4.6.2 Arquitectura y funcionamiento

- ***Captura y codificación**:* Cada cámara IP, generalmente alimentada mediante PoE (*Power over Ethernet*), captura vídeo y lo codifica utilizando estándares de compresión eficientes (H.264/H.265) [49]. Luego envía estos flujos mediante RTP sobre UDP [50].

- ***Transporte hasta la OLT**:* Las cámaras se conectan directamente a la ONT mediante *Ethernet*. La ONT encapsula el tráfico en GEM para transportarlo a través de la fibra óptica hacia la OLT [51].

- ***Concentración**:* La OLT agrega los flujos provenientes de todas las ONT conectadas y los envía centralizadamente al NVR o al centro de monitoreo para su almacenamiento y gestión [51].

- ***Gestión y visualización**:* El sistema VMS permite supervisión en vivo, reproducción histórica de grabaciones y diversas funciones analíticas para potenciar la seguridad [52].

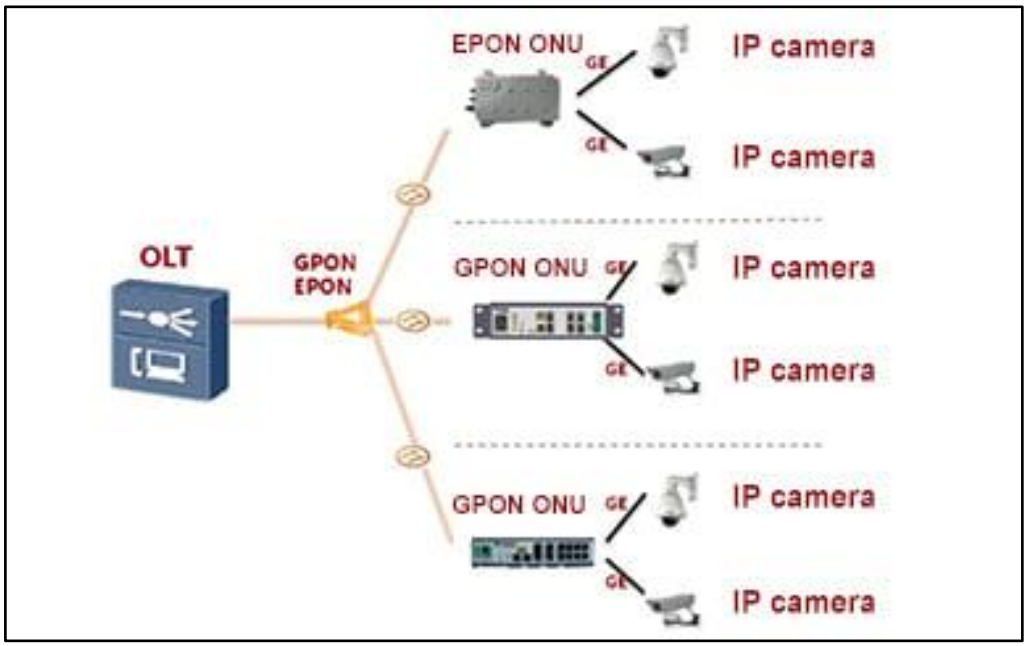


Figura 31. Esquema genérico CCTV-GPON

4.6.3 Integración con la red GPON

La integración del servicio de circuito cerrado de televisión se basó en los siguientes puntos:

- ***Aislamiento mediante VLAN**:* El tráfico CCTV se transporta en la VLAN 400, separada del tráfico de datos, VoIP o IPTV, garantizando un aislamiento efectivo [46].
- ***Priorización del tráfico**:* Tanto la OLT como los *switches* aplican mecanismos avanzados de QoS, utilizando colas de alta prioridad específicamente para los flujos de vídeo, minimizando así la latencia y pérdida de paquetes [46].
- ***Escalabilidad sencilla**:* Añadir nuevas cámaras implica únicamente conectar los dispositivos a puertos disponibles en la ONT o a *switches* PoE locales. La infraestructura GPON tiene capacidad suficiente para absorber canales adicionales sin necesidad de modificaciones complejas en el cableado [53].
- ***Seguridad mejorada**:* El aislamiento mediante VLAN, junto con la autenticación de ONT, garantiza que solo dispositivos autorizados puedan acceder a los flujos de vídeo. El NVR concentra las grabaciones en un entorno centralizado y protegido.

En resumen, integrar CCTV sobre GPON ofrece una solución con centralización eficiente, calidad excepcional del vídeo y facilidad de expansión. Mediante configuraciones flexibles (VLAN y QoS), el operador asegura un servicio robusto y escalable sin necesidad de desplegar infraestructura adicional paralela [53].

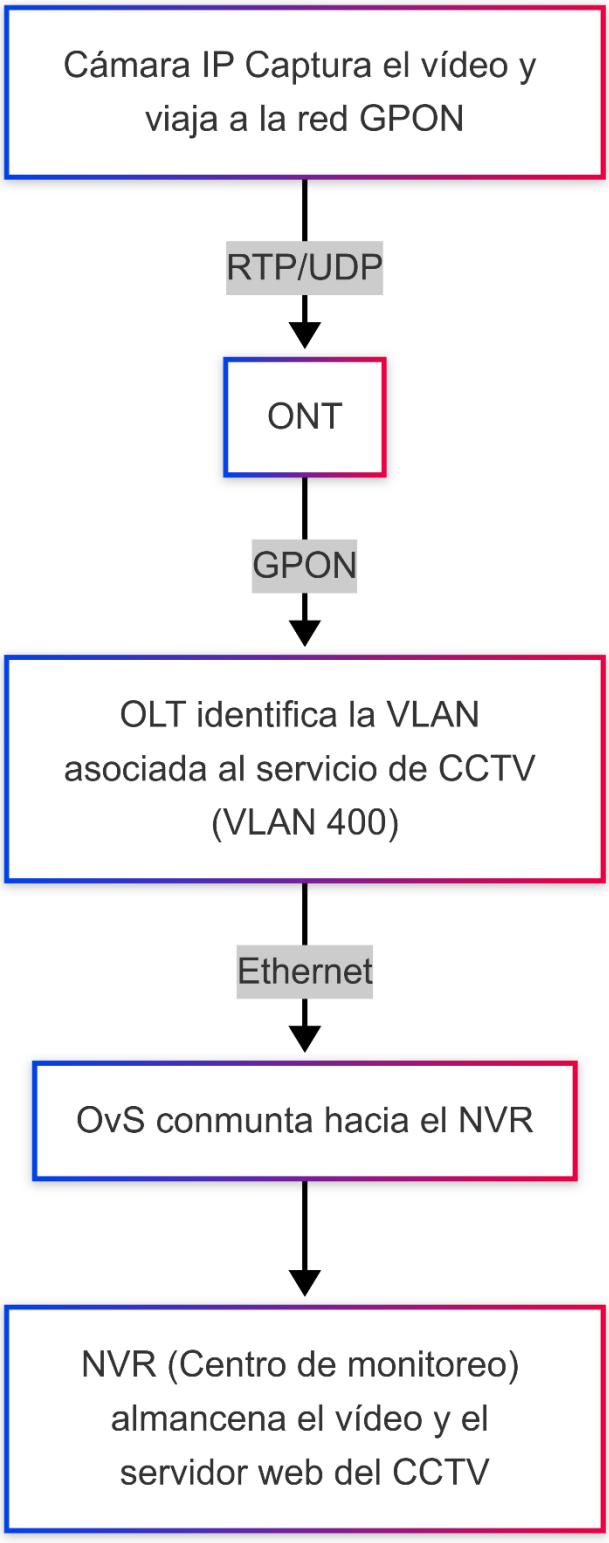


Figura 32. Flujo de comunicación: cámara IP y NVR. Elaboración propia.

Capítulo 5. Implementación del escenario GPON-SDN

5.1. Introducción

En este capítulo se describe de manera detallada el proceso de implementación del escenario GPON-SDN diseñado en este TFG. A partir de los conceptos teóricos y el análisis de requisitos expuestos en capítulos anteriores, se procede a integrar la tecnología de redes ópticas pasivas (GPON) con un plano de control definido por *software* (SDN).

Para ello, se presenta primero la topología de red implementada en el entorno de laboratorio, así como los componentes *hardware* y *software* empleados (OLT, ONTs, controlador SDN y agentes de red). A continuación, se detallan las fases de configuración y puesta en marcha: desde la inicialización de los dispositivos de red, pasando por la parametrización del controlador SDN y la definición de las políticas de reenvió de tráfico, hasta la validación final mediante pruebas de rendimiento y funcionamiento.

La organización de este capítulo es la siguiente:

- Descripción de los componentes de la red.
- Configuración de la red GPON.
- Configuración del *Open virtual Switch*.
- Configuración de los servicios virtualizados.
- Integración del controlador SDN.

5.2. Descripción de los componentes de la red

Los componentes utilizados para la implementación de la red GPON-SDN son los siguientes:

- Una *SmartOLT 240*.
- Dos *Routers-ONT WaveAccess 4520*.
- Tres ordenadores de sobremesa.

Una vez nombrados los componentes principales de nuestra red, pasaremos a hacer una breve descripción sobre el rol que toma cada uno de ellos en nuestra red.

En primer lugar, tenemos la *SmartOLT 240* que interconecta los servicios virtualizados con la red GPON establecida, siendo un componente principal de nuestra red, debido a que es el encargado de reenviar el tráfico a los distintos usuarios de la red teniendo en cuenta el etiquetado VLAN de cada trama que llega a este equipo.

En segundo lugar, tenemos a los dos *Routers-ONT WaveAccess 4520*, representando a los dos usuarios que están dados de alta en la red, y con la capacidad de ofrecer los distintos servicios que tengan contratados cada uno como veremos más adelante.

En tercer lugar, dos ordenadores de sobremesa que contaran con clientes (*software*) específicos para poder acceder a cada servicio que tenga contratado el usuario en cuestión.

En cuarto lugar, un ordenador de sobremesa ubicado del lado del servidor, que integra los distintitos servicios virtualizados, mediante el *Open Virtual Switch*, encargado de conmutar las tramas provenientes de cada servidor hacia la *Smart OLT 240* mencionada anteriormente.

Para un mayor entendimiento sobre la arquitectura de la red desarrollado, véase el siguiente esquema de red:

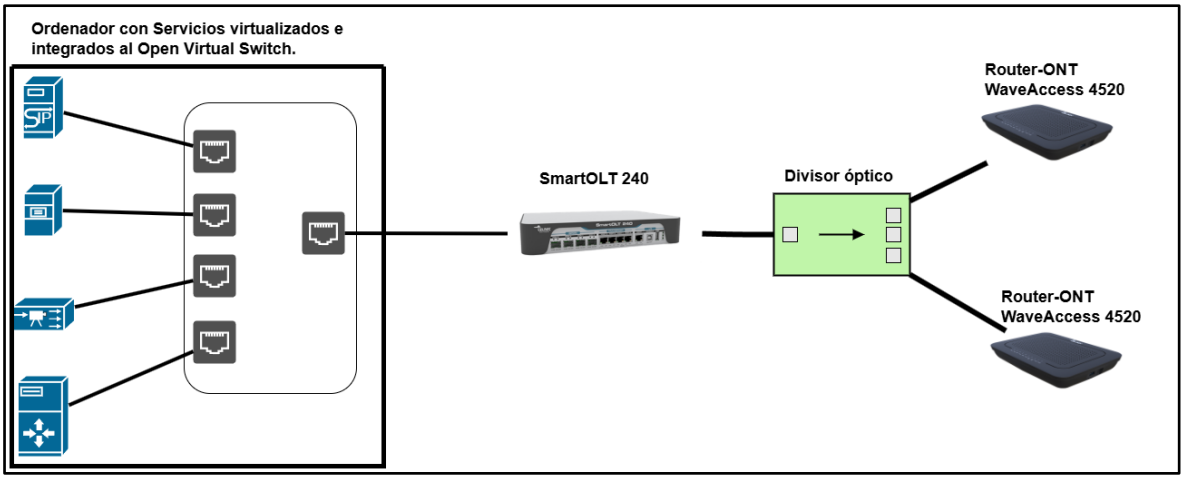


Figura 33. Esquema general de la red GPON implementada. Elaboración propia.

En la Figura 33 se presenta, de manera esquemática, la topología de red diseñada e implementada en este Trabajo de Fin de Grado. En el extremo izquierdo, previo a la SmartOLT 240, se ubica el servidor, constituida por un único ordenador de sobremesa que alberga:

1. Servicios virtualizados de voz, vídeo, datos y CCTV, desplegados en máquinas virtuales independientes.

2. *Open vSwitch* (OvS), que actúa como capa de conmutación virtual y gestiona el encaminamiento interno entre las instancias de servicio.

3. Funciones de red adicionales, integradas en el mismo equipo para centralizar la orquestación y facilitar el mantenimiento.

Cada máquina virtual se conecta al OvS, lo que permite una conmutación eficiente de los distintos flujos de tráfico hacia la *SmartOLT 240*.

La *SmartOLT 240* cumple la función de multiplexor y distribuidor óptico:

- ***Multiplexación*:** combina los flujos de voz, vídeo, datos y CCTV en una única señal óptica de banda ancha.

- ***Transmisión*:** envía dicha señal a través de fibra hasta un divisor óptico.

- ***Distribución*:** el divisor atenúa y reparte uniformemente la señal óptica, garantizando que cada Terminal de Red Óptica (ONT *WaveAccess 4520*), situadas en el extremo derecho del diagrama, reciba la potencia necesaria para ofrecer al usuario final los servicios con la calidad y el nivel de servicio previstos.

Después de haber realizado una descripción esquemática, pasamos a ver el montaje de la maqueta GPON de manera física como se ve en la figura 34.

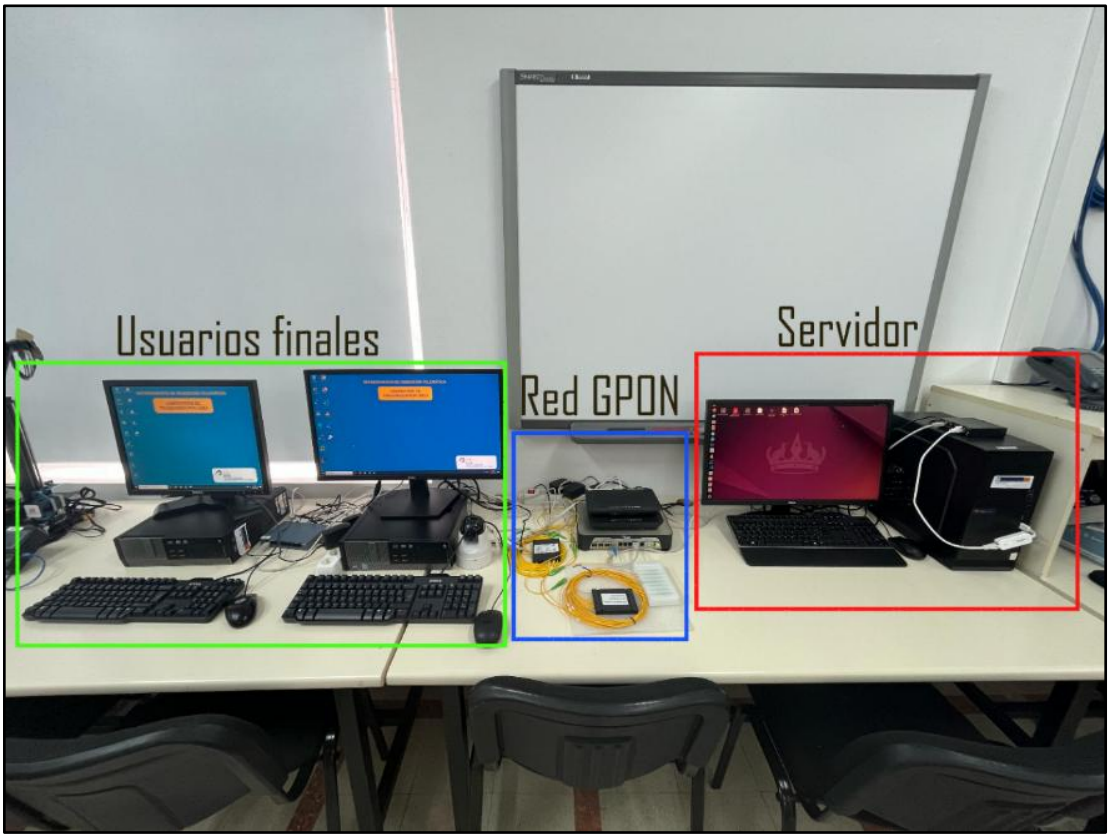


Figura 34. Maqueta física GPON. Elaboración propia.

Como se puede observar en la figura 34, la maqueta de la red se divide en tres sectores o tramos:

- ***Servidor:*** conectado a la red GPON a través de la interfaz física (enp6s0) de red del ordenador de sobremesa. Figura 35.

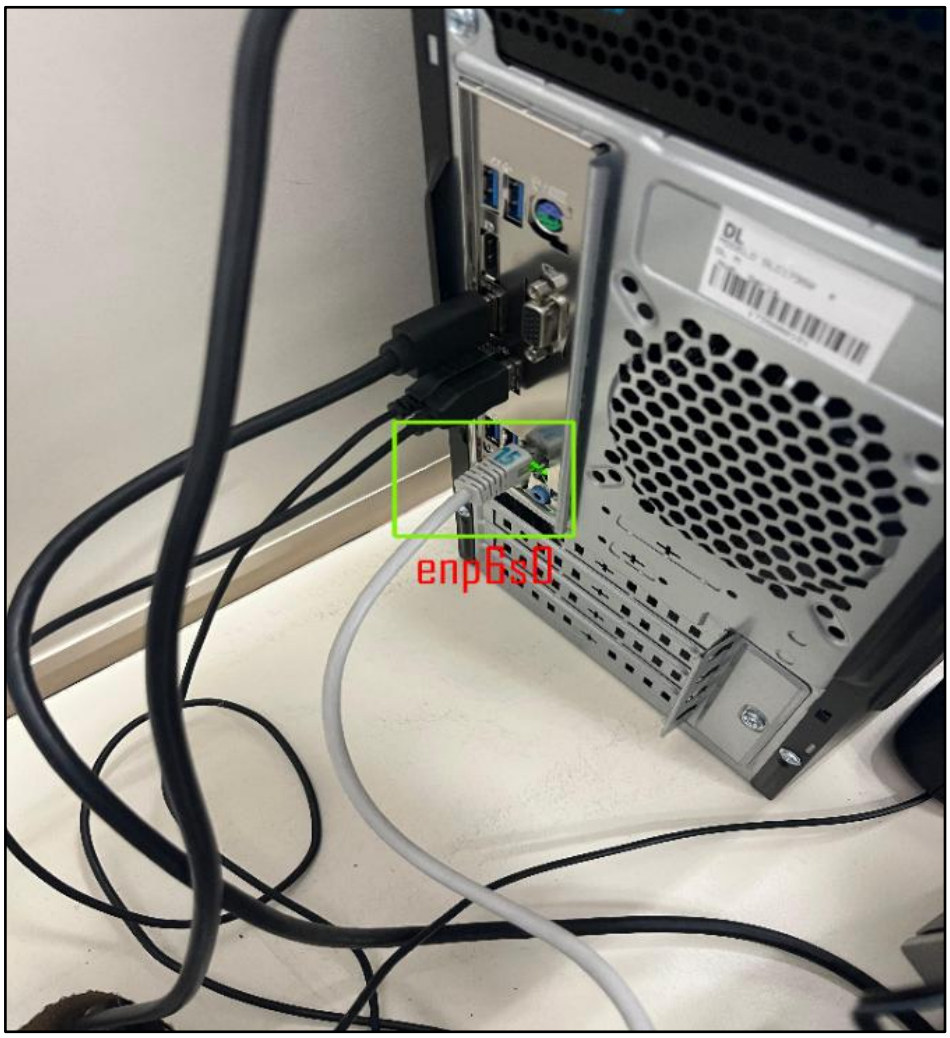


Figura 35. Interfaz de red física de salida desde el Servidor hacia la OLT. Elaboración propia.

- ***Red GPON:*** interconexión entre el servidor y los usuarios finales. La interfaz *enp6s0* del servidor se conecta a la interfaz de transporte *PORT 0* de la OLT, y esta a su vez, a través de un enlace de fibra óptica por la interfaz *PON 0* de la OLT cada ONT correspondiente a cada usuario. Figura 36.

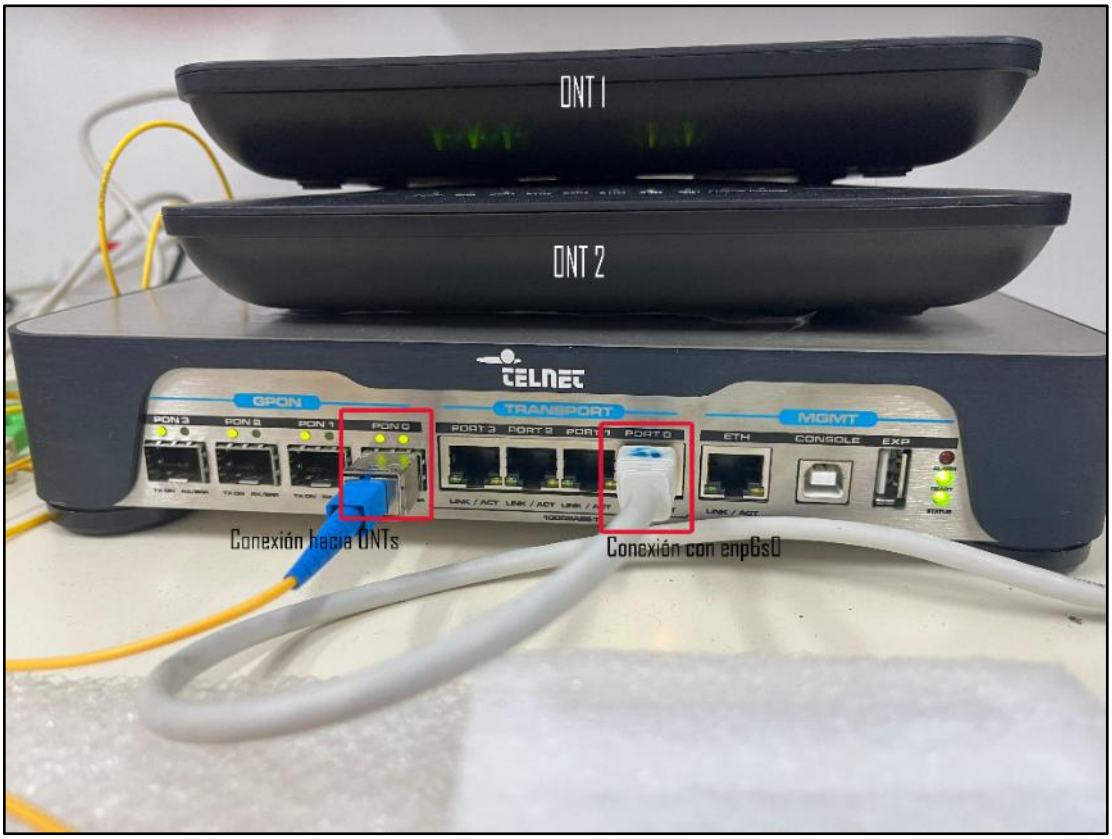


Figura 36. Equipos GPON: OLT, ONT 1, ONT 2. Elaboración propia.

- ***Usuarios finales:*** abonados a la red, subscritores de los servicios ofrecidos en la misma. Figura 37.

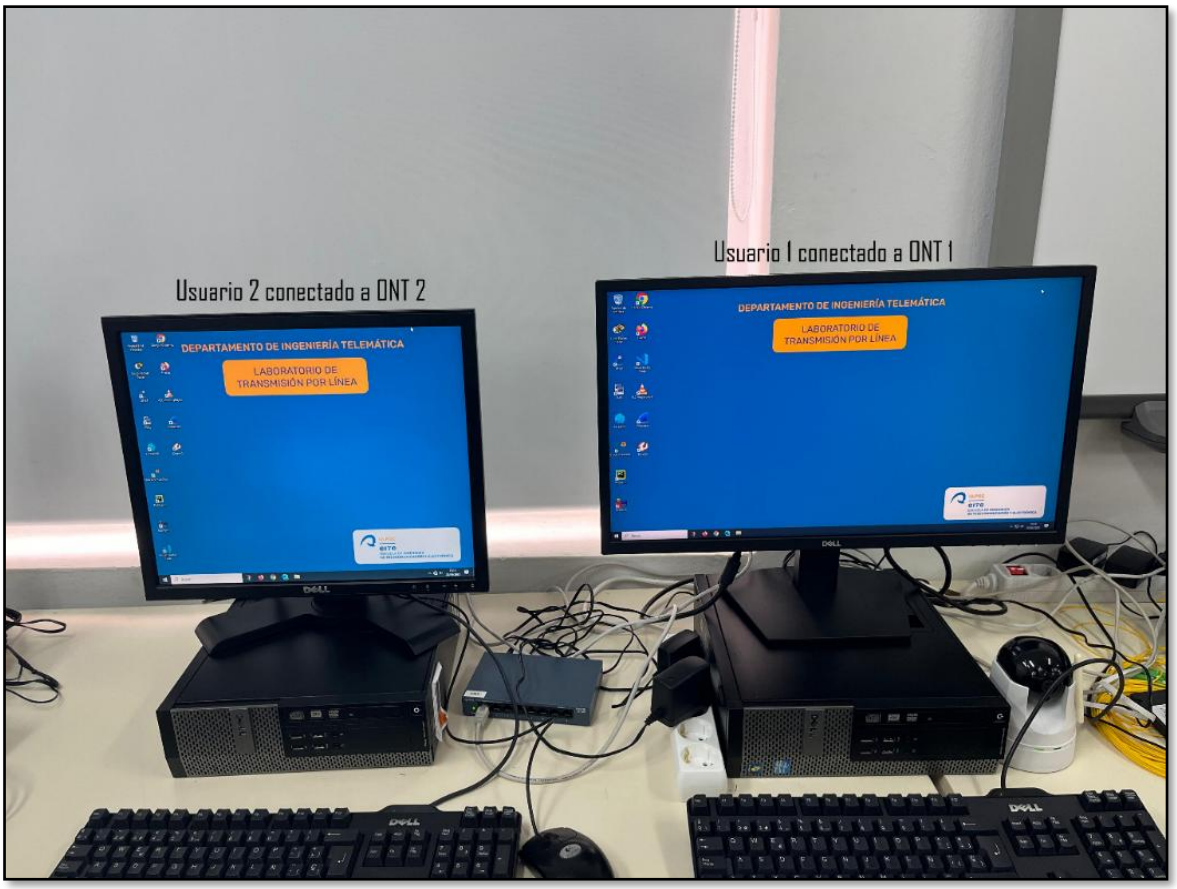


Figura 37. Usuarios finales (Abonados a la red GPON). Elaboración propia.

5.3. Configuración de la red GPON

En este apartado se describirá la conexión e interconexión de cada componente de la red. Empezando por la OLT:

Para realizar la configuración de este equipo, hicimos uso del *Telnet GPON Management System* (TGMS), desarrollado por el fabricante Telnet, es un *software* que permite al administrador configurar los servicios y parámetros de la red GPON mediante la siguiente interfaz gráfica:

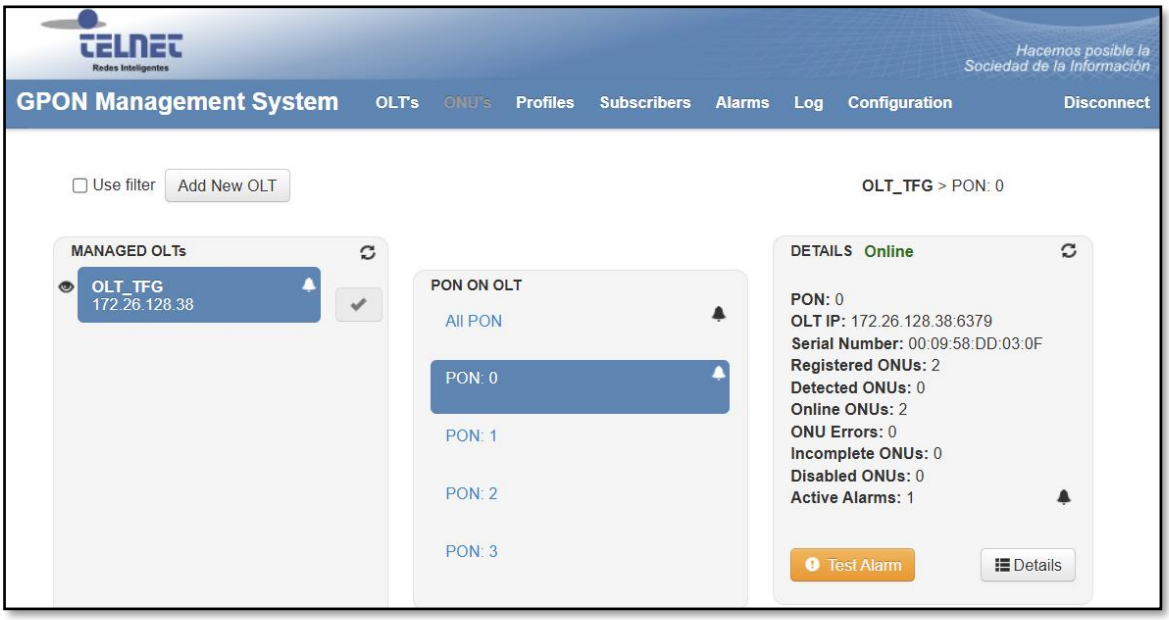


Figura 38. Menú principal del TGMS. Elaboración propia.

Como podemos observar en la figura 38, tenemos que en la parte de la izquierda la OLT que estamos utilizando está dada de alta como “OLT_TFG”, en la parte central conectada al puerto PON 0 y en la parte de la derecha un resumen de la configuración implementada en la OLT.

5.3.1 Alta de las ONTs

Antes de empezar con la configuración de los servicios, hay que dar de alta a las dos ONT’s que vamos a utilizar en nuestra red GPON.

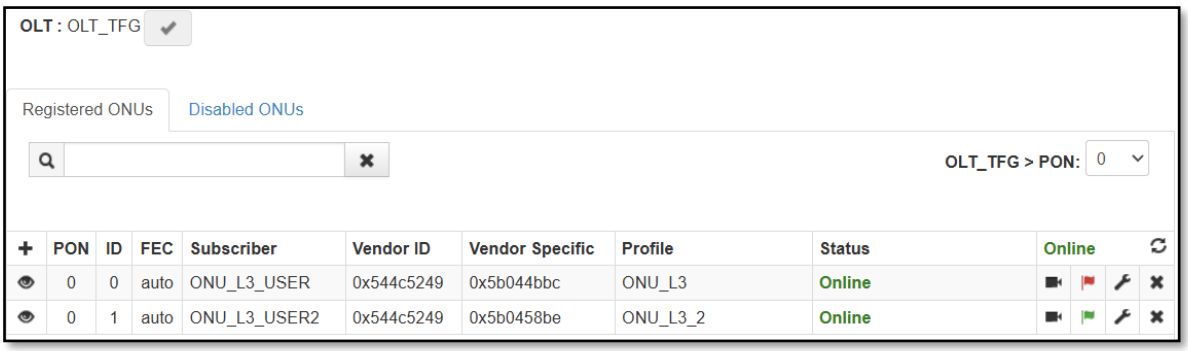


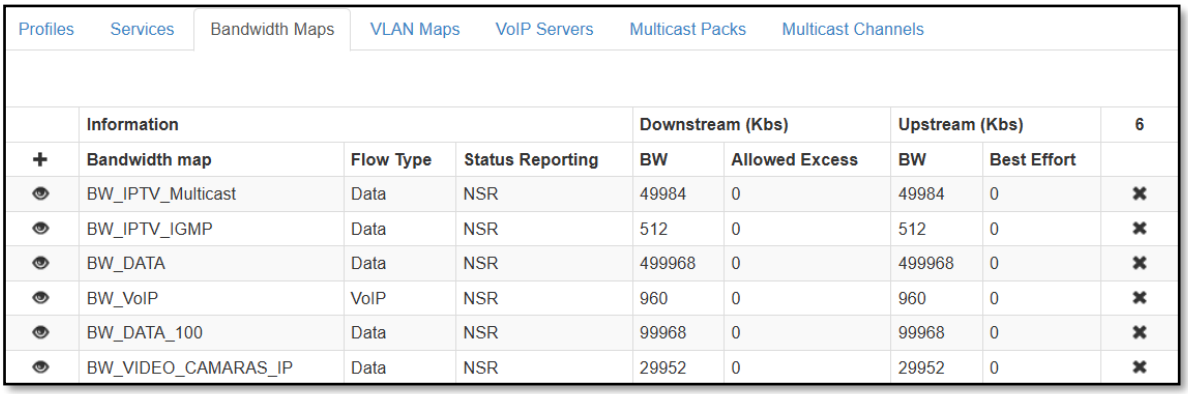
Figura 39. Parámetros configurados de cada ONT. Elaboración propia.

En la figura 39, podemos observar los distintos parámetros que componen la configuración de las dos ONTs utilizadas en la red. Representando cada uno de ellos lo siguiente:

- ***PON:*** puerto PON de la OLT al que se encuentran conectadas las ONT's.
- ***ID:*** identificación autoincrementa de la ONT dada de alta.
- ***FEC:*** mecanismo de corrección de errores con técnicas de redundancia.
- ***Subscriber:*** Usuario final conectado a la ONT correspondiente.
- ***Vendor ID:*** identificador del fabricante.
- ***Vendor Specific:*** últimos 4 octetos de la dirección MAC de cada ONT.
- ***Profile:*** descripción del perfil de cada ONT.
- ***Status:*** Estado en el que se encuentra la ONT.

5.3.2 Configuración de los mapas de ancho de banda

Posteriormente, se deben configurar los mapas de ancho de banda y los mapas de VLAN, los cuales se encargan, respectivamente, de limitar la cantidad máxima de información a transmitir y de etiquetar las tramas con el número de VLAN correspondiente al servicio específico.



	Information			Downstream (Kbs)		Upstream (Kbs)		6
+	Bandwidth map	Flow Type	Status Reporting	BW	Allowed Excess	BW	Best Effort	
	BW_IPTV_Multicast	Data	NSR	49984	0	49984	0	
	BW_IPTV_IGMP	Data	NSR	512	0	512	0	
	BW_DATA	Data	NSR	499968	0	499968	0	
	BW_VoIP	VoIP	NSR	960	0	960	0	
	BW_DATA_100	Data	NSR	99968	0	99968	0	
	BW_VIDEO_CAMARAS_IP	Data	NSR	29952	0	29952	0	

Figura 40. Mapas de ancho de banda. Elaboración propia.

La Figura 40 muestra la pestaña *"Bandwidth Maps"* de la interfaz de gestión de red, donde se listan los distintos perfiles de ancho de banda definidos. Para cada mapa aparece:

Bandwidth map: Indica el nombre del perfil de ancho de banda configurado, asociado específicamente a distintos tipos de servicios, tales como IPTV *Multicast*, datos, VoIP o cámaras IP.

Flow Type: Especifica la categoría del tráfico gestionado por cada perfil. Los tipos principales son Data (tráfico general), VoIP (voz sobre IP) e IGMP (tráfico de gestión para grupos multicast).

Status Reporting: Define el modo en que se reporta el estado del tráfico. En este caso, "NSR" (*No Status Reporting*) indica que no se realiza un reporte activo del estado del flujo, simplificando la gestión.

Downstream: Representa la tasa máxima permitida de datos desde la red hacia el usuario (descendente o *downstream*), expresada en *kilobits* por segundo (Kbps).

Downstream (Allowed Excess): Indica la cantidad adicional de tráfico permitida que puede superar temporalmente la tasa máxima configurada. En los perfiles mostrados está configurada a 0, indicando que no se permite exceso de tráfico.

Upstream: Indica la tasa máxima permitida para el tráfico enviado desde el usuario hacia la red.

Upstream (Best Effort): Refleja una configuración adicional para tráfico ascendente en modo "mejor esfuerzo".

En conjunto, este listado facilita la visualización rápida de los límites de tráfico aplicados a cada servicio, asegurando que no se sobrepase la capacidad establecida.

5.3.3 Configuración de los mapas de VLAN

Asociado a los mapas de ancho de banda se tienen que definir los mapas de VLAN, como se muestra en la figura siguiente.

VLAN	User - Priority	User - Tag	C - Priority	C - Tag	S - Priority	S - Tag
CCTV	Any	1400	Copy	400	Untagged	Untagged
DATA	Any	1100	Copy	100	Untagged	Untagged
IPTV	Any	1300	Copy	300	Untagged	Untagged
VoIP	Any	1200	Copy	200	Untagged	Untagged

Figura 41. Configuración de los mapas de VLAN. Elaboración propia.

Como se mencionó anteriormente, en la Figura 41 se presenta la configuración de los Mapas de VLAN en el dispositivo de red, donde se definen cuatro dominios lógicos para segregar el tráfico de CCTV, datos, IPTV y VoIP. Cada entrada permite aceptar cualquier prioridad entrante, copiar la etiqueta y la prioridad internas, y, finalmente, enviar las tramas sin etiqueta VLAN al enlace ascendente.

VLAN: Identificador del perfil asociado al servicio específico (CCTV, Datos, IPTV o VoIP). Este perfil agrupa las reglas de etiquetado VLAN necesarias para gestionar correctamente cada servicio en la infraestructura GPON.

User - Priority: Nivel de prioridad del tráfico asignado por el usuario en la interfaz UNI (interfaz del usuario final). Al aparecer "*Any*" se acepta cualquier prioridad asignada desde los equipos del usuario.

User - Tag: Es la etiqueta VLAN que aplica la ONU al tráfico saliente en la interfaz UNI hacia el equipo del usuario. Este valor debe coincidir con la configuración VLAN que espera el equipo conectado a la ONU, especialmente relevante cuando la ONU está en modo router (nivel 3) y se conecta a dispositivos que trabajan con VLANs.

C - Priority (Customer Priority): Prioridad del tráfico en la red GPON (entre la ONU y la OLT). La indicación "*Copy*" implica que la prioridad asignada por el usuario (*User-Priority*) se conserva y utiliza directamente en la red GPON sin modificaciones.

C - Tag (Customer Tag): Es la etiqueta principal utilizada para identificar y diferenciar los servicios en la red GPON, entre la ONU y la OLT. Este valor siempre debe especificarse explícitamente, ya que la OLT lo utiliza para enrutar correctamente cada tipo de servicio. En la práctica, debe coincidir con el valor del *User-Tag* para asegurar la correcta transferencia del tráfico etiquetado entre la red GPON y la interfaz UNI de la ONU.

S - Priority (Service Priority): Nivel de prioridad asignado al tráfico en la red de transporte (red troncal), donde circula el tráfico etiquetado con *S-tag* en caso de que exista doble etiquetado (QinQ). La opción "*Untagged*" indica que en esta configuración particular no se está utilizando doble etiquetado, por lo que no hay prioridad adicional aplicada en esta capa.

S - Tag (Service Tag): Etiqueta VLAN adicional para redes que utilizan doble etiquetado (QinQ) en la red de transporte. Diferencia a los distintos proveedores de servicios. La configuración "*Untagged*" señala que en este caso específico no se aplica esta etiqueta adicional, utilizándose únicamente un etiquetado simple (*C-tag*).

La configuración correcta de las diferentes VLAN en esta solución GPON requiere definir con precisión tres tipos de etiquetas: la etiqueta de proveedor de servicio (*S-Tag*), la etiqueta del cliente (*C-Tag*) y la etiqueta del usuario (*User-Tag*). En el tráfico descendente, proveniente de la red de transporte hacia la red GPON, la OLT elimina la *S-Tag* si se utiliza doble etiquetado (QinQ), dejando únicamente la *C-Tag* para identificar y clasificar el

servicio específico (Datos, IPTV, VoIP, CCTV, etc.). Al recibir estas tramas etiquetadas con *C-Tag*, la ONU debe reenviarlas hacia el usuario final mediante su interfaz UNI, aplicando la etiqueta correspondiente *User-Tag* que debe coincidir con la VLAN configurada en el equipo conectado. Por otro lado, en el tráfico ascendente (desde la interfaz UNI del usuario hacia la OLT), la ONU recibe el tráfico etiquetado con *User-Tag* por parte del equipo del usuario, traduciéndolo directamente al *C-Tag* correspondiente para que la OLT reconozca y enrute correctamente el servicio específico hacia la red de transporte.

Figura 42. Flujo genérico del tratamiento de las etiquetas VLAN de la red GPON. Elaboración propia.

A manera de ejemplo, tenemos el siguiente diagrama de flujo, en el que se representa el paso de las tramas para el servicio de datos:

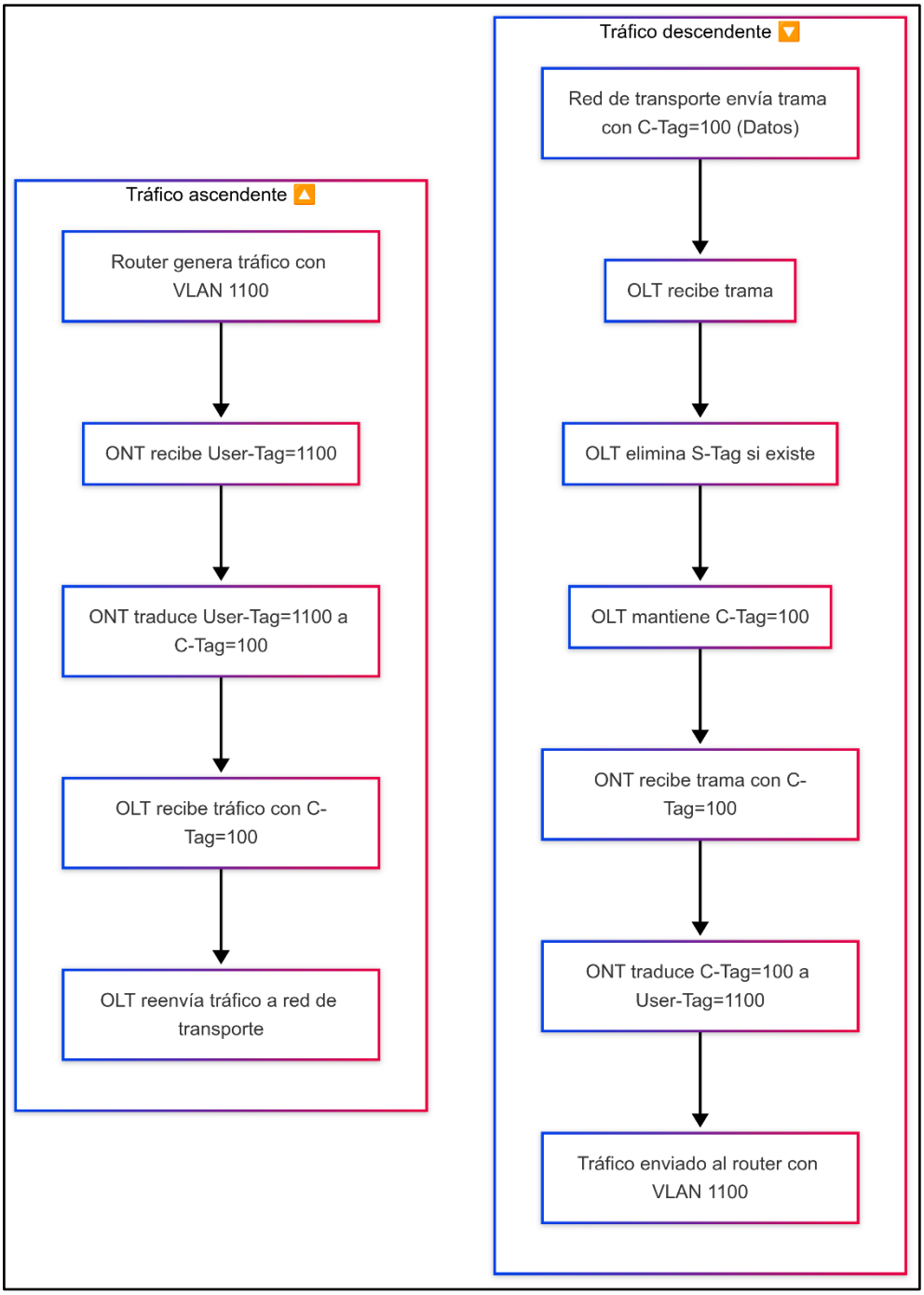
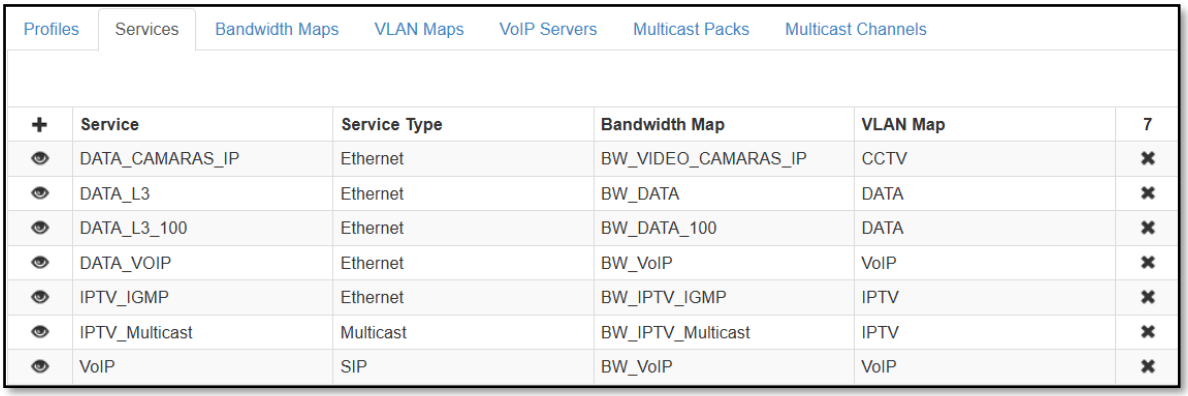


Figura 43. Flujo seguido por el tráfico de datos en la red GPON implementada. Elaboración propia.

5.3.4 Configuración de servicios en la OLT

Una vez finalizada la configuración de los mapas de ancho de banda, y los mapas de VLAN, se pasó a crear cada servicio como se muestra en la figura 44.



+	Service	Service Type	Bandwidth Map	VLAN Map	7
	DATA_CAMARAS_IP	Ethernet	BW_VIDEO_CAMARAS_IP	CCTV	
	DATA_L3	Ethernet	BW_DATA	DATA	
	DATA_L3_100	Ethernet	BW_DATA_100	DATA	
	DATA_VOIP	Ethernet	BW_VoIP	VoIP	
	IPTV_IGMP	Ethernet	BW_IPTV_IGMP	IPTV	
	IPTV_Multicast	Multicast	BW_IPTV_Multicast	IPTV	
	VoIP	SIP	BW_VoIP	VoIP	

Figura 44. Servicios creados. Elaboración propia.

En la Figura 44 se detallan los servicios creados en la configuración del sistema, cada uno asociado a un tipo de tráfico específico, un mapa de ancho de banda (*Bandwidth Map*) y un mapa de VLAN (*VLAN Map*). Esta relación es fundamental para garantizar la correcta asignación de recursos y la segregación lógica del tráfico dentro de la red GPON.

- ***DATA_CAMARAS_IP***
  - ***Service Type:*** *Ethernet*
  - ***Bandwidth Map:*** BW_VIDEO_CAMARAS_IP
  - ***VLAN Map:*** CCTV → indica que este servicio se asigna al tráfico de videovigilancia.
- ***DATA_L3***
  - ***Service Type:*** *Ethernet*
  - ***Bandwidth Map:*** BW_DATA
  - ***VLAN Map:*** DATA → tráfico de datos generales, sin segmentación adicional.
- ***DATA_L3_100***
  - ***Service Type:*** *Ethernet*

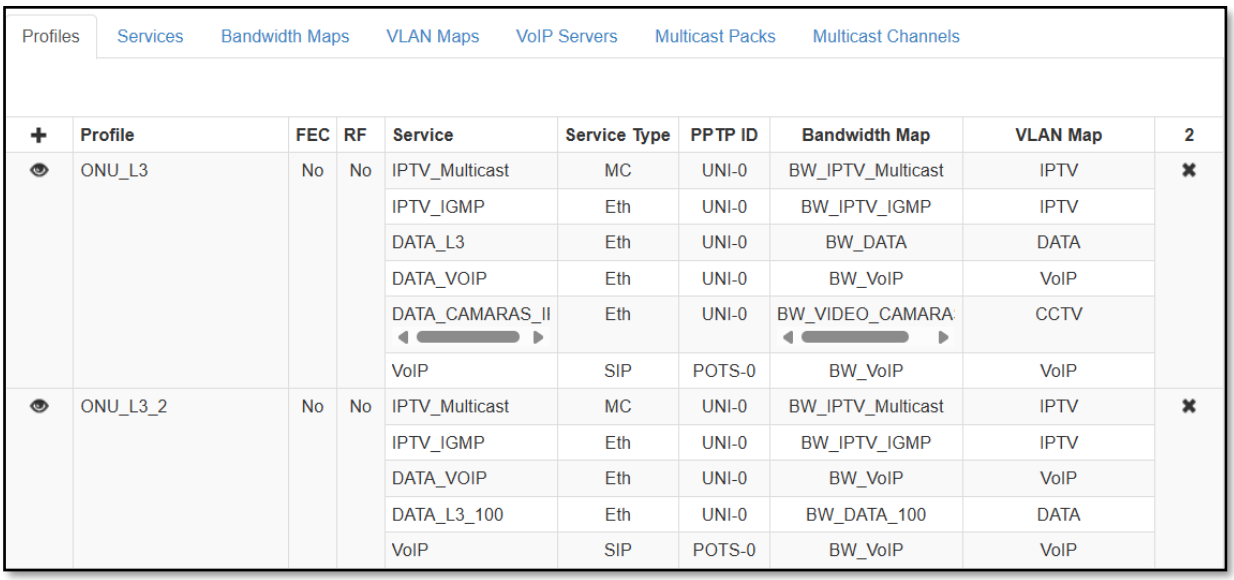
- ***Bandwidth Map:*** BW_DATA_100
- ***VLAN Map:*** DATA → mismo dominio de tráfico que el anterior, pero con diferente política de ancho de banda.

- ***DATA_VOIP***
  - ***Service Type:*** *Ethernet*
  - ***Bandwidth Map:*** BW_VoIP
  - ***VLAN Map:*** VoIP → tráfico de voz encapsulado en tramas *Ethernet*.
- ***IPTV_IGMP***
  - ***Service Type:*** *Ethernet*
  - ***Bandwidth Map:*** BW_IPTV_IGMP
  - ***VLAN Map:*** IPTV → canal de control para gestión de suscripciones *multicast* mediante IGMP.
- ***IPTV_Multicast***
  - ***Service Type:*** *Multicast*
  - ***Bandwidth Map:*** BW_IPTV_*Multicast*
  - ***VLAN Map:*** IPTV → canal de datos *multicast* correspondiente a los flujos de vídeo.
- ***VoIP***
  - ***Service Type:*** SIP
  - ***Bandwidth Map:*** BW_VoIP
  - ***VLAN Map:*** VoIP → servicio orientado a telefonía, con señalización y transporte mediante el protocolo SIP.

Esta configuración permite una gestión diferenciada del tráfico según su naturaleza, no solo a nivel lógico (VLAN), sino también en términos de calidad de servicio, ya que cada servicio cuenta con un mapa de ancho de banda específico que regula su uso de recursos.

5.3.5 Configuración de los perfiles

Finalmente, en la Figura 45 se muestran los perfiles de ONT definidos en la red GPON, cada uno agrupando varios servicios sobre su interfaz PON con parámetros de FEC, RF, identificador de PTP, mapa de ancho de banda y etiqueta VLAN. Esta distribución permite asignar de forma coherente los recursos y aplicar políticas diferenciadas según el tipo de usuario.



Profiles | Services | Bandwidth Maps | VLAN Maps | VoIP Servers | Multicast Packs | Multicast Channels

+	Profile	FEC	RF	Service	Service Type	PPTP ID	Bandwidth Map	VLAN Map	2
👁	ONU_L3	No	No	IPTV_Multicast	MC	UNI-0	BW_IPTV_Multicast	IPTV	✖
				IPTV_IGMP	Eth	UNI-0	BW_IPTV_IGMP	IPTV	
				DATA_L3	Eth	UNI-0	BW_DATA	DATA	
				DATA_VOIP	Eth	UNI-0	BW_VoIP	VoIP	
				DATA_CAMARAS_II	Eth	UNI-0	BW_VIDEO_CAMARA	CCTV	
				VoIP	SIP	POTS-0	BW_VoIP	VoIP	
👁	ONU_L3_2	No	No	IPTV_Multicast	MC	UNI-0	BW_IPTV_Multicast	IPTV	✖
				IPTV_IGMP	Eth	UNI-0	BW_IPTV_IGMP	IPTV	
				DATA_VOIP	Eth	UNI-0	BW_VoIP	VoIP	
				DATA_L3_100	Eth	UNI-0	BW_DATA_100	DATA	
				VoIP	SIP	POTS-0	BW_VoIP	VoIP	

Figura 45. Creación de perfiles: Asociación de los servicios con los Abonados de la red. Elaboración propia.

En la configuración de los perfiles, dos parámetros cobran especial relevancia:

- ***ServiceType***
  Define la naturaleza del servicio que entrega el ONT y determina cómo se procesan las tramas:
  - ***Eth (Ethernet):*** para datos genéricos, IPTV_IGMP o CCTV sobre VLAN.
  - ***MC (Multicast):*** para vídeo *multicast* nativo, gestionado directamente en capa 2.
  - ***SIP / POTS-0:*** para VoIP, separando señalización (SIP) y, si existe, voz analógica sobre POTS.
- ***PTPID***
  Es el identificador de "*Physical Termination Point*" donde se asigna cada servicio:

- ***UNI-0:*** interfaz *Ethernet* principal del suscriptor.
- ***POTS-0:*** canal analógico POTS dedicado a telefonía. Gracias al PTP ID, el sistema sabe en qué puerto físico o virtual debe habilitar cada servicio.

5.3.6 Configuración del servidor de VoIP y IPTV

Para el servicio de VoIP tenemos que definir los siguientes parámetros:

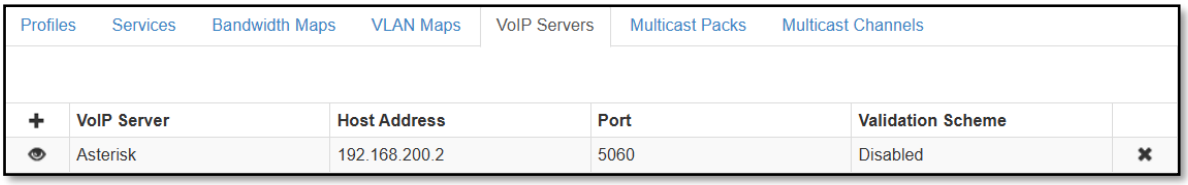


Figura 46. Configuración del servidor de VoIP. Elaboración propia.

En la Figura 46 se observa la configuración establecida, donde se ha creado una entrada identificada como *Asterisk*, que actúa como servidor VoIP.

Este servidor se encuentra ubicado en la dirección IP *192.168.200.2*, utilizando el puerto *5060*, que es el estándar para tráfico SIP. Además, el campo *Validation Scheme* se encuentra deshabilitado (*Disabled*), lo que implica que no se aplica ningún mecanismo de autenticación a nivel de la ONT, delegando así todo el control de acceso al propio servidor *Asterisk*.

Gracias a esta configuración, las ONT's pueden registrar sus terminales VoIP de forma transparente y encaminar correctamente las llamadas a través del servidor central.

En la Figura 47 se muestra ahora la pestaña *Multicast Channels*, donde se define el canal que posteriormente se incluirá en el paquete de multidifusión. En este caso, aparece un único canal:

- ***Channel:*** PRUEBA
- ***Start / End:*** *239.1.1.1*

Aquí PRUEBA actúa como identificador lógico del flujo *multicast*, y al usar la misma dirección IP en los campos *Start* y *End*, se establece un único grupo de multidifusión en la dirección *239.1.1.1*. De este modo, cuando la ONT se suscribe al paquete PAQUETE_PRUEBA, recibe correctamente los paquetes de vídeo dirigidos a dicho canal sin necesidad de configurar rangos adicionales.

Figura 47.Configuración del canal multicast. Elaboración propia.

Posteriormente en la Figura 48 se presenta la configuración del servicio de IPTV en el apartado de *Multicast Packs*, donde se define un paquete de canales *multicast*. Esta configuración es fundamental para permitir la distribución de contenido televisivo a través de direcciones IP de multidifusión, que pueden ser gestionadas y suscritas desde las ONT mediante IGMP.

En este caso, se ha creado un paquete denominado PAQUETE_PRUEBA, que contiene un canal llamado PRUEBA. Tanto la IP de inicio (*IP Start*) como la IP de fin (*IP End*) están establecidas en *239.1.1.1*, lo que indica que el canal se transmite en dicha dirección *multicast*. Esto permite que el tráfico sea correctamente reconocido y encaminado dentro de la red, garantizando que las ONTs suscritas al canal puedan recibir el contenido sin interferencias ni necesidad de asignaciones individuales.

Esta estructura es especialmente útil en redes GPON, donde el uso de direcciones *multicast* optimiza el ancho de banda, permitiendo que un único flujo de vídeo sea compartido por múltiples usuarios simultáneamente.

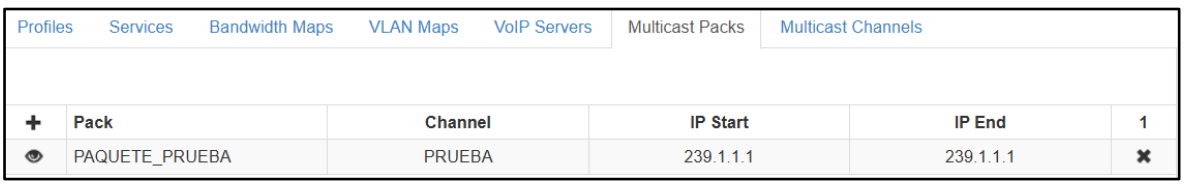
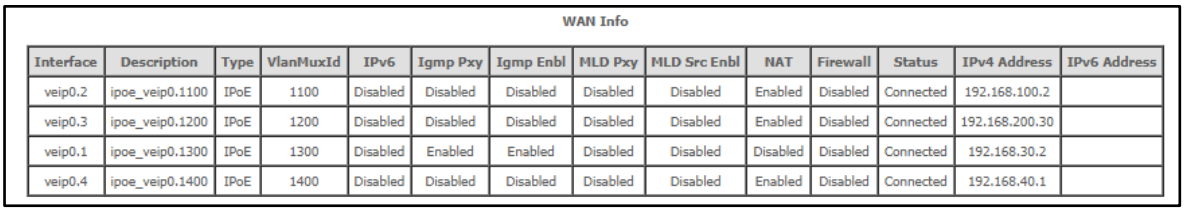


Figura 48. Configuración del paquete multicast. Elaboración propia.

5.3.7 Configuración ONT 1

Una vez configurados los distintos parámetros de la OLT para proporcionar los servicios correspondientes a los usuarios finales, resulta necesario ajustar la configuración de cada ONT para garantizar una sincronización óptima y la correcta recepción de la información. A continuación, se detallan los parámetros que deben establecerse en la ONT 1:



WAN Info

Interface	Description	Type	VlanMuxId	IPv6	Igmp Pxy	Igmp Enbl	MLD Pxy	MLD Src Enbl	NAT	Firewall	Status	IPv4 Address	IPv6 Address
veip0.2	ipoe_veip0.1100	IPoE	1100	Disabled	Disabled	Disabled	Disabled	Disabled	Enabled	Disabled	Connected	192.168.100.2	
veip0.3	ipoe_veip0.1200	IPoE	1200	Disabled	Disabled	Disabled	Disabled	Disabled	Enabled	Disabled	Connected	192.168.200.30	
veip0.1	ipoe_veip0.1300	IPoE	1300	Disabled	Enabled	Enabled	Disabled	Disabled	Disabled	Disabled	Connected	192.168.30.2	
veip0.4	ipoe_veip0.1400	IPoE	1400	Disabled	Disabled	Disabled	Disabled	Disabled	Enabled	Disabled	Connected	192.168.40.1	

Figura 49. Configuración WAN de la ONT 1. Elaboración propia.

La figura 49 muestra un resumen de los distintos parámetros configurados, correspondientes a cada servicio ofrecido en la red.

para el servicio de IPTV (interfaz *veip0.1*) se utilizan los siguientes valores específicos:

- ***VlanMuxID:*** se asigna la VLAN 1300 en la red troncal para el tráfico de vídeo.
- ***IGMP Proxy:*** ambos deben activarse para que la ONT gestione correctamente las suscripciones *multicast* del receptor IPTV.
- ***MLD Proxy / MLD Src Enable:*** no se emplea *multicast* sobre IPv6.
- ***NAT:*** la ONT traduce direcciones y recibe la IP 192.168.30.2 en este perfil.

Para el servicio de datos (interfaz *veip0.2*) se emplean los siguientes valores específicos:

- ***VlanMuxID:*** se asocia la VLAN 1100 al tráfico de datos de usuario.
- ***IGMP Proxy:*** al no ser un servicio *multicast*, se mantiene la gestión IGMP desactivada.
- ***MLD Proxy / MLD Src Enable:*** no aplica *multicast* IPv6.
- ***NAT***: la ONT realiza NAT para el tráfico de datos y recibe la dirección 192.168.100.2 en este perfil.

Para el servicio de VoIP (interfaz *veip0.3*) se emplean los siguientes valores específicos:

- ***VlanMuxID:*** se asigna la VLAN 1200 al tráfico de señalización y voz sobre IP.
- ***IGMP Proxy:*** como no hay tráfico *multicast*, estos parámetros permanecen desactivados.
- ***MLD Proxy / MLD Src Enable:*** no se utiliza *multicast* IPv6.
- ***NAT:*** se habilita la traducción de direcciones y se asigna la IP 192.168.200.30 para este perfil.

para el servicio de CCTV (interfaz *veip0.4*) se emplean los siguientes valores específicos:

- ***VlanMuxID:*** se asigna la VLAN 1400 al tráfico de cámaras y vídeo de vigilancia.
- ***IGMP Proxy:*** al no requerir gestión *multicast*, estos parámetros permanecen desactivados.
- ***MLD Proxy / MLD Src Enable***: no se utiliza *multicast* sobre IPv6.
- ***NAT:*** se habilita la traducción de direcciones y se asigna la IP 192.168.40.1 para este perfil.

La relación de IP y VLAN para la ONT 1 es la que se muestra en la siguiente tabla:

Tabla 9. Relación entre las VLANs-IPs-Servicios del abonado correspondiente a la ONT 1.

VLAN	Dirección IP	Servicio
1100	192.168.100.2	Datos
1200	192.168.200.30	VoIP
1300	192.168.30.2	IPTV
1400	192.168.40.1	CCTV

La cámara IP se encuentra en la red del cliente final; sin embargo, tanto el almacenamiento de vídeo como el servidor web que gestiona la interfaz de CCTV residen en el entorno del servidor centralizado. Para permitir el acceso remoto a dicha interfaz web, en el *router-ONT* se debe crear una regla de traducción de direcciones de red (NAT) que dirija todo el tráfico HTTP entrante hacia el servidor interno encargado del servicio de videovigilancia.

Server Name	External Port Start	External Port End	Protocol	Internal Port Start	Internal Port End	Server IP Address	WAN Interface	Remove
CCTV	80	80	TCP/UDP	80	80	192.168.40.2	veip0.4	☐

Figura 50. Configuración NAT en la ONT 1 para el servicio de CCTV. Elaboración propia.

En resumen, la ONT 1, queda configurada de la siguiente manera:

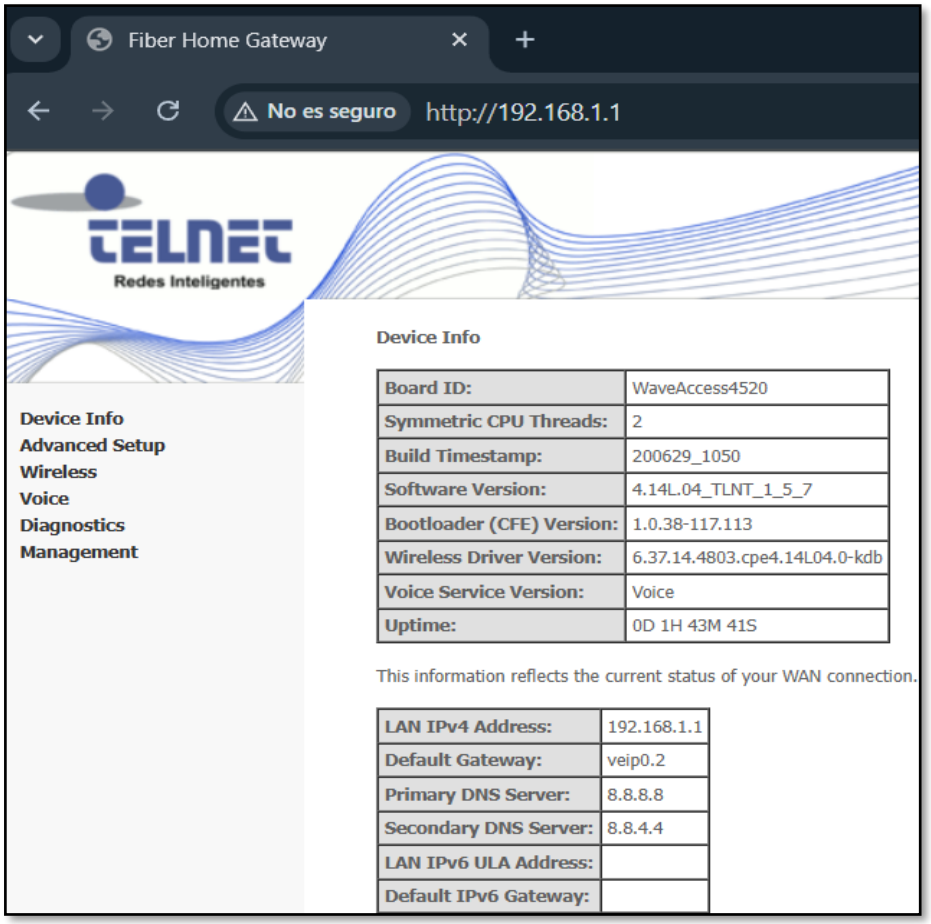
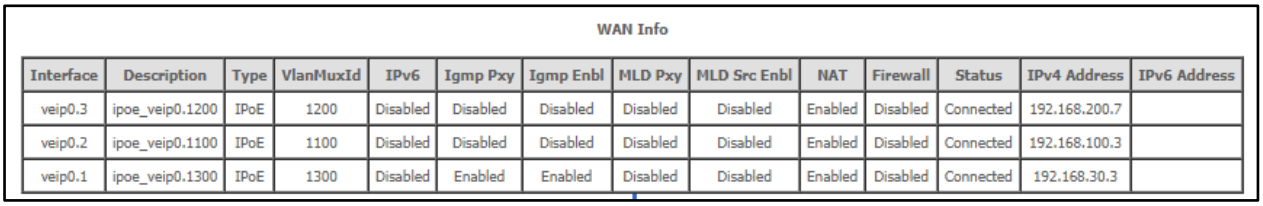


Figura 51. Información general de la ONT 1. Elaboración propia.

5.3.8 Configuración ONT 2

En el caso de la ONT 2, se han configurado los servicios de video, datos y voz, respectivamente.



WAN Info

Interface	Description	Type	VlanMuxId	IPv6	Igmp Pxy	Igmp Enbl	MLD Pxy	MLD Src Enbl	NAT	Firewall	Status	IPv4 Address	IPv6 Address
veip0.3	ipoe_veip0.1200	IPoE	1200	Disabled	Disabled	Disabled	Disabled	Disabled	Enabled	Disabled	Connected	192.168.200.7	
veip0.2	ipoe_veip0.1100	IPoE	1100	Disabled	Disabled	Disabled	Disabled	Disabled	Enabled	Disabled	Connected	192.168.100.3	
veip0.1	ipoe_veip0.1300	IPoE	1300	Disabled	Enabled	Enabled	Disabled	Disabled	Enabled	Disabled	Connected	192.168.30.3	

Figura 52. Configuración WAN de la ONT 2. Elaboración propia.

La funcionalidad de cada parámetro es la misma que la de la ONT 1, diferenciándose únicamente en las direcciones IP para cada servicio. En este caso, tenemos la siguiente relación:

Tabla 10. Relación entre las VLANs-IPs-Servicios del abonado correspondiente a la ONT 2.

VLAN	Dirección IP	Servicio
1100	192.168.100.3	Datos
1200	192.168.200.7	VoIP
1300	192.168.30.3	IPTV

En resumen, la ONT 2, queda configurada de la siguiente manera:

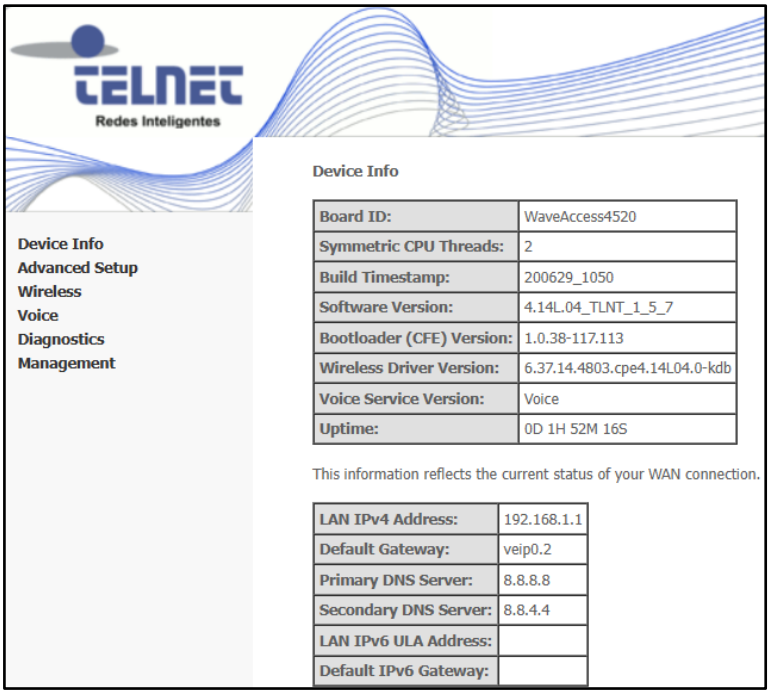


Figura 53. Información general de la ONT 1. Elaboración propia.

5.4. Configuración del *Open Virtual Switch*

En este apartado se describe el proceso de configuración del *Open Virtual Switch* (OVS) sobre el servidor, con el objetivo de interconectar de forma eficiente las distintas interfaces virtuales que sustentan nuestros servicios virtualizados. En primer lugar, se detallará la creación de las interfaces de red virtuales en el host servidor, especificando los comandos y parámetros requeridos. A continuación, se explicará cómo integrar dichas interfaces en un puente (*bridge*) gestionado por OVS, definiendo la topología lógica. Finalmente, se abordarán aspectos avanzados, como la configuración de VLANs, y reglas de *OpenFlow*, que permiten garantizar el aislamiento y el rendimiento de cada servicio.

5.4.1. Configuración de interfaces virtuales

Antes de configurar el *Open vSwitch* (OvS), es necesario activar previamente las interfaces virtuales en el servidor, permitiendo posteriormente su integración con el OvS para establecer la comunicación con los servicios virtualizados. Con el fin de automatizar este proceso y evitar su configuración manual tras cada reinicio del sistema, se ha desarrollado un *script* que ejecuta automáticamente esta tarea durante el arranque. El código del *script* es el siguiente:

```
#!/bin/bash
# Script para crear y levantar interfaces virtuales usando TUN/TAP

# Lista de interfaces a crear y levantar
interfaces=("vport1" "vport2" "vport3" "vport4")

for iface in "${interfaces[@]}"; do
    # Verifica si la interfaz ya existe
    if ip link show "$iface" > /dev/null 2>&1; then
        echo "La interfaz $iface ya existe."
    else
        echo "Creando la interfaz $iface en modo TAP..."
        sudo ip tuntap add dev "$iface" mode tap
        if [ $? -eq 0 ]; then
            echo "Interfaz $iface creada exitosamente."
        else
            echo "Error al crear la interfaz $iface."
            continue
        fi
    fi

    echo "Levantando la interfaz $iface..."
```

```
    sudo ip link set "$iface" up
    if [ $? -eq 0 ]; then
        echo "Interfaz $iface levantada correctamente."
    else
        echo "Error al levantar la interfaz $iface."
    fi
done

# Levantando la interfaz GLbridge
echo "Levantando la interfaz GLbridge..."
sudo ip link set GLbridge up
if [ $? -eq 0 ]; then
    echo "Interfaz GLbridge levantada correctamente."
else
    echo "Error al levantar la interfaz GLbridge."
Fi
```

Una vez que las interfaces virtuales estén operativas y correctamente configuradas en nuestro servidor, procederemos a integrarlas con el *Open vSwitch* (OvS). Previamente, es necesario realizar la instalación del *software* OvS según el procedimiento descrito detalladamente en el Anexo A1. Tras completar la instalación, se procede a crear un *bridge* (conmutador) para luego incorporan las interfaces virtuales al *switch* virtual siguiendo los siguientes pasos:

1. Añadir el *bridge* que vamos a utilizar en nuestra red:

sudo ovs-vsctl add-br GLbridge

2. Añadir las interfaces necesarias en el *bridge*:

sudo ovs-vsctl add-port GLbridge enp6s0

sudo ovs-vsctl add-port GLbridge vport1

sudo ovs-vsctl add-port GLbridge vport2

sudo ovs-vsctl add-port GLbridge vport3

sudo ovs-vsctl add-port GLbridge vport4

Para finalmente conseguir la siguiente configuración en nuestro OvS llamado “GLbridge”:

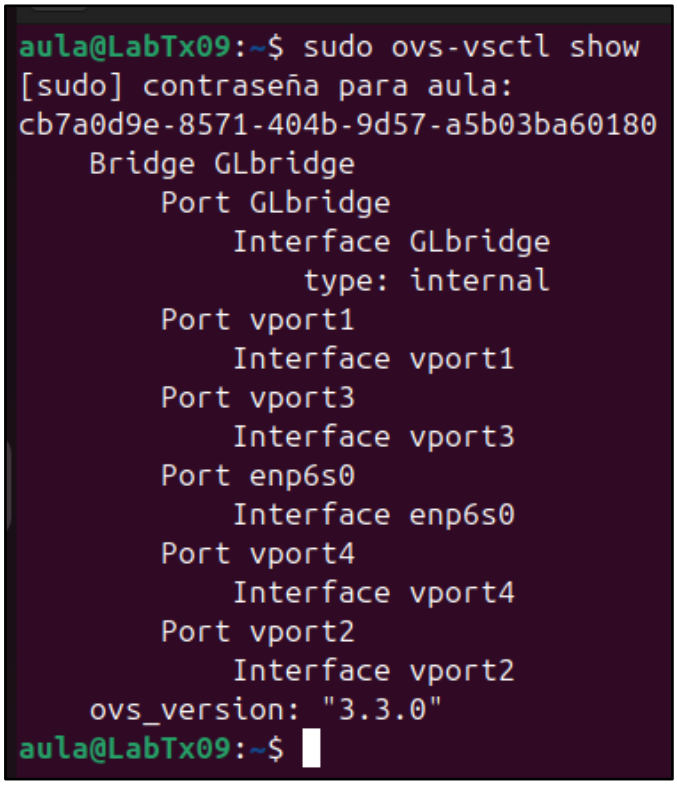


Figura 54. Configuración general del OvS. Elaboración propia.

5.5. Configuración de los servidores virtualizados

Llegado a este punto, después de tener operativo el OvS, pasaremos a configurar los servicios virtualizados que se ofrecen en la red implementada en este TFG.

5.5.1 Configuración del servidor de datos

Para implementar el servicio de datos, se procedió a desplegar un *router* virtual basado en VyOS, una plataforma de código abierto ampliamente utilizada para entornos virtualizados. Inicialmente, se descargó la imagen ISO oficial de VyOS para posteriormente crear y configurar la máquina virtual que actuará como servidor encargado del tráfico de datos dentro de la infraestructura diseñada. En el caso del *router* virtual, tenemos que establecer la siguiente configuración de red, que permitirá a los usuarios finales de la red GPON, tener acceso a internet.

En primera instancia, se configuro el “Adaptador 1” de red en modo “Adaptador puente”, logrando así que este interfaz sea la puerta de salida, ya que es la que está conectada al proveedor *fiber to the home* (FTTH) instalado en el laboratorio.

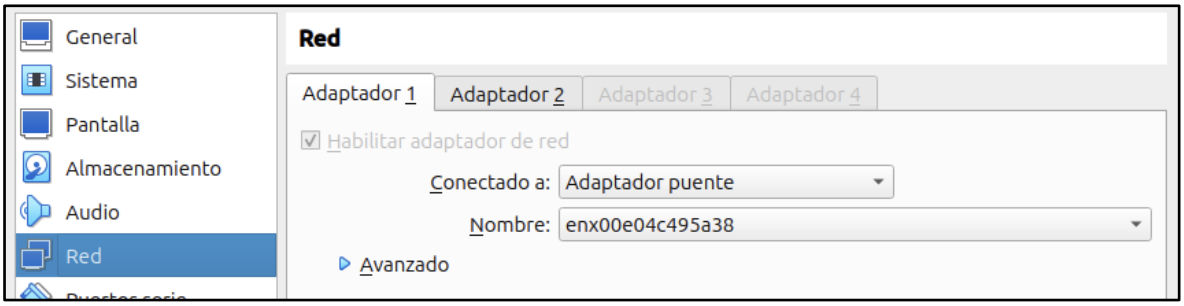


Figura 55. Configuración del adaptador 1 del router VyOS, salida hacia internet. Elaboración propia.

En segundo lugar, se configuro el "Adaptador 2" de red en modo "Adaptador puente" ya que será la interfaz conectada al interior de la red GPON, con la que los usuarios tendrán acceso directo.

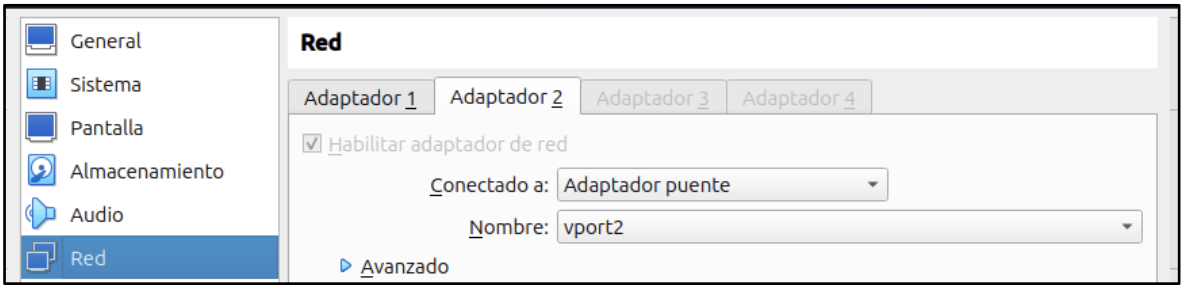


Figura 56. Configuración del adaptador 2 del router VyOS, conexión hacia el OvS. Elaboración propia.

Finalmente, dentro de la configuración del *router* virtual se han tenido que establecer dos interfaces, cada una conectada a la interfaz de red mencionada anteriormente.

```
interfaces {
    ethernet eth0 {
        address dhcp
        hw-id 08:00:27:33:7d:b0
        offload {
            gro
            gso
            sg
            tso
        }
    }
    ethernet eth1 {
        hw-id 08:00:27:a7:44:e6
        offload {
            gro
            gso
            sg
            tso
        }
        vif 100 {
            address 192.168.100.1/24
        }
    }
    loopback lo {
:_
```

Figura 57. Resumen de la configuración de interfaces del router VyOS. Elaboración propia.

En la figura superior, tenemos lo siguiente:

- Interfaz *ethernet eth0* conectada a *enx00e04c495a38.*
- Interfaz *ethernet eth1* conectada al *vport2* del OvS.

Para que lo anterior tenga sentido, tenemos que, finalmente añadir una regla de NAT, que hará la traducción de direcciones entre la red local (GPON) y la red pública (Proveedor FTTH del laboratorio).

Figura 58. Resumen de la configuración NAT del router VyOS.

En la figura superior, vemos que configurada la *regla 100*, que se encarga de traducir todo el tráfico de la red interna *192.168.100.0* a la red pública externa de forma enmascarada.

A manera de resumen se muestra la siguiente tabla con la relación de interfaces-direcciones:

Tabla 11. Configuración de interfaces de red del Servidor de datos, router VyOS.

Adaptador VitualBox	Interface VyOS	HW-ID (MAC)	Dirección IP
Adaptador 1 (enx00e04c495a38)	eth0	08:00:27:33:7D:B0	DHCP (FTTH)
Adaptador 2 (vport2)	eth1	08:00:27:A7:44:E6	—
—	eth1 (vif 100): subinterfaz VLAN 100	—	192.168.100.1/24

5.5.2 Configuración del servidor de vídeo

Para el servicio de vídeo, se hizo uso la VLC como *software* de emisión y reproducción del tráfico *multicast* que comprende este servicio. Antes de nada, se tiene que establecer la configuración de red:

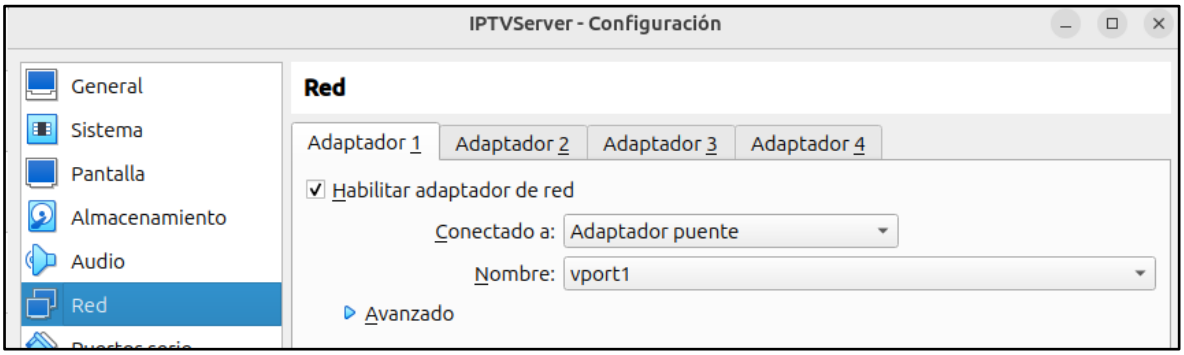


Figura 59. Configuración del adaptador 1 del servidor IPTV, conexión hacia el OvS. Elaboración propia.

Como se observa en la figura superior, se ha configurado el "Adaptador 1" como "Adaptador puente", para así poder estar directamente conectado a la interfaz *vpor1* del OvS que proporciona este servicio en particular.

Quedando la relación entre interfaces y direcciones IP en caso, de la siguiente manera:

Tabla 12. Configuración de interfaces de red del Servidor de video, IPTVServer.

Adaptador VitualBox	Interface MV	HW-ID (MAC)	Dirección IP
Adaptador 1 (vport1)	enp0s3.300	08:00:27:8F:88:11	192.168.30.10

5.5.3 Configuración del servidor de VoIP

Para el despliegue del servicio de VoIP, se utilizó la plataforma *FreePBX*, descrita previamente en el capítulo 4, la cual permite gestionar de manera eficiente y sencilla la centralita telefónica *Asterisk* mediante una interfaz web intuitiva. Inicialmente, se descargó la imagen ISO oficial del sistema *FreePBX* para posteriormente crear y configurar una máquina virtual que actúa como centralita principal encargada del tráfico del servicio de voz dentro de la red GPON implementada en este trabajo.

La configuración de red establecida para este servicio es la siguiente:

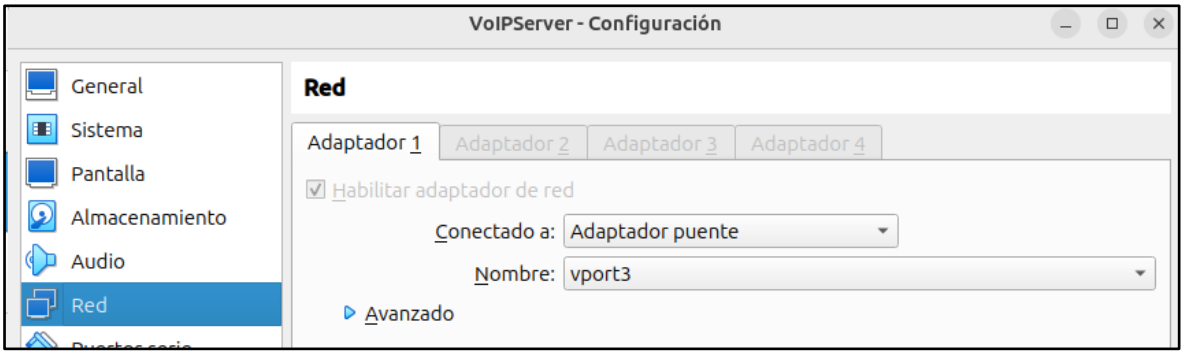


Figura 60. Configuración del adaptador 1 del VoIPServer, conexión hacia el OvS. Elaboración propia.

Como podemos observar en la ilustración superior, el "Adaptador1" se ha configurado en "Adaptador puente" para así conectarse al *vport3* previamente configurado en nuestro OvS.

De esta manera, dentro de la máquina virtual, tenemos la siguiente configuración de red:

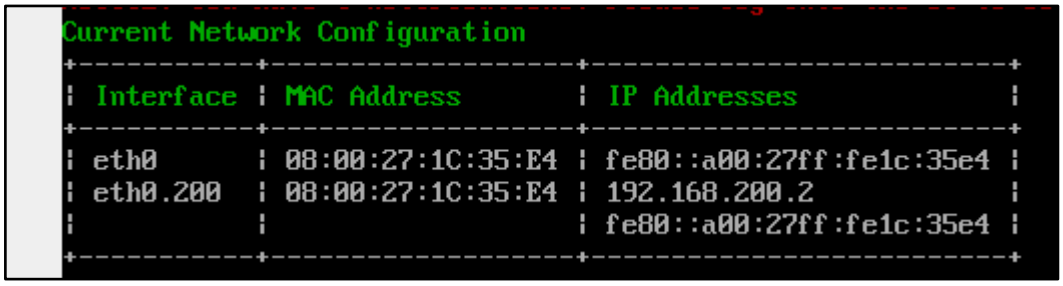


Figura 61. Configuración de interfaces del VoIPServer. Elaboración propia.

Quedando la relación entre interfaces y direcciones IP en caso, de la siguiente manera:

Tabla 13. Configuración de interfaces de red del Servidor de voz, VoIPServer.

Adaptador VitualBox	Interface PBX	HW-ID (MAC)	Dirección IP
Adaptador 1 (vport3)	eth0.200	08:00:27:1C:35:E4	192.168.200.2

5.5.4 Configuración del servidor de CCTV

Para implementar el servicio de videovigilancia (CCTV), se empleó el *software* de código abierto *ZoneMinder*, que facilita la gestión integral y centralizada de múltiples cámaras IP solicitadas por los usuarios finales de la red GPON.

Este servicio fue configurado siguiendo el esquema detallado a continuación:

- Cada usuario que contrata este servicio dispone de una cámara IP conectada directamente a su red local.
- Se proporciona al usuario final una interfaz web intuitiva para la visualización y monitorización en tiempo real de la cámara.
- El almacenamiento y gestión de las imágenes y vídeos generados por estas cámaras queda bajo responsabilidad del proveedor del servicio, que realiza dicha tarea en el servidor centralizado implementado con *ZoneMinder*.

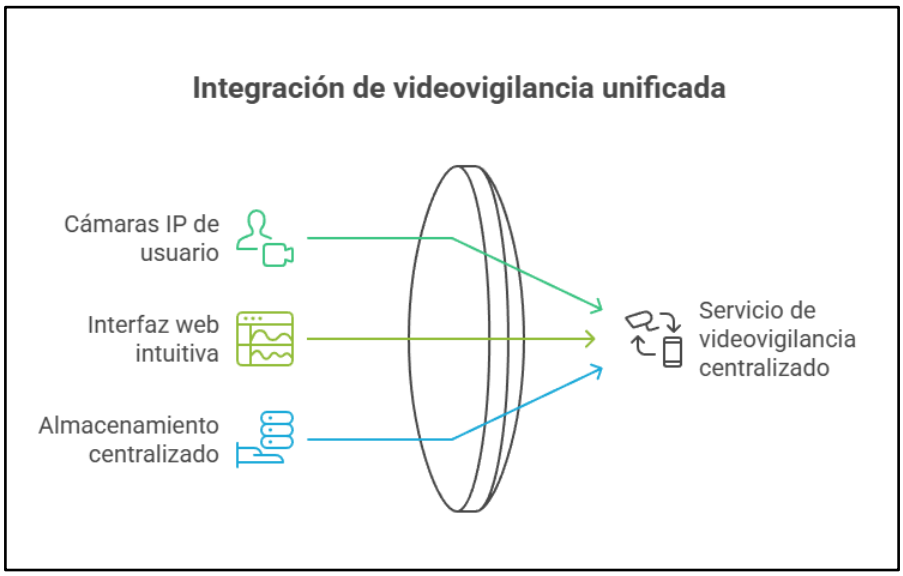


Figura 62. Esquema resumen de los componentes del servicio de CCTV. Elaboración propia.

La configuración de red establecida para este servicio es la siguiente:

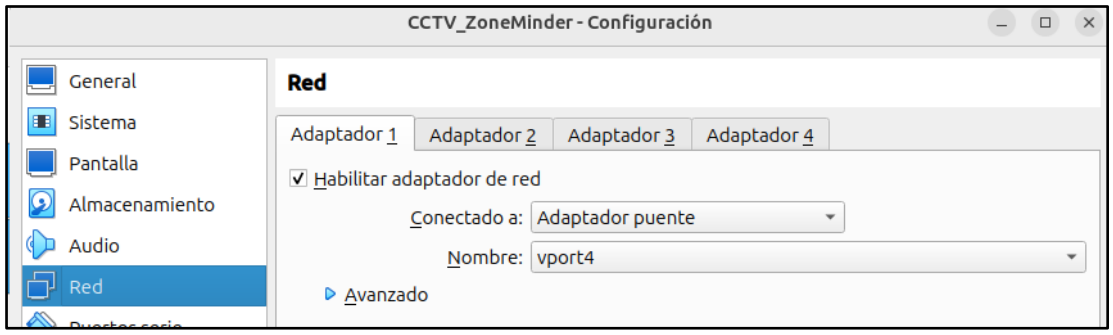


Figura 63. Configuración del adaptador 1 del CCTV Server, conexión hacia el OvS. Elaboración propia.

Como podemos observar en la ilustración superior, el “Adaptador1” se ha configurado en “Adaptador puente” para así conectarse al *vport4* previamente configurado en nuestro OvS. Quedando la relación entre interfaces y direcciones IP en caso, de la siguiente manera:

Tabla 14. Configuración de interfaces de red del Servidor de videovigilancia, CCTVServer.

Adaptador VitualBox	Interface MV	HW-ID (MAC)	Dirección IP
Adaptador 1 (vport4)	vlan400	08:00:27:48: BC:46	192.168.40.2

5.6 Integración del controlador SDN

5.6.1 Introducción

Una vez completada la configuración inicial de la red, se procedió a realizar la integración del controlador SDN de *ONOS* con la red GPON previamente establecida. Para este propósito, se optó por una implementación basada en virtualización por contenedores, la cual consiste en desplegar el controlador SDN mediante un contenedor de *Docker*. La conexión del controlador de *ONOS* y la red consta de los siguientes elementos.

- ***Controlador SDN ONOS:*** desplegado en el equipo local del administrador de red, permitiendo la gestión remota y centralizada del *switch* virtual *OpenVSwitch* (OvS).

- ***OpenVPN:*** plataforma VPN (*Virtual Private Network*) de *software* libre configurada en ambos extremos de la comunicación (máquina del administrador y servidor OvS), garantizando un túnel seguro para la administración remota de la infraestructura SDN.

- ***OpenVSwitch (OvS):*** *switch* virtual de código abierto instalado en el servidor central, proporcionando las funciones de conmutación y enrutamiento programables mediante el protocolo *OpenFlow* desde el controlador SDN.

El escenario descrito anteriormente es como se muestra en la figura 64:

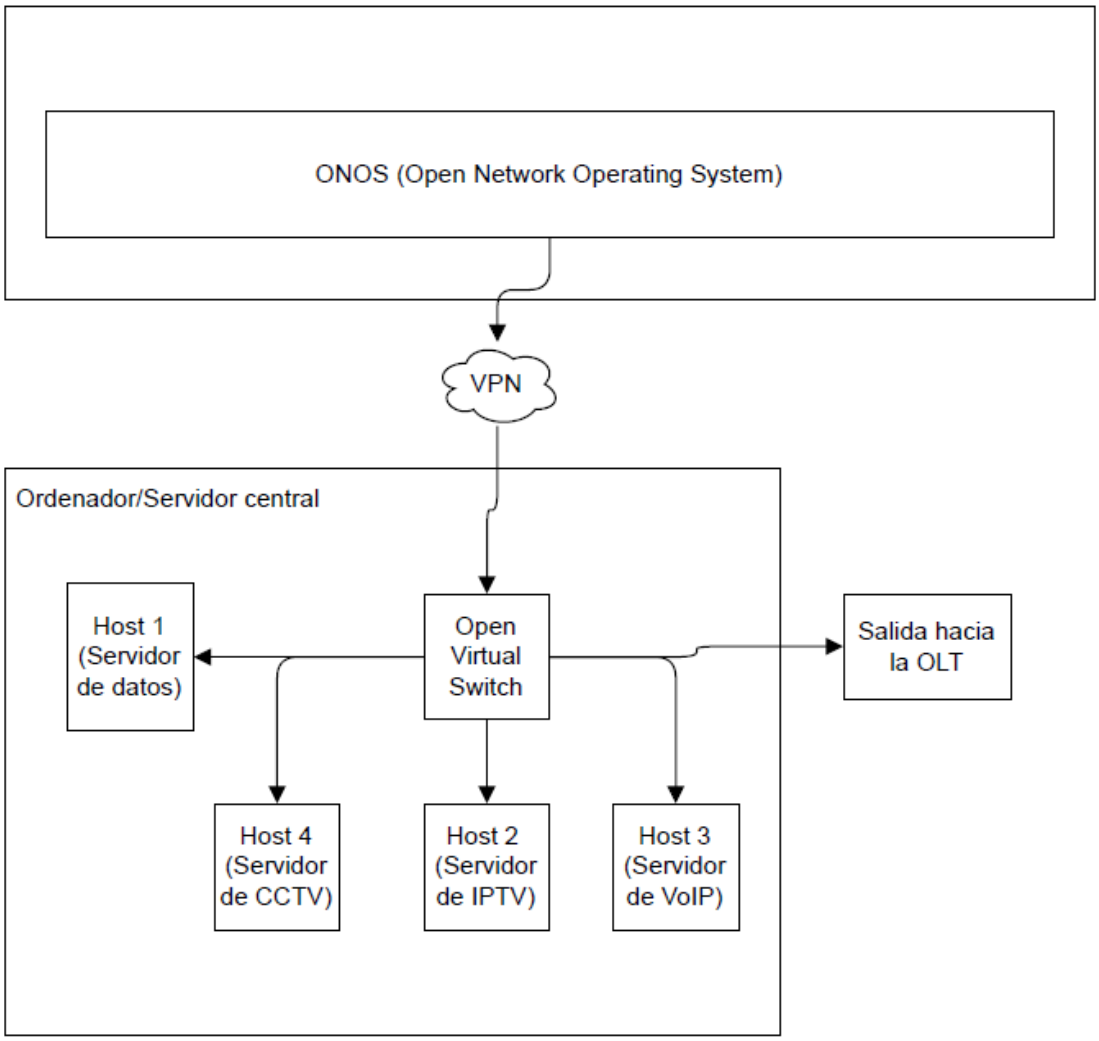


Figura 64. Arquitectura de integración SDN-GPON. Elaboración propia.

5.6.2 Configuración del controlador

Creación del contenedor de *Docker*

La integración del controlador SDN *ONOS* con la infraestructura GPON requiere el despliegue del *software* en un entorno virtualizado. Para esta implementación se utilizó *Docker Desktop* ejecutándose sobre una plataforma *Windows*, aprovechando las ventajas de la contenerización para el aislamiento y gestión de recursos.

La configuración inicial se realizó mediante la descarga de la imagen oficial de *ONOS* en su versión 2.7.0 (LTS) disponible en *Docker Hub*, proceso que se ilustra en la figura siguiente.

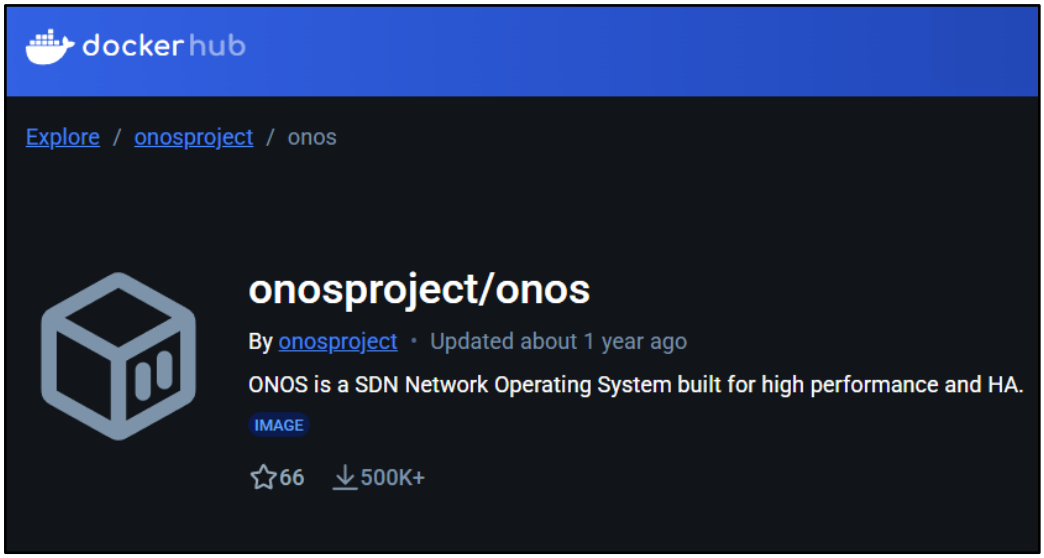


Figura 65. Repositorio oficial de la imagen Docker de ONOS [71].

Mediante el comando:

PS C:\Users\34603> docker pull onosproject/onos:2.7.0

Nos descargamos la imagen anterior, para luego introducirla en *Docker Desktop*, donde nos aparecerá de la siguiente manera.

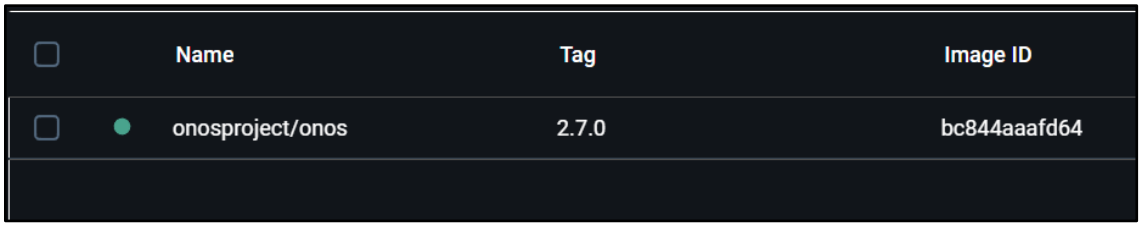


Figura 66. Controlador SDN ONOS versión 2.7.0 LTS en Docker Desktop. Elaboración propia.

Posteriormente creamos el contenedor mediante el siguiente comando, habilitando los puertos y parámetros de manera persistente:

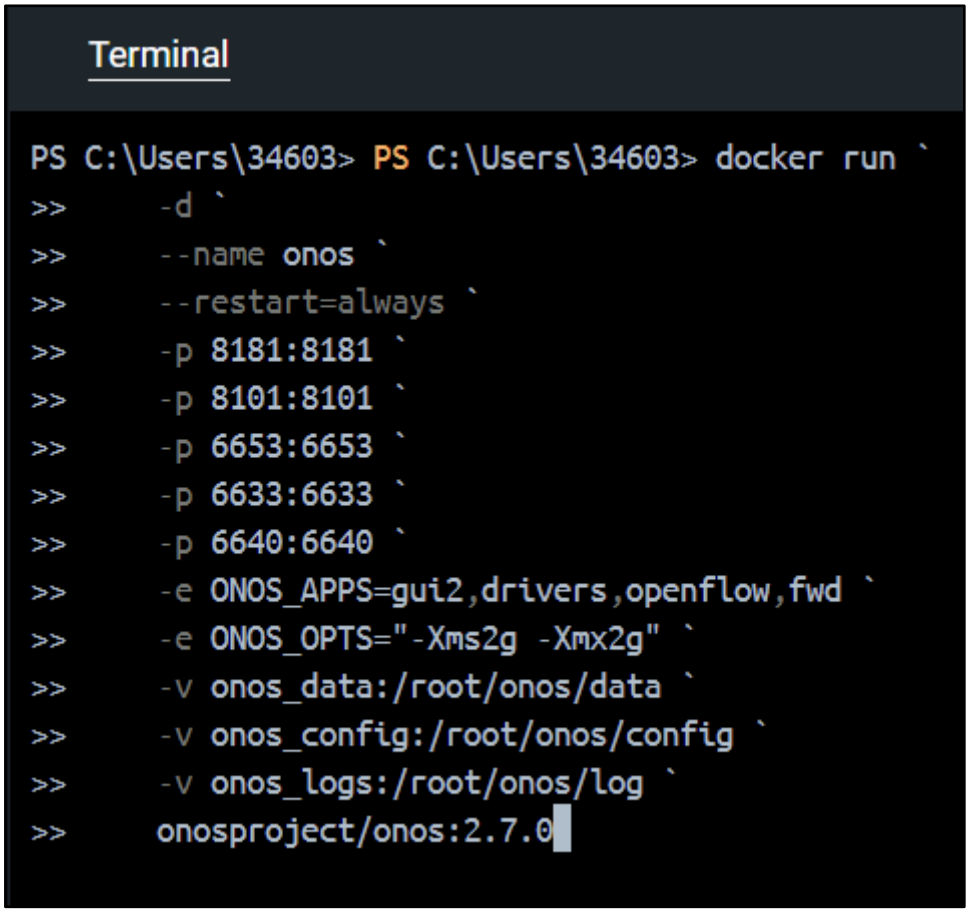


Figura 67. Creación del contenedor Docker de ONOS. Elaboración propia.

Como resultado de lo anterior, tenemos el contendor creado como se observa en la figura 68.

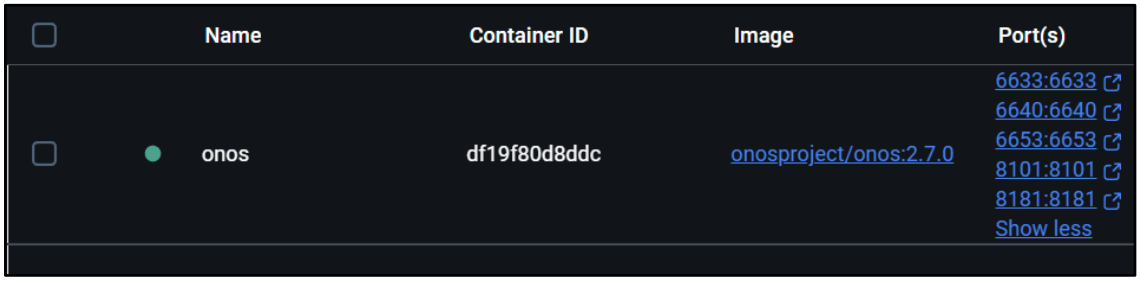


Figura 68. Contenedor onos en ejecución. Elaboración propia.

Finalmente, con el contenedor en ejecución, ya tendremos acceso al controlador de ONOS mediante la siguiente url:

Figura 69. URL local para acceder al controlador ONOS. elaboración propia.

Siendo la interfaz gráfica de acceso al controlador como se muestra en la figura 70.

Figura 70. Interfaz gráfica de inicio de sesión de ONOS. Elaboración propia.

Posteriormente, después de acceder con las credenciales oficiales (Usuario: *onos*, Contraseña: *rocks*), se accede a la interfaz gráfica de configuración y topología de la red

vinculada al controlador. En esta interfaz es posible instalar y gestionar aplicaciones de forma dinámica mediante la sección *Applications*, accesible desde el menú lateral desplegable, donde el botón "+" habilita la carga de paquetes *.oar*, posibilitando activar o desactivar módulos —por ejemplo, el *firewall* ACL o extensiones *REST*— sin necesidad de reiniciar la instancia.

Figura 71. Aplicaciones disponibles en ONOS 2.7.0. Elaboración propia.

La vista de topología, sustentada en el algoritmo *D3 Force Layout*, representa automáticamente los *switches* y *hosts* detectados, trazando sus enlaces y actualizándose en tiempo real con cada evento de descubrimiento. Mientras no haya dispositivos registrados, el lienzo muestra el aviso *"No Devices Are…"*, permitiendo identificar de un vistazo un dominio aún vacío, tal como se aprecia en la Figura 72.

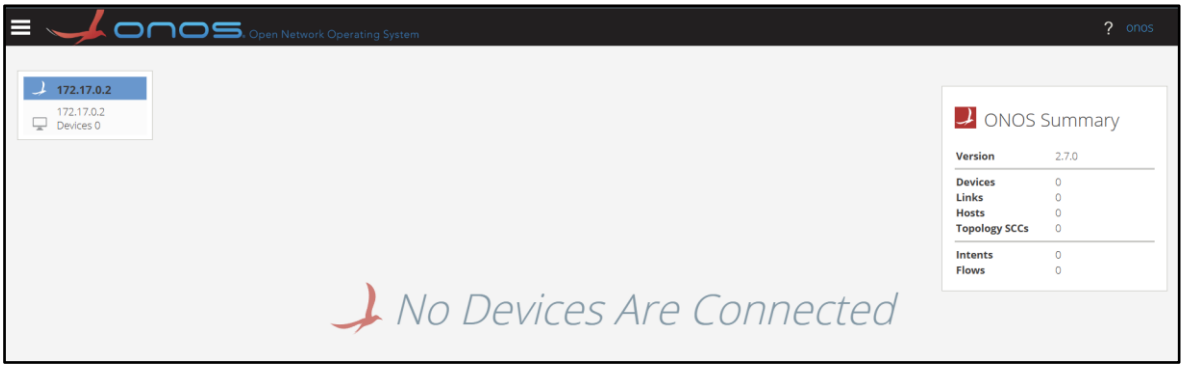


Figura 72. Vista inicial de la interfaz gráfica de ONOS (GUI2). Elaboración propia.

En el margen derecho se habilita el panel *Overview*, que sintetiza métricas críticas —versión del controlador, número de dispositivos, enlaces, *hosts*, *intents* y flujos— y cuya visibilidad puede conmutarse pulsando la tecla “O”, optimizando así el área de trabajo para redes extensas.

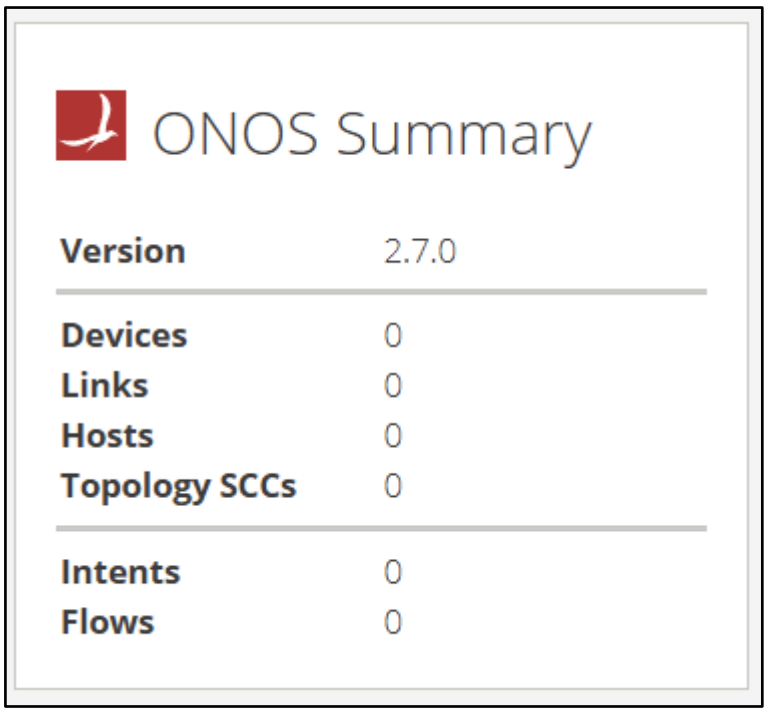


Figura 73. Panel “ONOS Summary” tras el arranque inicial del controlador. Elaboración propia.

En conjunto, estos componentes posicionan la interfaz gráfica de usuario de ONOS como una herramienta integral para la gestión modular y de alta disponibilidad de redes SDN, coherente con los principios de escalabilidad promovidos por la plataforma.

Conexión con la VPN:

Conforme a lo establecido previamente, la comunicación entre el controlador *ONOS* y el *switch* virtual se implementó utilizando un túnel VPN, garantizando así la seguridad y confidencialidad de las comunicaciones remotas en la infraestructura SDN.

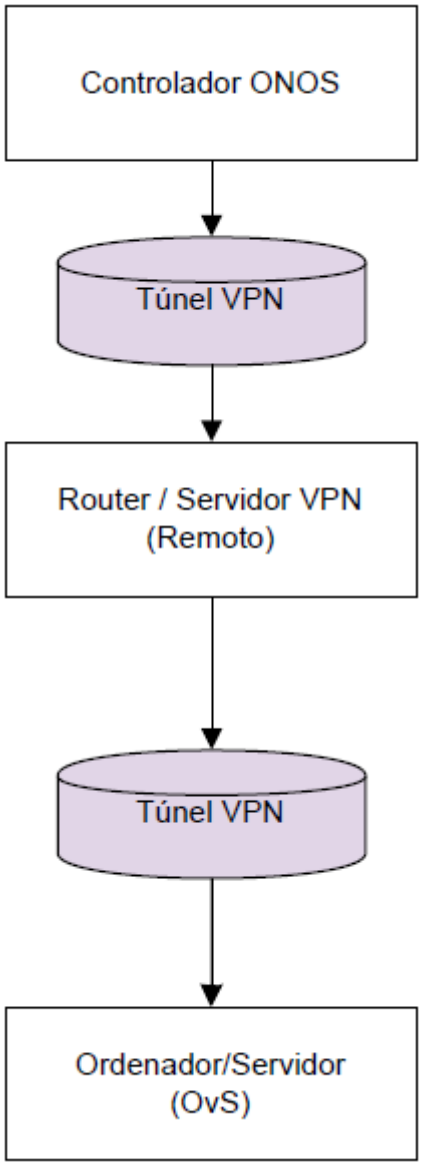


Figura 74. Conexión segura del plano de control con el OvS mediante túnel VPN. Elaboración propia.

Para establecer una conexión adecuada entre los equipos terminales, es necesario instalar un certificado de cliente en cada equipo que participa en la red. En este caso específico, se instaló un certificado de cliente en el equipo donde se despliega el controlador ONOS

(portátil personal *Asus Vivobook PRO 15*) y otro en el servidor donde se ejecuta el *OpenVSwitch*.

El cliente VPN en el equipo donde se aloja el controlador es el de la figura 75:

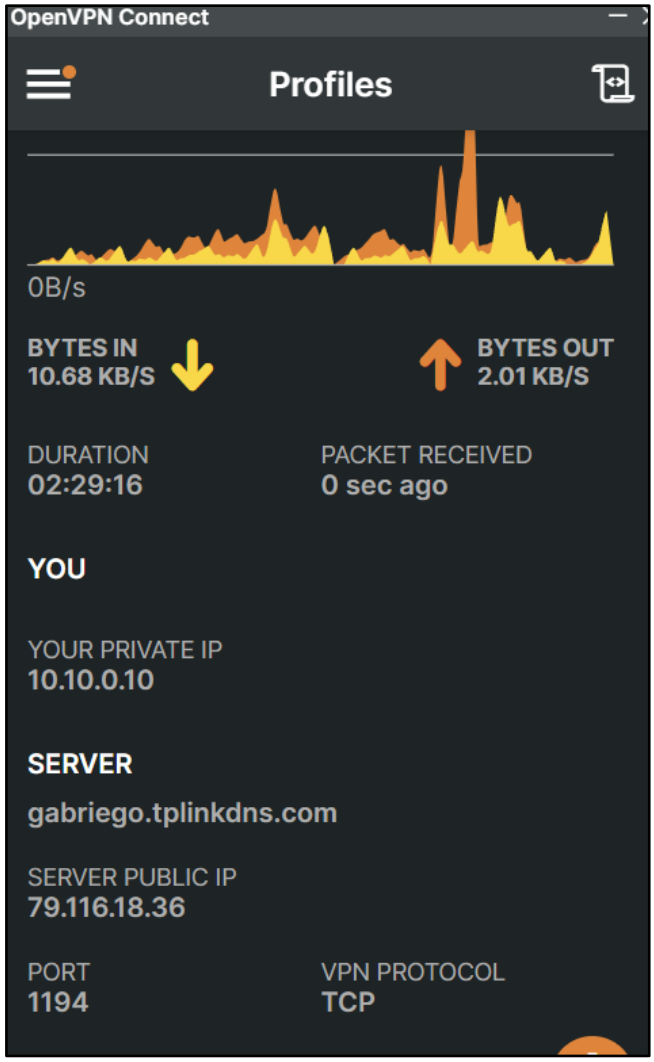


Figura 75. Sesión activa de OpenVPN Connect durante las pruebas de acceso remoto desde el controlador de ONOS. Elaboración propia.

Cuya dirección IP asignada es la 10.10.0.10.

Por el otro extremo se encuentra el otro cliente VPN, instalado en el Ordenador/Servidor como se observa en la figura siguiente:

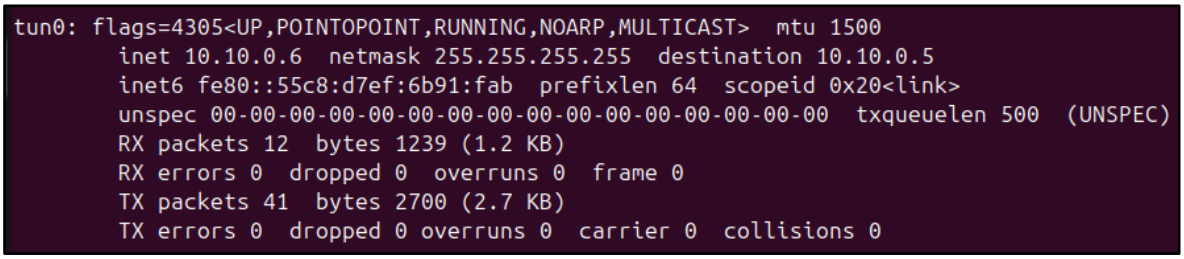


Figura 76.Sesión activa de OpenVPN Connect durante las pruebas de acceso remoto desde el Ordenador/Servidor donde se aloja el OvS. Elaboración propia.

Cuya dirección IP es la 10.10.0.6.

Conexión entre equipos finales:

Una vez establecidos los clientes VPN en funcionamiento, se configura el túnel de comunicación seguro que interconecta el controlador SDN *ONOS* con el *switch* virtual, lo que permite la administración remota y centralizada de la infraestructura GPON desplegada.

Para validar esta conectividad, se realizaron pruebas bidireccionales de comunicación entre los componentes del sistema:

***Prueba de conectividad controlador-switch*:** Se verificó la comunicación desde el controlador *ONOS* hacia el servidor que aloja el *switch* virtual, confirmando la accesibilidad del plano de control hacia el plano de datos, como se evidencia en la figura siguiente.

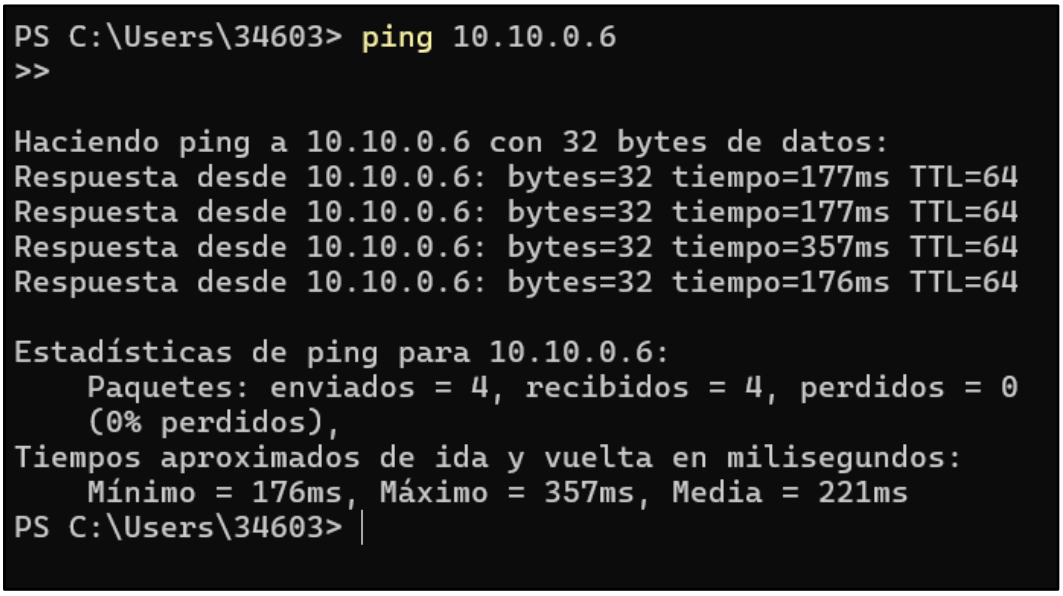


Figura 77. Prueba de conectividad desde el controlador ONOS hacia el Ordenador/Servidor (OvS). Elaboración propia.

Prueba de conectividad switch-controlador: De forma complementaria, se validó la comunicación inversa desde el servidor que aloja el *switch* virtual hacia el equipo donde reside el controlador *ONOS*, garantizando la comunicación bidireccional requerida para el correcto funcionamiento del paradigma SDN, tal como se muestra en la figura correspondiente.

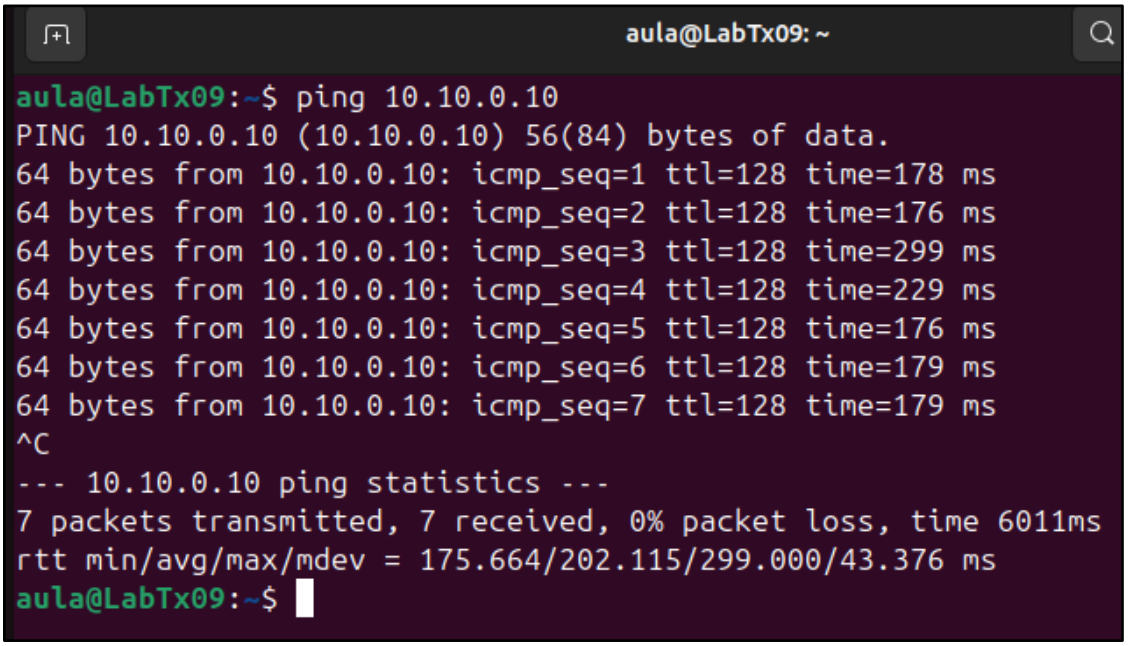


Figura 78. Prueba de conectividad desde el Ordenador/Servidor (OvS) hacia el controlador ONOS. Elaboración propia.

Capítulo 6. Resultados finales y conclusiones

6.1. Introducción

Este capítulo presenta y analiza los resultados obtenidos tras la implementación de la red GPON y su gestión mediante el controlador SDN *ONOS*. Los puntos principales que se desarrollan en este capítulo son:

- Evaluación de los resultados finales obtenidos para cada uno de los servicios implementados (Datos, IPTV, VoIP, y CCTV).
- Análisis del desempeño del controlador SDN *ONOS*, centrándose en la gestión centralizada de parámetros críticos de red.
- Comprobación de funcionalidades específicas, incluyendo la aplicación de políticas de calidad de servicio (QoS), la limitación dinámica del ancho de banda mediante medidores (*meters)* y la gestión eficiente del tráfico en la red.
- Extracción de conclusiones técnicas y operativas sobre la integración y funcionamiento de los servicios GPON bajo un entorno SDN, valorando la viabilidad, eficiencia y potencialidad de futuras implementaciones similares.

Este análisis permite determinar la eficacia de la solución propuesta y establecer recomendaciones para futuras líneas de investigación y desarrollo.

6.2. Identificación de elementos de la red

Antes de proceder con las pruebas, resulta necesario identificar los elementos que participan en la infraestructura de red. Estos componentes incluyen los *hosts*, los puertos y las VLANs asociadas con cada servicio implementado en el presente Trabajo de Fin de Grado. En la siguiente podemos identificar la relación previamente comentada.

Tabla 15. Relación de parámetros de la red SDN-GPON.

Host / Nombre	Dirección IP	VLAN ID	Interface (Puerto)	MAC Address
Abonado IPTV	Multicast: 239.1.1.1	300	enp6s0 (ID 4)	78:3D:5B:04:4B:C4
Abonado IPTV	Multicast: 239.1.1.1	300	enp6s0 (ID 4)	78:3D:5B:04:58:C5
Servidor CCTV	192.168.40.2	400	vport4 (ID 5)	08:00:27:48:BC:46
Abonado CCTV	192.168.40.1	400	enp6s0 (ID 4)	78:3D:5B:04:4B:C1
Servidor IPTV	192.168.30.10	300	vport1 (ID 1)	08:00:27:8F:88:11
Abonado VoIP	192.168.200.7	200	enp6s0 (ID 4)	78:3D:5B:04:58:C4
Servidor VoIP	192.168.200.2	200	vport3 (ID 3)	08:00:27:1C:35:E4
Abonado Datos	192.168.100.3	100	enp6s0 (ID 4)	78:3D:5B:04:58:C3
Abonado Datos	192.168.100.2	100	enp6s0 (ID 4)	78:3D:5B:04:4B:C2
Servidor de Datos	192.168.100.1	100	vport2 (ID 2)	08:00:27:A7:44:E6

6.3. Resultados de los servicios implementados

Para realizar un análisis preciso de los resultados obtenidos tras la implementación de los servicios desplegados sobre la red GPON, es necesario utilizar un analizador de paquetes. En este caso, se ha optado por *Wireshark* debido a su amplia popularidad y compatibilidad con una gran variedad de protocolos de red y dispositivos.

El análisis se efectúa desde el Ordenador/Servidor, el núcleo de la red, debido que por el pasa todo el tráfico involucrado en esta solución red.

Resultados del servicio de Datos:

Para evaluar la funcionalidad de este servicio, hemos de acotar las pruebas de conectividad en diferentes escalas.

Primero, comprobaremos que las direcciones IP involucradas para este servicio logran comunicarse. En la figura 79 siguiente podemos observar la correcta comunicación entre el la interfaz local de *router* VyOS con dirección IP = 192.168.100.1, y la interfaz virtual de la ONT correspondiente al usuario final veip0.2 con dirección IP 192.168.100.2.

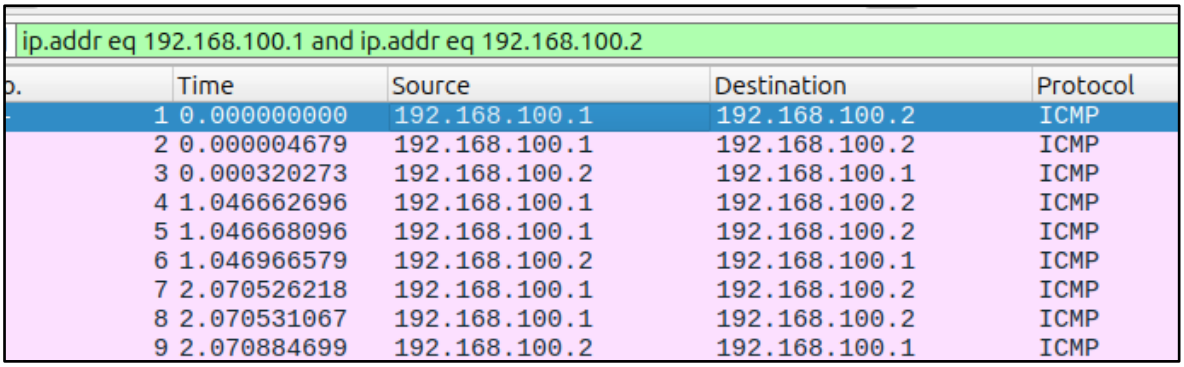


Figura 79. Prueba de conectividad entre Abonado y Servidor de Datos (Router virtual). Elaboración propia.

Siendo el detalle de los paquetes anteriores, como se muestra a continuación en la figura siguiente.

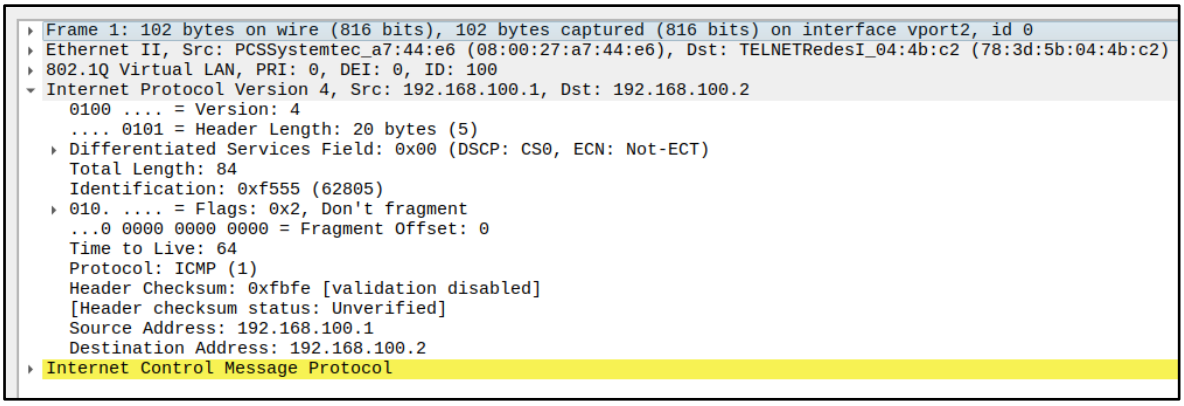


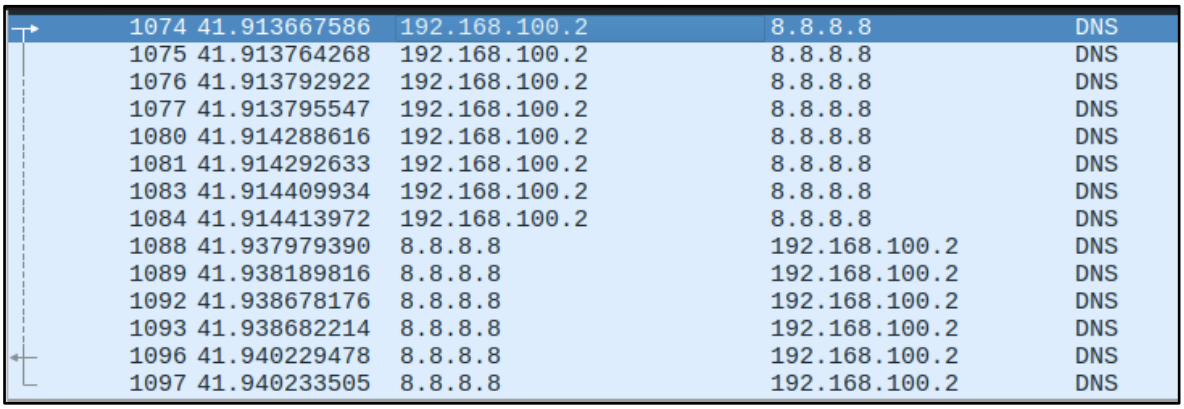
Figura 80. Detalle del paquete ICMP entre el Abonado y el Servidor de Datos (Router virtual). Elaboración propia.

La captura mostrada en la figura 80, obtenida con *Wireshark*, presenta un *frame Ethernet* etiquetado con VLAN 802.1Q correspondiente a la VLAN 100 (Servicio de Datos). El *frame* transporta un mensaje ICMP de tipo *Echo Request* (*ping*) encapsulado en un datagrama IPv4, generado por el host 192.168.100.1 con destino al host 192.168.100.2. El análisis del tráfico confirma que ambos dispositivos se encuentran en la misma subred de red, lo que

permite la comunicación directa sin necesidad de enrutamiento a través de dispositivos intermedios.

Esta configuración permite que el usuario final tenga conectividad desde su red local hacia la red GPON implementada mediante un primer NAT. Sin embargo, para que el abonado con servicio de Datos contratado pueda acceder a Internet, es necesario implementar una segunda traducción NAT entre la dirección IP privada 192.168.100.1 y la dirección pública 10.11.204.28. Esta última corresponde a la interfaz WAN del *router* VyOS, cuyo rango de direccionamiento pertenece al proveedor FTTH contratado para el laboratorio donde se desarrolló el proyecto. El objetivo de esta configuración es proporcionar acceso directo a Internet a los usuarios suscritos a este servicio.

Como segunda comprobación, a manera de ejemplo pasaremos a realizar una búsqueda al portal web de la ULPGC (www.ulpgc.es). Como se observa en la figura 81.



1074	41.913667586	192.168.100.2	8.8.8.8	DNS
1075	41.913764268	192.168.100.2	8.8.8.8	DNS
1076	41.913792922	192.168.100.2	8.8.8.8	DNS
1077	41.913795547	192.168.100.2	8.8.8.8	DNS
1080	41.914288616	192.168.100.2	8.8.8.8	DNS
1081	41.914292633	192.168.100.2	8.8.8.8	DNS
1083	41.914409934	192.168.100.2	8.8.8.8	DNS
1084	41.914413972	192.168.100.2	8.8.8.8	DNS
1088	41.937979390	8.8.8.8	192.168.100.2	DNS
1089	41.938189816	8.8.8.8	192.168.100.2	DNS
1092	41.938678176	8.8.8.8	192.168.100.2	DNS
1093	41.938682214	8.8.8.8	192.168.100.2	DNS
1096	41.940229478	8.8.8.8	192.168.100.2	DNS
1097	41.940233505	8.8.8.8	192.168.100.2	DNS

Figura 81. Resolución DNS para el portal web de la ULPGC. Elaboración propia.

En la figura 81, vemos que el abonado hace la petición, cuya resolución es resuelta en primera instancia por el servidor DNS configurado, en este caso el 8.8.8.8 de *Google*.

Internamente el abonado consigue acceder a la página solicitada mediante el doble NAT necesario en esta solución de red, que se logra enmascarando la dirección del abonado anterior con la dirección WAN que se muestra en la siguiente figura.

321	12.830737792	10.11.204.28	185.125.190.17	TCP
322	12.882352395	185.125.190.17	10.11.204.28	TCP
323	12.882373164	10.11.204.28	185.125.190.17	TCP
324	12.882428829	10.11.204.28	185.125.190.17	HTTP
325	12.934235185	185.125.190.17	10.11.204.28	HTTP
326	12.934235306	185.125.190.17	10.11.204.28	TCP
327	12.934262998	10.11.204.28	185.125.190.17	TCP
328	12.934333261	10.11.204.28	185.125.190.17	TCP
329	12.985921725	185.125.190.17	10.11.204.28	TCP

Figura 82. Conexión de la interfaz WAN del Router virtual con la página de la ULPGC. Elaboración propia.

Como se observa en la figura 82, al realizar una petición HTTP por parte del abonado, la dirección IP que establece la conexión con el servidor web donde se encuentra alojada la página es la 10.11.204.28, enmascarando de esta manera las direcciones IP privadas de la red GPON local, tal como se mencionó anteriormente.

De esta manera, el usuario final puede acceder a cualquier servicio disponible en Internet.

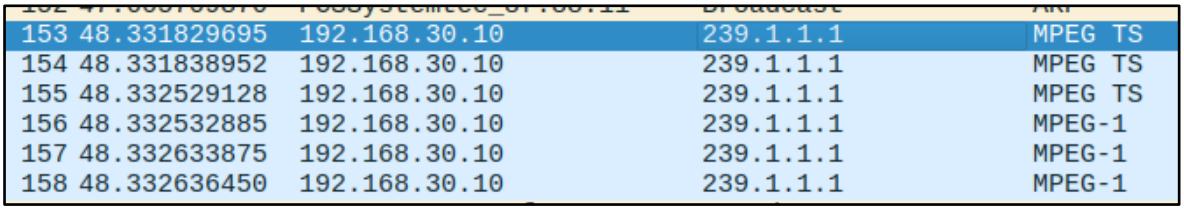
```
   Speedtest by Ookla

      Server: WiFi Canarias - La Orotava (id: 11990)
         ISP: Telefonica de Espana
Idle Latency:    10.18 ms   (jitter: 0.12ms, low: 10.06ms, high: 10.25ms)
    Download:   136.19 Mbps (data used: 157.2 MB)
                  7.82 ms   (jitter: 4.95ms, low: 7.43ms, high: 72.64ms)
      Upload:     0.65 Mbps [============/      ] 68%   - latency: 8.30 ms    .82 ms
```

Figura 83. Prueba de velocidad por parte del abonado final correspondiente a la ONT 1. Elaboración propia.

Resultados del servicio de IPTV:

Para evaluar el servicio de IPTV, se realizará la emisión de un *stream* de video y se analizarán los paquetes y protocolos involucrados para garantizar el correcto funcionamiento y visualización del servicio por parte del abonado.

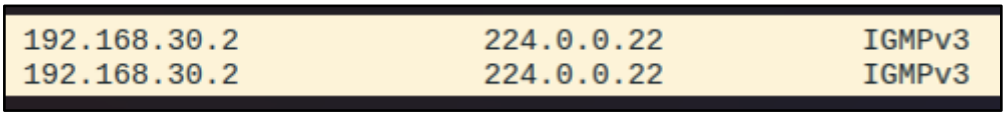
Al iniciar el *stream*, desde el servidor de video, la dirección IP 192.168.30.10 emite el flujo hacia una dirección *multicast,* 239.1.1.1, la cual se propaga a través de toda la red, permitiendo que los abonados se suscriban al grupo *multicast* asociado a dicha dirección.



153 48.331829695	192.168.30.10	239.1.1.1	MPEG TS
154 48.331838952	192.168.30.10	239.1.1.1	MPEG TS
155 48.332529128	192.168.30.10	239.1.1.1	MPEG TS
156 48.332532885	192.168.30.10	239.1.1.1	MPEG-1
157 48.332633875	192.168.30.10	239.1.1.1	MPEG-1
158 48.332636450	192.168.30.10	239.1.1.1	MPEG-1

Figura 84. Conexión del servidor de IPTV con el grupo Multicast. Elaboración propia.

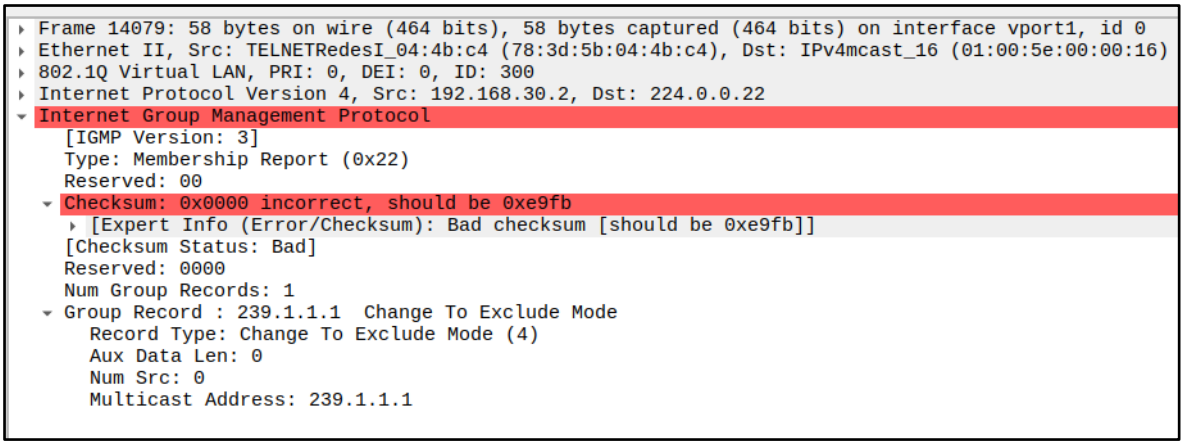
Del lado del abonado, al este sintonizar el *stream*, aparece tráfico IGMP, que muestra el funcionamiento de unión al grupo *multicast* relacionado con la transmisión efectuada, como se muestra en la siguiente figura.



192.168.30.2	224.0.0.22	IGMPv3
192.168.30.2	224.0.0.22	IGMPv3

Figura 85. Conexión del abonado con el grupo Mulitcast. Elaboración propia.

En la figura 85 se observa al host 192.168.30.2 enviando IGMPv3 *Membership Report* a la dirección *link-local* 224.0.0.22, utilizada para registrar altas o renovaciones de suscripción a grupos *multicast* en la subred, siendo nuestro caso el 239.1.1.1, como se observa en la siguiente figura.



```
▸ Frame 14079: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface vport1, id 0
▸ Ethernet II, Src: TELNETRedesI_04:4b:c4 (78:3d:5b:04:4b:c4), Dst: IPv4mcast_16 (01:00:5e:00:00:16)
▸ 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 300
▸ Internet Protocol Version 4, Src: 192.168.30.2, Dst: 224.0.0.22
▾ Internet Group Management Protocol
    [IGMP Version: 3]
    Type: Membership Report (0x22)
    Reserved: 00
  ▾ Checksum: 0x0000 incorrect, should be 0xe9fb
    ▸ [Expert Info (Error/Checksum): Bad checksum [should be 0xe9fb]]
    [Checksum Status: Bad]
    Reserved: 0000
    Num Group Records: 1
  ▾ Group Record : 239.1.1.1  Change To Exclude Mode
      Record Type: Change To Exclude Mode (4)
      Aux Data Len: 0
      Num Src: 0
      Multicast Address: 239.1.1.1
```

Figura 86. Detalle del IGMPv3 Mermbership Report (Unión al grupo Multicast). Elaboración propia.

La figura superior ratifica lo expuesto anteriormente: el host 192.168.30.2 (VLAN 300) envía un mensaje IGMPv3 *Membership Report* dirigido a la dirección *link-local* 224.0.0.22. Dentro del mensaje se incluye el *Group Record* para la dirección 239.1.1.1 con la acción *"Change*

to Exclude Mode", lo que indica que el host comunica al emisor del *stream multicast* (Servidor de IPTV) su nueva preferencia de filtrado de fuentes para dicho grupo.

En el caso que el usuario final deje de visualizar el *stream*, tendremos un abandono de grupo *multicast*, como se puede observar en la siguiente figura.

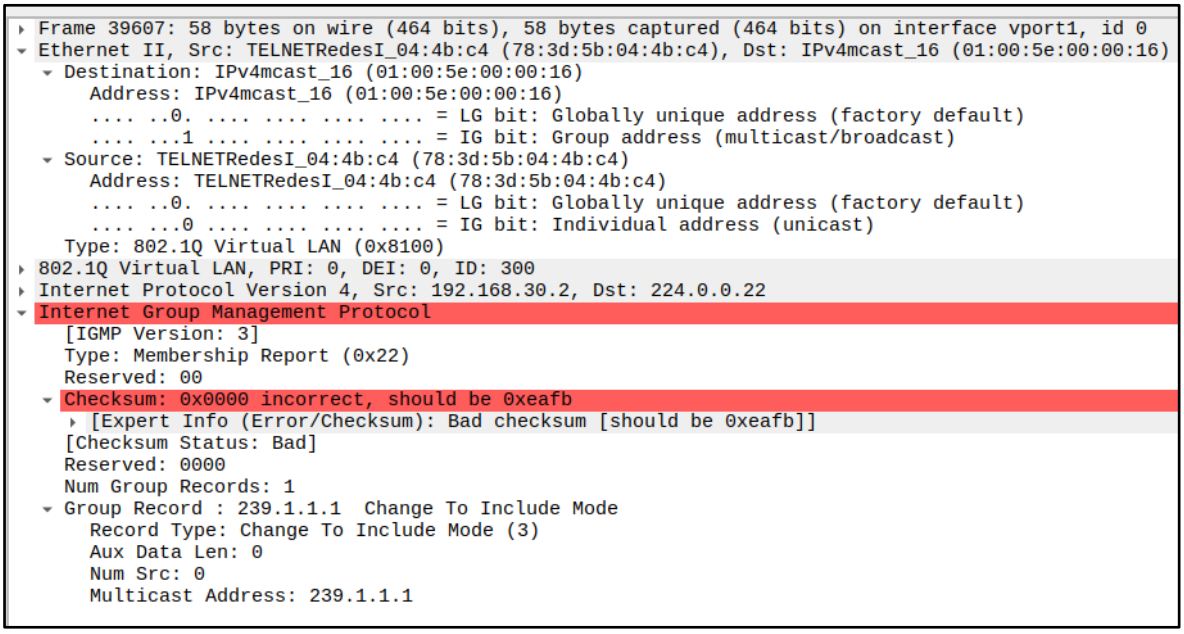


Figura 87. Detalle del IGMPv3 Membership Report (Liberación del grupo Multicast). Elaboración propia.

Cuando el usuario decide dejar de ver el contenido *multicast*, su dispositivo 192.168.30.2 (conectado a la VLAN 300) necesita avisar a la red que ya no está interesado en recibir ese flujo de video. Para ello, envía un mensaje IGMPv3 *Membership Report* a la dirección especial *link-local* 224.0.0.22, incluyendo información sobre el grupo 239.1.1.1 con el comando *Change to Include Mode* y configurando Num Src = 0. En términos sencillos, esto significa que el dispositivo está diciéndole a los equipos de red: "ya no quiero recibir este canal". Es como cuando te das de baja de un servicio de *streaming*. Por último, es normal que *Wireshark* muestre una advertencia sobre *checksum* incorrecto; esto ocurre porque las tarjetas de red modernas procesan esta información de manera optimizada, pero no significa que haya algún problema real con la transmisión [85][86][87].

Finalmente, y después de analizar cómo funciona el tráfico de este servicio para que llegue a funcionar correctamente en el usuario final, observamos en la figura 88 la reproducción del contenido de este por parte de uno de los abonados.

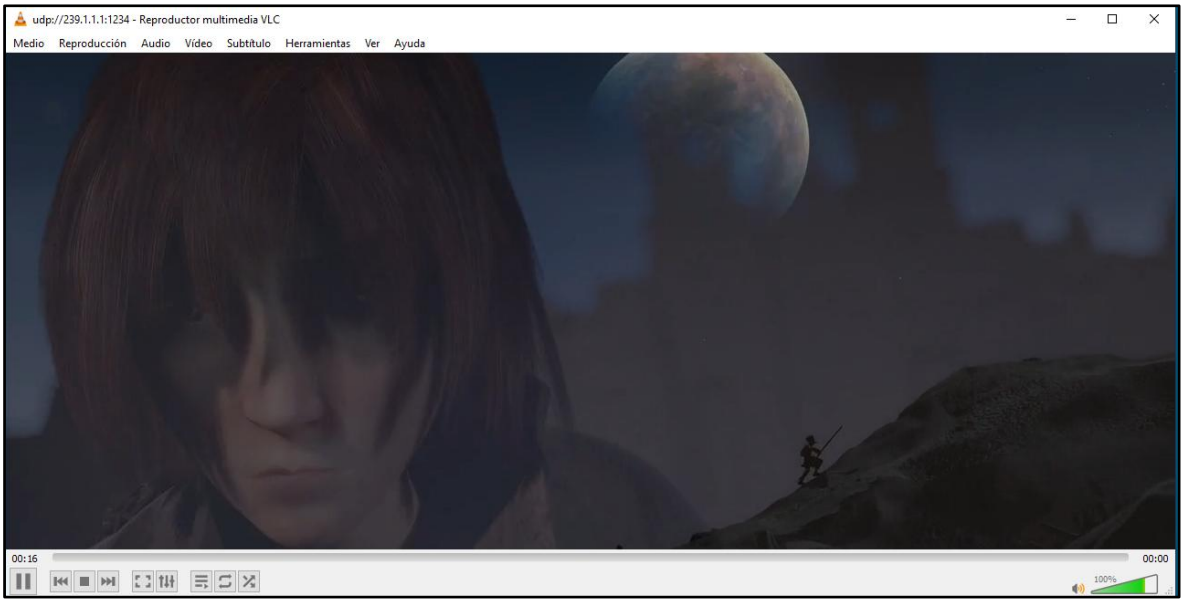


Figura 88. Reproducción del stream por parte del abonado final conectado a la ONT 1. Elaboración propia.

Resultados del servicio de VoIP:

Para evaluar el funcionamiento del servicio de VoIP, se implementará un protocolo de verificación de conectividad. En esta fase inicial, se procederá a conectar las dos líneas o extensiones registradas en la centralita telefónica virtual, lo que permitirá analizar los componentes internos necesarios para garantizar el correcto funcionamiento del servicio. Para ello, conectaremos la extensión perteneciente a la ONT 1 como se observa en la figura 89 (Se hace los mismo para la extensión correspondiente a la ONT 2).

192.168.200.30	192.168.200.2	SIP
192.168.200.30	192.168.200.2	SIP
192.168.200.2	192.168.200.30	SIP
192.168.200.2	192.168.200.30	SIP
192.168.200.30	192.168.200.2	SIP
192.168.200.30	192.168.200.2	SIP
192.168.200.2	192.168.200.30	SIP
192.168.200.2	192.168.200.30	SIP
192.168.200.2	192.168.200.30	SIP
192.168.200.2	192.168.200.30	SIP
192.168.200.30	192.168.200.2	SIP
192.168.200.30	192.168.200.2	SIP
192.168.200.2	192.168.200.30	SIP
192.168.200.2	192.168.200.30	SIP
192.168.200.30	192.168.200.2	SIP
192.168.200.30	192.168.200.2	SIP

Figura 89. Conexión entre la extensión conectada a la ONT 1 (192.168.200.30) y la centralita virtual (192.168.200.2). Elaboración propia.

Siguiendo el detalle de los paquetes anteriores, tenemos la correspondencia en la figura 90.

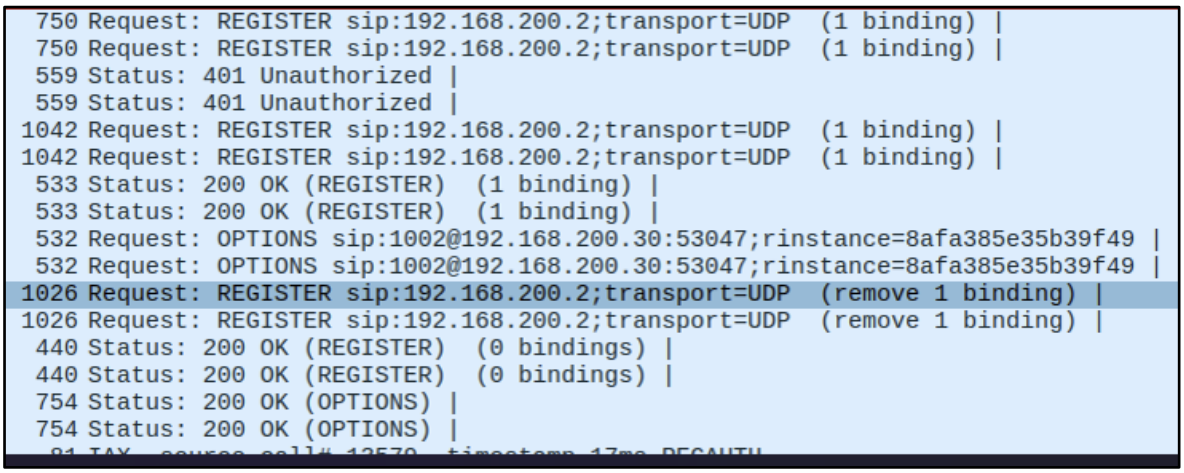


Figura 90. Detalle de conexión entre la extensión de la ONT 1 (192.168.200.30) y la centralita virtual (192.168.200.2). Elaboración propia.

La figura 90 ilustra la secuencia característica del protocolo SIP (*Session Initiation Protocol*) durante los procesos de registro y des registro de una extensión en la centralita de VoIP. El proceso se inicia cuando la extensión ubicada en la dirección IP 192.168.200.30 transmite un mensaje *REGISTER* hacia la centralita con dirección IP 192.168.200.2. Como respuesta inicial, la centralita emite un mensaje 401 *Unauthorized*, requiriendo la autenticación mediante credenciales *Digest*.

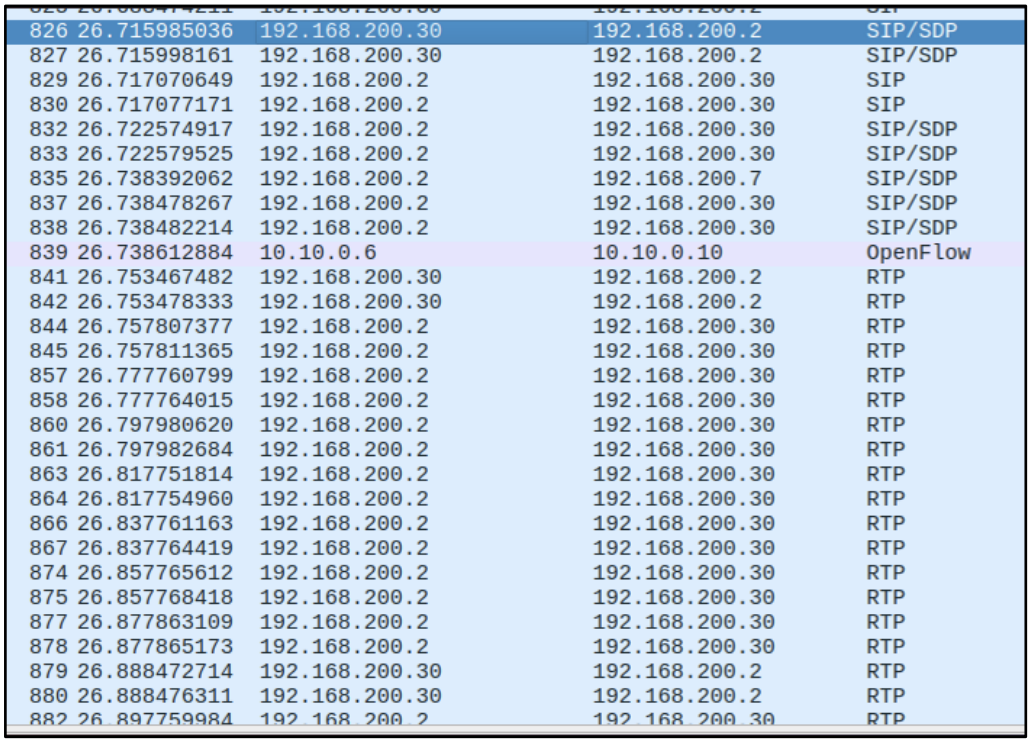
Posteriormente, la extensión retransmite el mensaje *REGISTER* incorporando las credenciales solicitadas, momento en el cual la centralita valida la información y confirma la aceptación del registro mediante un mensaje 200 *OK*, estableciendo de esta manera el enlace activo (*binding*).

Durante la sesión activa, la centralita implementa un mecanismo de supervisión continua mediante el envío periódico de mensajes *OPTIONS*, diseñados para verificar el estado de conectividad del cliente SIP. La extensión responde consistentemente a estas consultas con mensajes 200 *OK*, confirmando su disponibilidad operativa.

El proceso concluye cuando la extensión requiere desconectarse o reiniciarse, momento en el que transmite un mensaje *REGISTER* con el parámetro *Expires*=0 (denominado "*remove binding*"), instruyendo la eliminación del registro. La centralita procesa esta

solicitud y confirma la finalización del enlace mediante un mensaje final 200 *OK*. El mismo proceso sucede para la extensión correspondiente al segundo abonado.

En segundo lugar, haremos una llamada desde el abonado con extensión 1002 (Abonado de la ONT 1) al abonado con extensión 1003 (Abonado de la ONT 2), la cual analizaremos en *Wireshark* como hemos venido haciendo.



826	26.715985036	192.168.200.30	192.168.200.2	SIP/SDP
827	26.715998161	192.168.200.30	192.168.200.2	SIP/SDP
829	26.717070649	192.168.200.2	192.168.200.30	SIP
830	26.717077171	192.168.200.2	192.168.200.30	SIP
832	26.722574917	192.168.200.2	192.168.200.30	SIP/SDP
833	26.722579525	192.168.200.2	192.168.200.30	SIP/SDP
835	26.738392062	192.168.200.2	192.168.200.7	SIP/SDP
837	26.738478267	192.168.200.2	192.168.200.30	SIP/SDP
838	26.738482214	192.168.200.2	192.168.200.30	SIP/SDP
839	26.738612884	10.10.0.6	10.10.0.10	OpenFlow
841	26.753467482	192.168.200.30	192.168.200.2	RTP
842	26.753478333	192.168.200.30	192.168.200.2	RTP
844	26.757807377	192.168.200.2	192.168.200.30	RTP
845	26.757811365	192.168.200.2	192.168.200.30	RTP
857	26.777760799	192.168.200.2	192.168.200.30	RTP
858	26.777764015	192.168.200.2	192.168.200.30	RTP
860	26.797980620	192.168.200.2	192.168.200.30	RTP
861	26.797982684	192.168.200.2	192.168.200.30	RTP
863	26.817751814	192.168.200.2	192.168.200.30	RTP
864	26.817754960	192.168.200.2	192.168.200.30	RTP
866	26.837761163	192.168.200.2	192.168.200.30	RTP
867	26.837764419	192.168.200.2	192.168.200.30	RTP
874	26.857765612	192.168.200.2	192.168.200.30	RTP
875	26.857768418	192.168.200.2	192.168.200.30	RTP
877	26.877863109	192.168.200.2	192.168.200.30	RTP
878	26.877865173	192.168.200.2	192.168.200.30	RTP
879	26.888472714	192.168.200.30	192.168.200.2	RTP
880	26.888476311	192.168.200.30	192.168.200.2	RTP
882	26.897759984	192.168.200.2	192.168.200.30	RTP

Figura 91. Ejecución de llamada entre la extensión 1002 y 1003. Elaboración propia.

Continuando con la captura anterior podemos ver la descripción de cada paquete involucrado en la llamada efectuada como podemos observar en la figura 92.

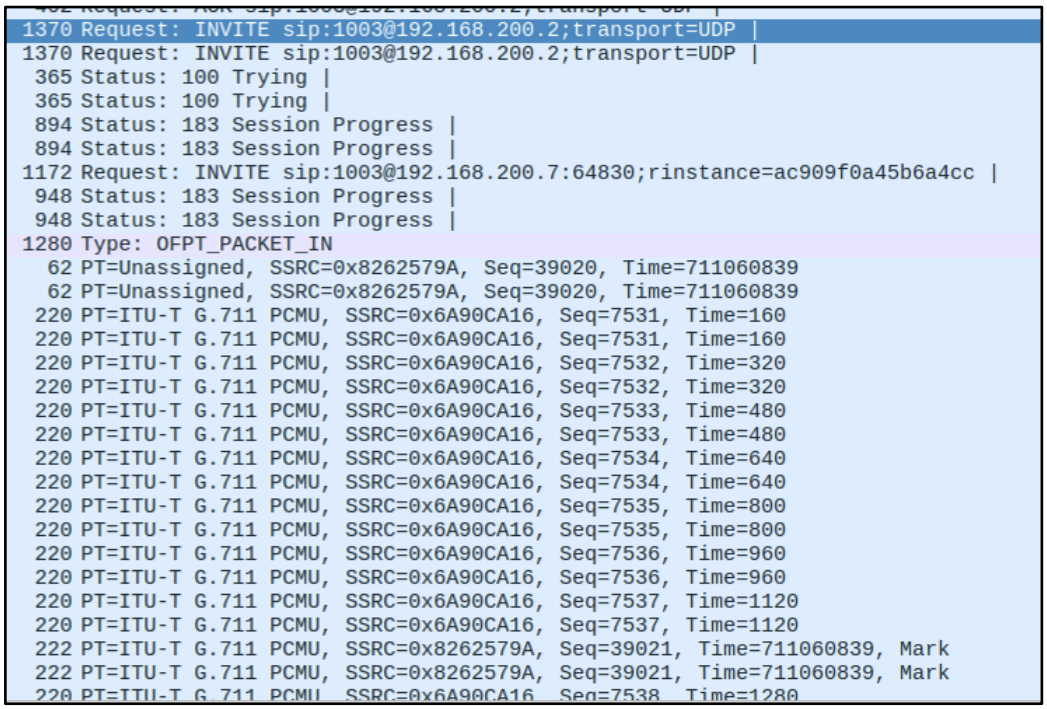


Figura 92. Detalle de la ejecución de llamada entre la extensión 1002 y 1003. Elaboración propia.

Las figuras 91 y 92 presentan la secuencia del establecimiento de una llamada VoIP mediante SIP y la posterior transmisión de audio a través del protocolo RTP. El proceso se inicia cuando la extensión SIP con dirección IP 192.168.200.30 transmite un mensaje *INVITE* hacia la centralita VoIP ubicada en 192.168.200.2, solicitando el establecimiento de comunicación con la extensión 1003.

La centralita ejecuta una respuesta escalonada: primero emite un mensaje provisional 100 *Trying* para confirmar el procesamiento de la solicitud, seguido de un mensaje 183 *Session Progress* que incluye información SDP, habilitando la reproducción temprana de medios previo a la aceptación definitiva de la llamada.

Paralelamente, la centralita reenvía la solicitud *INVITE* al terminal destinatario identificado con la dirección IP 192.168.200.7 para completar el establecimiento del diálogo de comunicación. Durante este proceso, se registra un mensaje *OFPT_PACKET_IN* originado desde el controlador SDN (10.10.0.10), evidenciando que el conmutador *OpenFlow* ha detectado un nuevo flujo RTP y ha notificado al controlador SDN para la gestión dinámica de las reglas del plano de datos.

Completada la negociación SDP, se establece un flujo de audio RTP bidireccional entre la extensión originadora (192.168.200.30) y la centralita (192.168.200.2). La transmisión emplea el códec G.711 PCMU con una generación de paquetes cada 20 ms, resultando en un ancho de banda de 64 kbit/s por cada dirección de transmisión.

Esta secuencia confirma la integración efectiva y la coordinación óptima entre los planos de señalización (SIP), transmisión de medios (RTP) y control de red (*OpenFlow*), asegurando así la calidad y estabilidad del servicio VoIP implementado sobre la infraestructura GPON bajo gestión SDN.

Finalmente, en la figura 93, vemos como se produce la comunicación entre las dos extensiones mencionadas anteriormente.

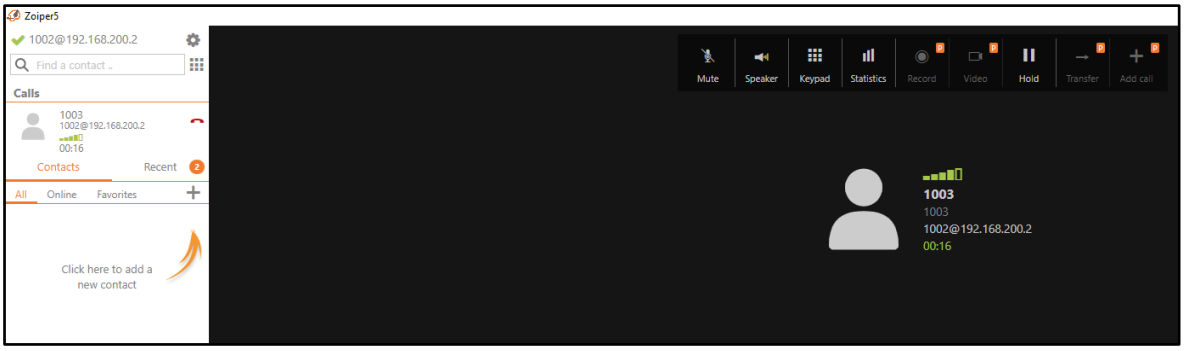


Figura 93. Desarrollo de la llamada entre la extensión 1003 y la 1002. Elaboración propia.

Resultados del servicio de CCTV:

Para evaluar el funcionamiento del servicio del CCTV, se realizó una conexión al servicio desde el abonado correspondiente a la ONT 1, hacia el servicio web de gestión de cámaras instalado en el servidor (NVR). Para ello haremos la siguiente solitud HTTP desde un cliente web por parte del abonado: 192.168.40.2/zm, provocando así el tráfico de la figura 94.

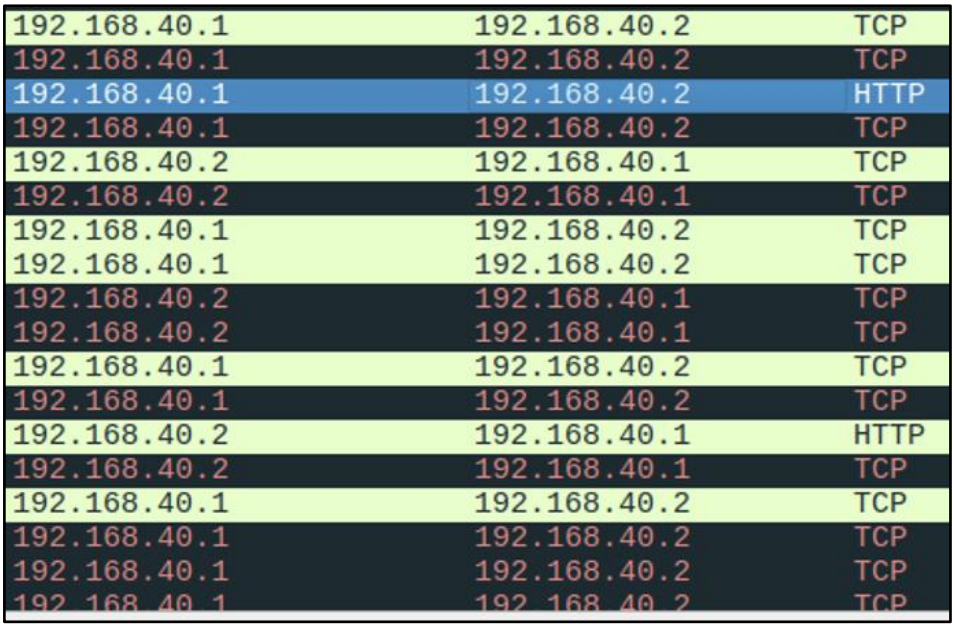


Figura 94. Tráfico generado entre el abonado de la ONT 1 (192.168.40.1) y el servidor NVR del CCTV (192.168.40.2). Elaboración propia.

Consecuente con la figura 94, tenemos la descripción de casa paquete como podemos ver en la figura 95.

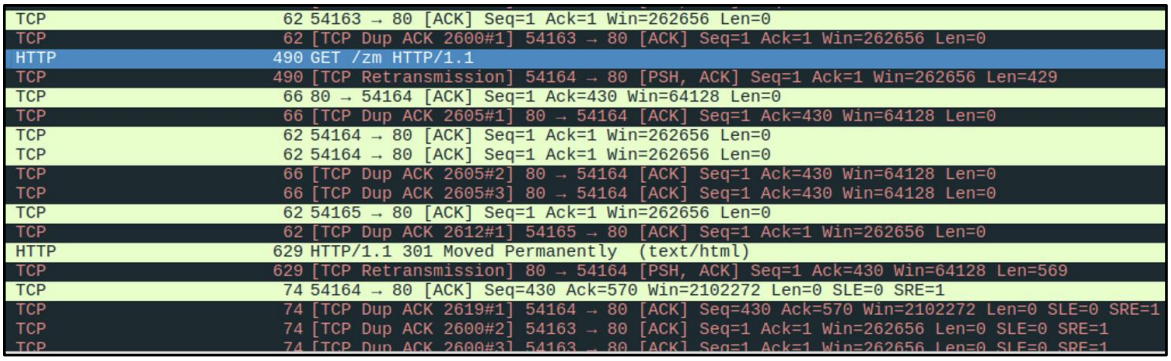


Figura 95. Detalle del tráfico generado entre el abonado de la ONT 1 (192.168.40.1) y el servidor NVR del CCTV (192.168.40.2). Elaboración propia.

En la figura 95 se muestra una captura de tráfico HTTP correspondiente a la VLAN 400, configurada específicamente para el servicio de CCTV dentro de la infraestructura GPON bajo gestión SDN. La secuencia analizada presenta una solicitud HTTP iniciada desde el cliente con dirección IP 192.168.40.1 hacia el servidor web 192.168.40.2, empleando el método *GET* para acceder al recurso */*zm.

Posteriormente, si el abonado final inicia un video en directo o una grabación desde la interfaz gráfica proporcionada para que gestiones las cámaras IP, NVR de *ZoneMinder,* podemos observar en *Wireshark* el tráfico de video en directo en la figura 96.

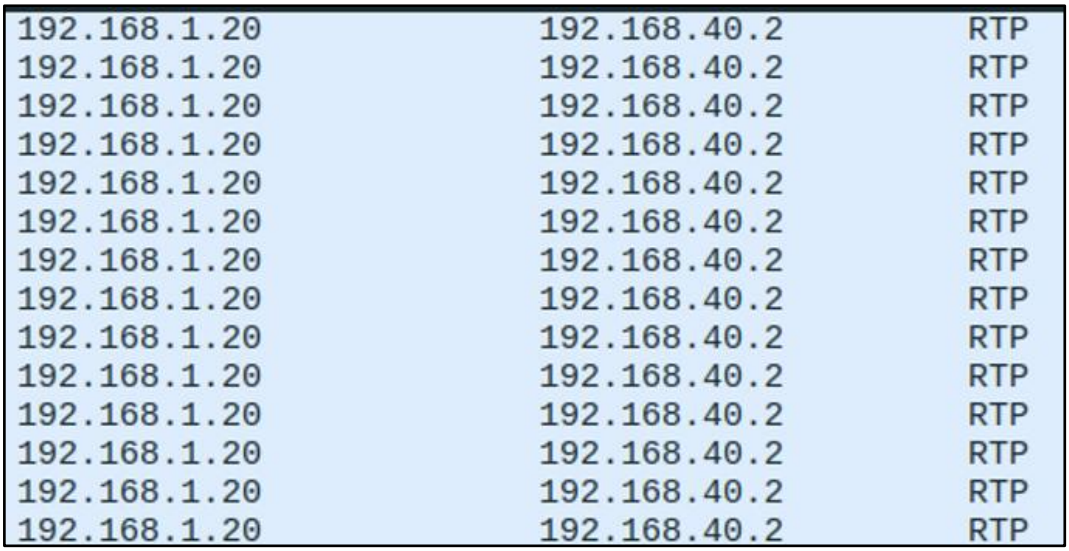


Figura 96. Tráfico generado por la transmisión de vídeo capturado por la cámara IP del abonado en su red local (192.168.1.20) hacia el servidor (192.168.40.2). Elaboración propia.

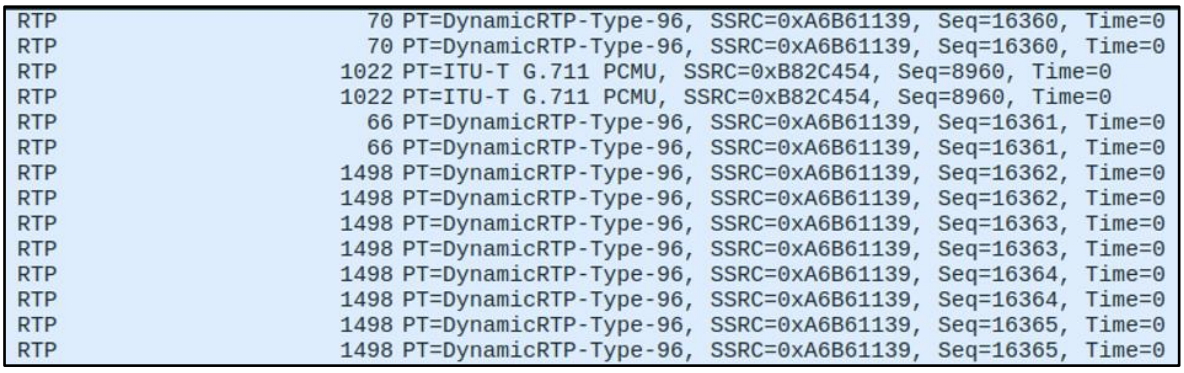


Figura 97. Detalle del tráfico generado por la transmisión de vídeo capturado por la cámara IP del abonado en su red local (192.168.1.20) hacia el servidor (192.168.40.2). Elaboración propia.

La Figura 97 presenta el análisis del tráfico RTP generado durante la visualización en tiempo real de una cámara IP a través de *ZoneMinder*, estableciéndose el flujo desde el dispositivo 192.168.1.20 (red local del abonado) hacia el servidor NVR 192.168.40.2.

Se identifican dos flujos RTP simultáneos: el flujo de video H.264 (*Payload Type* 96, SSRC 0xA6B61139) con paquetes de aproximadamente 1498 bytes que se ajustan a la MTU Ethernet y presentan secuencia numérica continua (Seq=16360-16374) sin pérdidas; y el flujo de audio G.711 μ-law (Payload Type 0, SSRC 0xB82C454) con paquetes menores (≈1022 bytes) coherentes con el códec y muestreo de 20 ms.

Los resultados validan tres aspectos fundamentales: la correcta sincronización audio-video mediante convergencia hacia la misma dirección de destino con puertos UDP negociados por RTSP, permitiendo ensamblaje sin *jitter* perceptible; la eficiencia de la infraestructura GPON-SDN evidenciada por la transmisión sostenida a tasa constante sin retransmisiones ni discontinuidades, demostrando latencias y *buffers* adecuados en la red troncal y conmutadores *OpenFlow*; y el correcto aislamiento por VLAN específica para CCTV, garantizando que el consumo de ancho de banda del flujo H.264 no comprometa la QoS de servicios concurrentes (VoIP, IPTV, navegación web).

En conclusión, el análisis confirma la capacidad de la arquitectura GPON-SDN para soportar transmisión multimedia en tiempo real con calidad profesional, cumpliendo los requisitos operativos para la integración de *ZoneMinder* en el proyecto.

Como resultado del correcto funcionamiento del servicio de CCTV, por parte del usuario final, tenemos el siguiente resultado de ejecución, como se muestra en la figura 98.

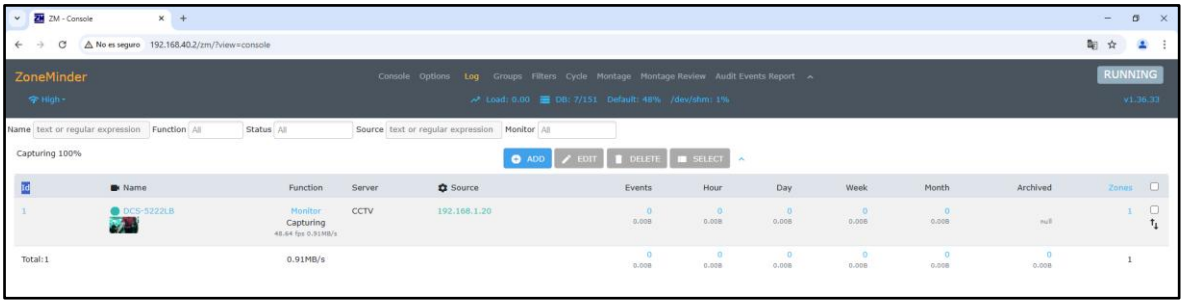


Figura 98. Visión general del NVR del servicio de CCTV, ejecutado por parte del abonado correspondiente a la ONT 1. Elaboración propia.

Siendo el detalle de la figura anterior, la visualización al completo de la cámara IP en cuestión, como se observa en la figura 99.

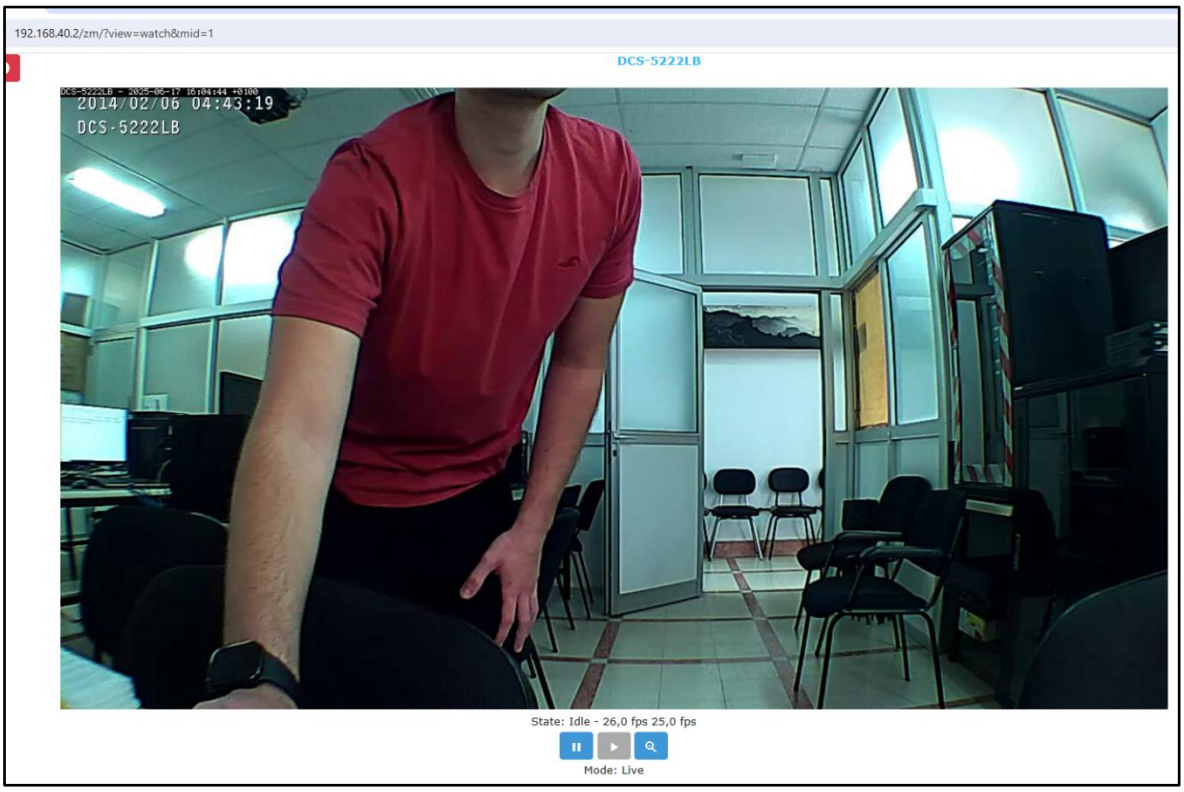


Figura 99. Detalle del video en capturado en modo "live", ejecutado por parte del abonado correspondiente a la ONT 1. Elaboración propia.

Conclusiones generales de los servicios implementados:

A continuación, se presentan las conclusiones generales obtenidas de la evaluación de los servicios implementados sobre la red GPON con gestión SDN:

- ***Servicio de Datos**:*
    - Se confirmó la conectividad directa entre el abonado (192.168.100.2) y el servidor de datos/*router* virtual (192.168.100.1), operando sobre la VLAN 100.
    - Para el acceso a Internet, se implementó con éxito un doble NAT, enmascarando las direcciones IP privadas de la red GPON local con una dirección WAN pública (10.11.204.28) a través del *router* VyOS, lo que permite a los usuarios acceder a cualquier servicio disponible en Internet.

- Se verificó la resolución DNS y el acceso a portales web externos, demostrando la plena funcionalidad del servicio de datos.

- ***Servicio de IPTV**:*

  - La emisión del *stream* de video desde el servidor (192.168.30.10) a la dirección *multicast* (239.1.1.1) y su propagación en la red fue exitosa.

  - El tráfico IGMPv3 *Membership Report* (a 224.0.0.22) validó el correcto funcionamiento de la suscripción y liberación de grupos *multicast* por parte de los abonados (192.168.30.2 en VLAN 300), asegurando la correcta recepción y cese del *stream*.

  - Finalmente, se confirmó la reproducción fluida y correcta del contenido de IPTV en el dispositivo del abonado.

- ***Servicio de VoIP**:*

  - El proceso de registro y des registro SIP (*Session Initiation Protocol)* se realizó correctamente entre las extensiones (ej., 192.168.200.30) y la centralita VoIP (192.168.200.2), incluyendo la autenticación mediante credenciales *Digest* y la supervisión de conectividad con mensajes *OPTIONS*.

  - El establecimiento de llamadas entre extensiones (ej., 1002 y 1003) mostró una secuencia exitosa que involucró mensajes *INVITE***,** 100 *Trying***,** 183 *Session Progress* y SDP.

  - La detección de flujos RTP por parte del conmutador *OpenFlow*, evidenciada por el mensaje *OFPT_PACKET_IN* al controlador SDN (10.10.0.10), confirmó la gestión dinámica de reglas y la coordinación óptima entre los planos de señalización (SIP), transmisión de medios (RTP) y control de red (*OpenFlow*).

  - La transmisión de audio se estableció bidireccionalmente mediante RTP con *códec* G.711 PCMU, asegurando la calidad y estabilidad del servicio VoIP.

- ***Servicio de CCTV**:*

  - La conexión al servicio de gestión de cámaras (NVR) se realizó correctamente a través de tráfico HTTP entre el abonado (192.168.40.1) y

el servidor NVR (192.168.40.2), utilizando la VLAN 400 específica para este servicio.

- La visualización en tiempo real de la cámara IP generó flujos RTP simultáneos de video (H.264) y audio (G.711 μ-law), transmitiéndose eficientemente desde la cámara (192.168.1.20) hacia el servidor NVR.
- Se validó la correcta sincronización audio-video sin *jitter* perceptible, la eficiencia de la infraestructura GPON-SDN para una transmisión sostenida sin retransmisiones ni discontinuidades, y el aislamiento efectivo por VLAN que evita la interferencia con la QoS de otros servicios concurrentes.
- En síntesis, la arquitectura GPON-SDN demostró su capacidad para soportar la transmisión multimedia en tiempo real con calidad profesional, cumpliendo los requisitos operativos para la integración de *ZoneMinder* en el proyecto.

En conjunto, los resultados obtenidos tras la implementación de los servicios ofrecidos sobre la red GPON y su gestión mediante el controlador SDN ONOS validan la viabilidad, eficiencia y potencialidad de la solución propuesta, estableciendo una base sólida para futuras implementaciones similares y líneas de investigación.

6.4. Resultados de integración del controlador SDN de *ONOS*

Una vez analizado los servicios implementados de forma individualizada, se procede a la ejecución de las pruebas funcionales de gestión SDN. Para llevar a cabo esta evaluación, se empleará la interfaz *Swagger* integrada en *ONOS*, la cual proporciona una plataforma web interactiva que facilita la interacción directa con la *API REST* del controlador.

Swagger constituye un *framework* de código abierto que permite documentar, probar y consumir servicios web *RESTful* de manera intuitiva. En el contexto de *ONOS*, esta herramienta se presenta como una interfaz gráfica que expone todas las funcionalidades disponibles a través de la API, permitiendo la ejecución de operaciones de configuración, monitorización y gestión de la red de forma sencilla y sistemática.

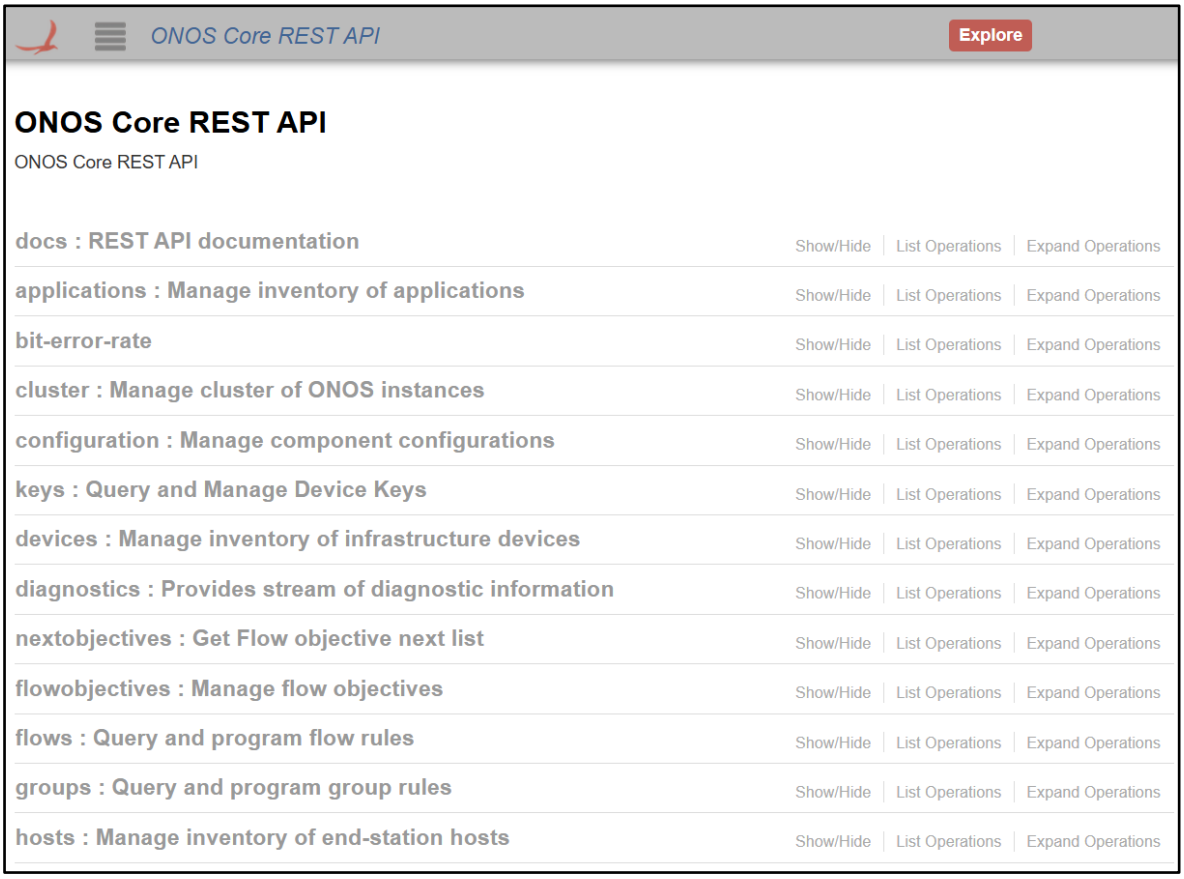


Figura 100. Swagger de ONOS. Elaboración propia.

Limitación de ancho de banda:

En este punto vamos a hacer prueba de limitar el ancho de banda de uno de los abonados de la red GPON. En este caso *ONOS* permite crear *meters OpenFlow* para limitar el caudal de flujos. Por ejemplo, para limitar a ~30 Mbps el tráfico de un abonado GPON, podemos crear un *meter* con unidad en KB/s y una banda de tipo *DROP*. En el entorno dado (*switch* of:000008bfb8f01b1c), el JSON sería:

```
{
  "deviceId": "of:000008bfb8f01b1c",
  "unit": "KB_PER_SEC",
  "burst": true,
  "bands": [
    {
      "type": "DROP",
      "rate": 30000,
      "burstSize": 30000
    }
  ]
}
```

Implementamos el código anterior en el *Swagger*, como se puede observar en la figura 101.

Figura 101. Implementación del meter en Swagger. Elaboración propia.

Añadiéndose correctamente la base de datos del controlador de *ONOS* como se observa a en la figura 102.

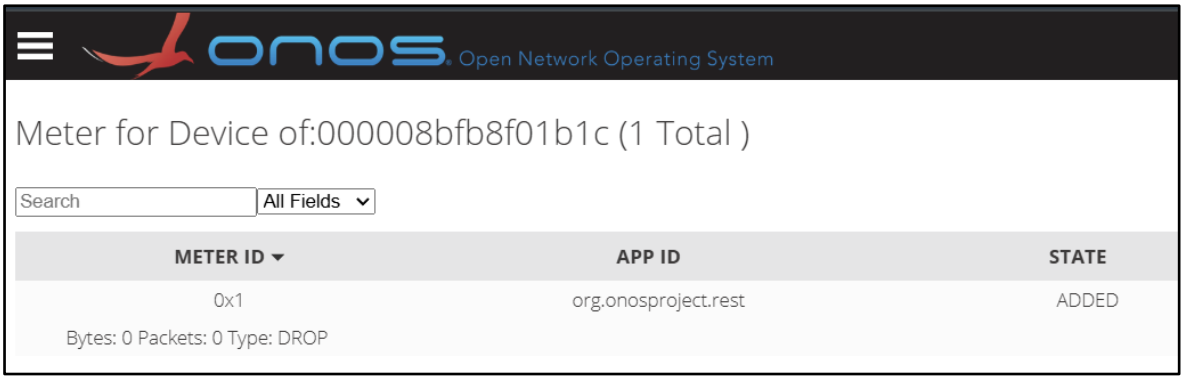


Figura 102. inserción del meter en la base de datos vía API. Elaboración propia.

Este *meter* permitirá al usuario final llegar a una velocidad de conexón de hasta ~30 Mbps y descartará el resto. Para aplicarlo, asociamos el *meter* a una regla de flujo. Por ejemplo, para limitar el tráfico entrante por el puerto 4 (abonados GPON) con VLAN 100 hacia Internet (puerto 2):

```
{
  "flows": [
    {
      "deviceId": "of:000008bfb8f01b1c",
      "priority": 30000,
      "isPermanent": true,
      "treatment": {
        "instructions": [
          { "type": "METER", "meterId": 1 },
          { "type": "OUTPUT", "port": "4" }
        ]
      },
      "selector": {
        "criteria": [
          { "type": "IN_PORT", "port": "2" }
        ]
      }
    }
  ]
}
```

La instrucción *METER* en el flujo aplica la limitación definida. Así, cualquier paquete que llegue por el puerto 2 pasará primero por el medidor y luego saldrá por el puerto 4 con el caudal restringido.

Aplicando el código anterior en el *Swagger*, tenemos como resultado la correcta implementación en la base de datos del controlador, y por tanto en la regla de flujo insertada en el *Open virtual Switch* como podemos observar en la siguiente figura.

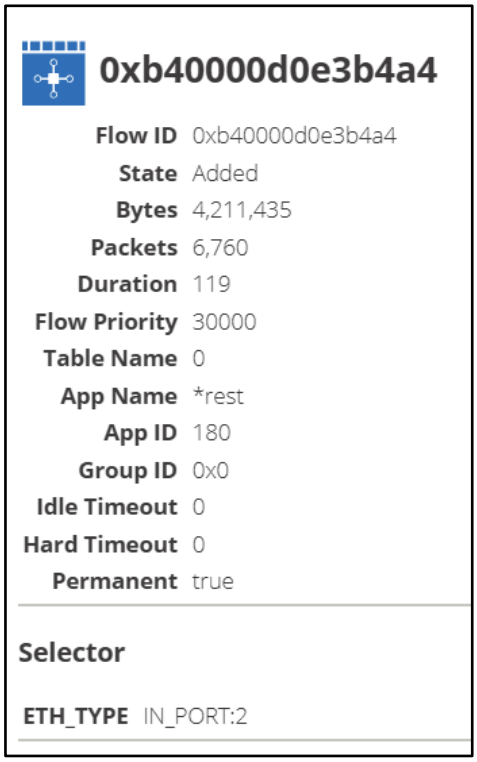


Figura 103. Visualización en UI de la regla de flujo que limita el ancho de banda de la red a 30Mbps. Elaboración propia.

Visualizando lo anterior en *Wireshark,* nos encontramos que la inserción de la regla de flujo sigue la siguiente secuencia de paquetes.

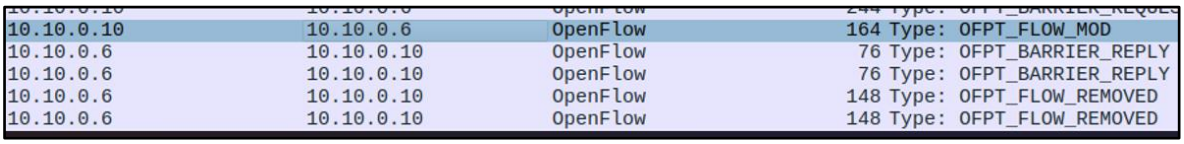


Figura 104. Secuencia de paquetes en inserción de flujo para el protocolo OpenFlow. Elaboración propia.

Que analizando a fondo podemos observar que traza revela la presencia de un mensaje *OFPT_FLOW_MOD (ADD)* caracterizado por el identificador único (*cookie*) 0xb40000d0e3b4a4, el cual corresponde al flujo previamente registrado a través de la *API REST*. La inspección del contenido del paquete confirma la configuración de coincidencia establecida para el puerto de entrada *in_port* 2, así como la implementación de las instrucciones programadas: aplicación del medidor 1 para el control de tasa de transferencia y redirección del tráfico hacia el puerto de salida *OUTPUT*:4.

La configuración incluye adicionalmente el *flag SEND_FLOW_REMOVED*, el cual instruye al conmutador para generar un reporte estadístico al finalizar el ciclo de vida del flujo, garantizando de esta manera la supervisión completa por parte del controlador durante toda la duración de la regla de reenvío, como podemos observar en la figura 105.

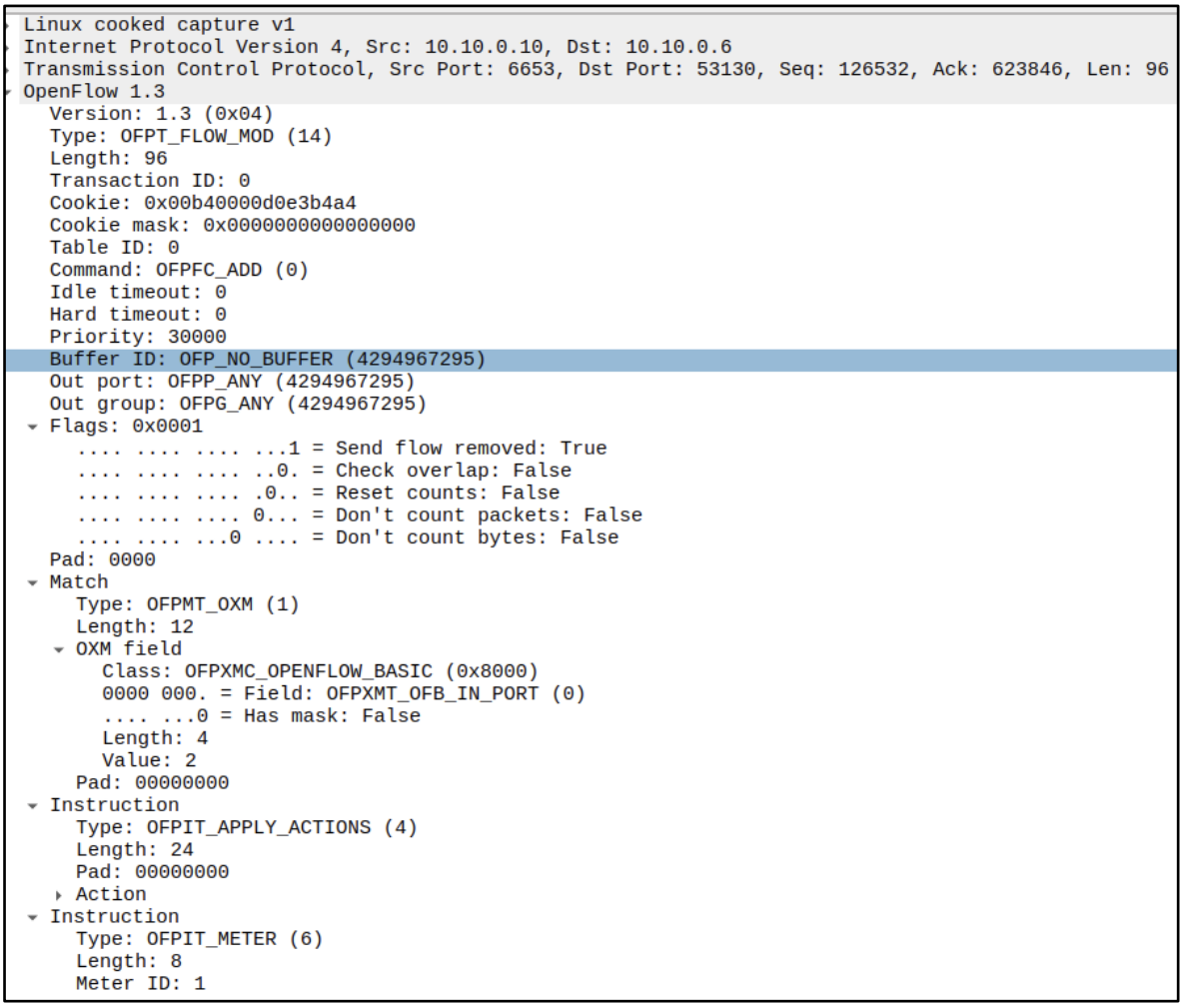


Figura 105. Detalle del paquete OpenFlow (ADD) para la inserción del flujo de limitación de ancho de banda a 30 Mbps. Elaboración propia.

Finalmente, concluyendo con este análisis podemos confirmar que el tráfico anterior corresponde con la secuencia mínima de mensajes *OpenFlow* 1.3 intercambiados entre el controlador *ONOS* (10.10.0.10) y el conmutador OvS (10.10.0.6) durante el procesamiento de una petición *POST /flows* ejecutada por la aplicación *REST*.

La secuencia temporal analizada revela cinco mensajes: inicialmente, *ONOS* transmite un mensaje *OFPT_FLOW_MOD (ADD)* en el instante 390.6223 ms para instalar una regla de

flujo caracterizada por cookie 0xb40000d0e3b4a4, prioridad 30000 en la tabla 0, con instrucciones específicas para aplicar el medidor 1 y redirigir hacia el puerto 4. Posteriormente, OvS responde con dos mensajes *OFPT_BARRIER_REPLY* consecutivos (390.6224 y 390.6225 ms), confirmando la aplicación completa de la regla y garantizando la inmediatez de la operación mediante el mecanismo de barreras implementado por ONOS para delimitar bloques lógicos de configuración.

La secuencia concluye con dos mensajes *OFPT_FLOW_REMOVED* (390.6226 y 390.6227 ms) que notifican la eliminación del flujo, incluyendo estadísticas acumuladas de 163,889 paquetes procesados y 240,450,360 bytes transferidos, confirmando la retirada completa en todos los contextos operativos.

El análisis de esta interacción evidencia tres fases operativas diferenciadas. La fase de inserción comprende la traducción por parte de *ONOS* de la solicitud *REST* a un mensaje *FLOW_MOD (ADD)*, acompañado de un *BARRIER_REQUEST* implícito para garantizar la aplicación inmediata de la regla. La fase de confirmación se materializa mediante las respuestas *BARRIER_REPLY* del conmutador, certificando que la regla está activa y operativa para el reenvío de tráfico. Finalmente, la fase de retirada se ejecuta mediante los mensajes *FLOW_REMOVED*, que proporcionan estadísticas de uso para contabilización y verificación de políticas por parte del controlador.

Esta secuencia demuestra la naturaleza funcional del protocolo *OpenFlow*, donde el mensaje *BARRIER_REPLY* marca la finalización exitosa de la instalación, mientras que *FLOW_REMOVED* certifica el ciclo de vida completo del flujo programado a través de la *API REST*, asegurando la coherencia entre el plano de control (*ONOS*) y el plano de datos (OvS)."

Finalmente, desde el abonado correspondiente a la ONT 1 es posible verificar que el ancho de banda de descarga (*download*) en sentido descendente se limita efectivamente a 30 Mbps, conforme a las especificaciones del flujo insertado desde el controlador hacia el OvS.

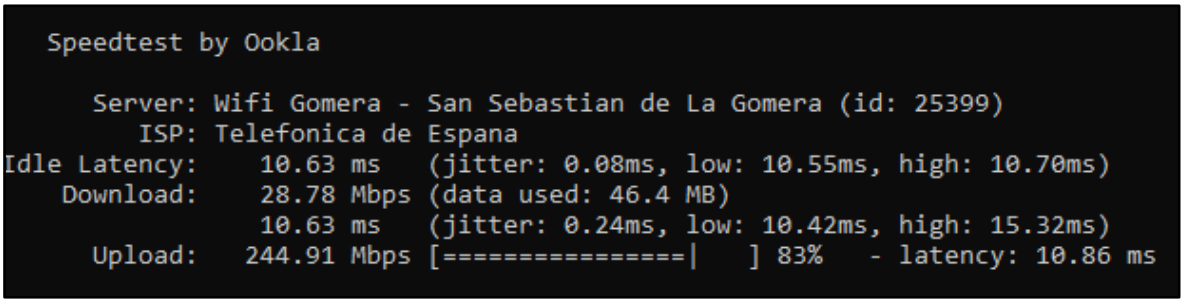


Figura 106. Comprobación de ancho de banda limitado a 30Mbps por parte del abonado correspondiente a la ONT 1. Elaboración propia.

6.4.1 Conclusiones generales de la implementación de ONOS en la red GPON

Validación de la solución propuesta

La implementación de la red GPON bajo la gestión del controlador SDN *ONOS* ha demostrado la viabilidad técnica, eficiencia operativa y potencial de escalabilidad de la solución propuesta, estableciendo una base sólida para futuras implementaciones y líneas de investigación en el ámbito de las redes de acceso definidas por *software*.

Gestión y programación centralizada

La evaluación del controlador ONOS reveló capacidades destacadas en términos de gestión centralizada de la infraestructura de red. La integración nativa de la interfaz *Swagger* proporciona una plataforma web interactiva que facilita la interacción directa con la *API REST* del controlador. Esta funcionalidad permite la exposición completa de las capacidades del sistema mediante interfaces programáticas, habilitando la ejecución sistemática de operaciones de configuración, monitorización y gestión de red. La centralización resultante del control de red automatiza significativamente las tareas operativas tradicionalmente manuales.

Control de tráfico y calidad de servicio

El sistema implementado demostró capacidades avanzadas de control de tráfico mediante la utilización de medidores *OpenFlow*. La validación experimental incluyó la implementación de limitaciones de ancho de banda para un abonado GPON específico, estableciendo un límite de aproximadamente 30 Mbps. La creación de estos medidores se

ejecuta mediante la *API REST* de ONOS, siendo incorporados correctamente en la base de datos del controlador.

La aplicación efectiva de estas limitaciones se materializa mediante la asociación del medidor con reglas de flujo específicas que ONOS inserta en el conmutador *Open vSwitch* (OvS). Esta configuración permite el control bidireccional del tráfico, aplicando las limitaciones definidas de manera precisa. Los resultados experimentales confirmaron que el ancho de banda de descarga del abonado correspondiente a la ONT 1 se limitó efectivamente a 30 Mbps, validando la correcta aplicación del flujo insertado desde el controlador hacia el conmutador.

Coherencia e interacción con el plano de datos

El análisis del tráfico *OpenFlow* reveló que la inserción de reglas de flujo por parte de *ONOS* sigue una secuencia protocolaria específica, observable mediante análisis de tráfico. Esta secuencia evidencia la transmisión de mensajes *OFPT_FLOW_MOD (ADD)* desde ONOS (10.10.0.10) hacia el conmutador OvS (10.10.0.6) para la instalación de reglas de flujo.

La respuesta del conmutador mediante mensajes *OFPT_BARRIER_REPLY* confirma la aplicación completa e inmediata de las reglas, aprovechando el mecanismo de barreras implementado por ONOS para delimitar bloques lógicos de configuración. Adicionalmente, la observación de mensajes *OFPT_FLOW_REMOVED* proporciona notificaciones de eliminación de flujos junto con estadísticas acumuladas, garantizando la supervisión integral por parte del controlador durante el ciclo de vida completo de cada regla.

Esta interacción confirma la naturaleza operacional del protocolo *OpenFlow* y asegura la coherencia operativa entre el plano de control (ONOS) y el plano de datos (OvS).

6.4.2 Otras pruebas de gestión con ONOS

Para complementar la evaluación de las capacidades básicas del controlador *ONOS*, se ejecutaron pruebas adicionales orientadas a validar funcionalidades avanzadas de gestión y seguridad en tiempo real. Estos experimentos permitieron demostrar la versatilidad del sistema implementado más allá de las operaciones convencionales de aprovisionamiento de ancho de banda y distribución de contenido *multicast*.

Las pruebas realizadas se pueden visualizar en el anexo A4 disponible al final del documento.

6.5. Posibles ampliaciones y líneas futuras

Virtualización y uso de contenedores de componentes de red

La evolución natural de este trabajo se centra en la virtualización completa de los componentes de red mediante tecnologías emergentes. Una línea de investigación prioritaria consiste en la implementación de un vOLT (OLT virtual) utilizando *hardware* de propósito general y *software* de código abierto, siguiendo los principios arquitectónicos establecidos por proyectos mencionados como lo es CORD. Esta aproximación permitiría sustituir completamente la OLT física por soluciones nativas de red definida por *software* (SDN), proporcionando mayor flexibilidad operacional y reduciendo la dependencia de *hardware* propietario.

Paralelamente, los servicios virtualizados actualmente desplegados (IPTV, VoIP, CCTV y Datos) constituyen candidatos ideales para su migración hacia arquitecturas basadas en contenedores *Docker* o plataformas de orquestación como *Kubernetes*. Esta transición optimizaría el uso de recursos computacionales y mejoraría la escalabilidad horizontal del sistema. La adopción de tecnologías mediante contenedores facilitaría la automatización del despliegue de funciones de red virtualizadas (VNFs) y su posterior integración con plataformas de gestión NFV MANO, acelerando los ciclos de desarrollo y validación.

Automatización avanzada y gestión inteligente

El ámbito de la automatización presenta oportunidades mediante la integración de herramientas especializadas de orquestación de flujos de trabajo, tales como n8n o *Node-RED*. Estas plataformas permitirían coordinar eventos de red complejos a través de flujos de trabajo sin código. Por ejemplo, n8n podría automatizar procesos críticos como el aprovisionamiento de nuevos dispositivos ONT, la actualización dinámica de políticas de calidad de servicio, o la respuesta automatizada a alarmas mediante integración con las APIs del controlador ONOS y sistemas externos de gestión.

La implementación de *pipelines* de integración y despliegue continuo (CI/CD) específicos para *software* de red representaría un avance hacia un mantenimiento ágil y reproducible. Estos *pipelines* gestionarían automáticamente *scripts* de configuración, repositorios Git y validaciones de integridad, asegurando la coherencia operacional del sistema.

Como extensión natural de las capacidades SDN implementadas, se propone explorar la aplicación de técnicas de inteligencia artificial para la gestión dinámica de recursos de red.

Bibliografía y Referencias

[1] N. Merayo *et al.*, “An Experimental OpenFlow Proposal over Legacy GPONs to Allow Real-Time Service Reconfiguration Policies,” *Applied Sciences*, vol. 11, no. 3, Art. 903, ene. 2021. Accedido: 02 de junio de 2025. [En línea]. Disponible en: https://www.mdpi.com/2076-3417/11/3/903

[2] J. Kani *et al.*, “Disaggregation and Virtualization for Future Access and Metro Networks [Invited Tutorial],” *Journal of Optical Communications and Networking*, vol. 17, no. 1, pp. A1-A12, ene. 2025. Accedido: 02 de junio de 2025. [En línea]. Disponible en: https://opg.optica.org/abstract.cfm?URI=jocn-17-1-A1

[3] Edgecore Networks Corp., “Open Networking for Telecom and Internet Service Providers—CORD Solution Brief,” Hsinchu, Taiwán, white paper, 2024. Accedido: 02 de junio de 2025. [En línea]. Disponible en: https://www.edge-core.com/solution-inquiry.php?cls=5&id=20

[4] K. Shibata *et al.*, “Standardization Trends of Virtualized Access Systems by the Broadband Forum,” *NTT Technical Review*, vol. 16, no. 7, pp. —, jul. 2018. Accedido: 02 de junio de 2025. [En línea]. Disponible en: https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201807gls.html

[5] D. Goovaerts, “OLTs Are the Last Frontier for Network Disaggregation,” *Fierce Network*, 28 feb. 2024. Accedido: 02 de junio de 2025. [En línea]. Disponible en: https://www.fierce-network.com/broadband/olts-are-last-frontier-network-disaggregation

[6] Intel Corporation, “Open vSwitch* Enables SDN and NFV Transformation,” White Paper, Intel Network Builders, dic. 2015. Accedido: 2 de junio de 2025. [En línea]. Disponible en: https://builders.intel.com/docs/open-vswitch-enables-sdn-and-nfv-transformation-paper.pdf

[7] S. S. W. Lee, K.-Y. Li y M.-S. Wu, “Design and Implementation of a GPON-Based Virtual OpenFlow-Enabled SDN Switch,” *IEEE/OSA Journal of Lightwave Technology*, vol. 34, n.° 10, pp. 2552–2561, may 2016. Accedido: 2 de junio de 2025. [En línea]. Disponible en: https://doi.org/10.1109/JLT.2016.2540244

[8] R. F. Moyano, D. Fernández, N. Merayo, C. M. Lentisco y A. Cardenas, “NFV and SDN-Based Differentiated Traffic Treatment for Residential Networks,” *IEEE Access*, vol. 8, pp. 34038–34055, feb. 2020. Accedido: 2 de junio de 2025. [En línea]. Disponible en: https://doi.org/10.1109/ACCESS.2020.2974504

[9] __ Imran, “Software Defined Networking,” *Encyclopedia*, 13 abr. 2021. Accedido: 03 de junio de 2025. [En línea]. Disponible en: https://encyclopedia.pub/entry/8640

[10] J. English, “ONOS (Open Network Operating System),” *TechTarget*, 23 mar. 2023. Accedido: 03 de junio de 2025. [En línea]. Disponible en: https://www.techtarget.com/searchnetworking/definition/ONOS-Open-Network-Operating-System

[11] P. Berde *et al.*, “ONOS: Towards an Open, Distributed SDN OS,” *Proc. ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking* (HotSDN ’14), Chicago, EE. UU., 22 ago. 2014, pp. 1–6. Accedido: 03 de junio de 2025. [En línea]. Disponible en: https://www.researchgate.net/publication/266660316_ONOS_towards_an_open_distributed_SDN_OS

[12] M. N. A. Sheikh, I.-S. Hwang, M. S. Raza y M. S. Ab-Rahman, “A Qualitative and Comparative Performance Assessment of Logically Centralized SDN Controllers via Mininet Emulator,” *Computers*, vol. 13, n.° 4, Art. 85, abr. 2024. Accedido: 03 de junio de 2025. [En línea]. Disponible en: https://www.mdpi.com/2073-431X/13/4/85

[13] D. A. Priano *et al.*, “Comparative Analysis of SDN Controllers: A Study on Installation, Protocols Interaction, Network Topologies Monitoring, and GUI Experience,” *Review of Computer Engineering Studies*, vol. 10, n.° 3, pp. 41–47, sep. 2023. Accedido: 03 de junio de 2025. [En línea]. Disponible en: https://doi.org/10.18280/rces.100302

[14] “OpenFlow,” *Wikipedia, The Free Encyclopedia*, 30 may. 2025. Accedido: 03 de junio de 2025. [En línea]. Disponible en: https://en.wikipedia.org/wiki/OpenFlow

[15] Y.-X. Huang y J. Chou, “A Survey of NFV Network Acceleration from ETSI Perspective,” *Electronics*, vol. 11, no. 9, Art. 1457, may. 2022. Accedido: 03 de junio de 2025. [En línea]. Disponible en: https://www.mdpi.com/2079-9292/11/9/1457

[16] M. Harris, “Defining the Elements of NFV Architectures,” *The Equinix Blog*, 17 oct. 2019. Accedido: 03 de junio de 2025. [En línea]. Disponible en: https://blog.equinix.com/blog/2019/10/17/networking-for-nerds-defining-the-elements-of-nfv-architectures/

[17] B. E. Fernandes, “Transform to a Network of the Future at Your Pace,” *IEEE Softwarization eNewsletter*, nov. 2015. Accedido: 03 de junio de 2025. [En línea]. Disponible en: https://sdn.ieee.org/newsletter/november-2015/transform-to-a-network-of-the-future-at-your-pace

[18] TELNET Redes Inteligentes S.A., *Manual de instalación y configuración SMART OLT Serie 200*, ver. 2, doc. 200080199-02. Zaragoza, España: TELNET Redes Inteligentes S.A., 2015.

[19] TELNET Redes Inteligentes S.A., “ONT GPON WaveAccess 4520 – Especificaciones Técnicas,” ficha técnica, nov. 2018. Accedido: 04 de junio de 2025. [En línea]. Disponible en: https://telnet-fo.es/wp-content/uploads/2018/11/DS_EQ_GPON_WaveAccess_4520_EN.pdf

[20] ITU-T, “Gigabit-capable Passive Optical Networks (GPON): General characteristics,” Recomendación G.984.1, mar. 2008. Accedido: 04 de junio de 2025. [En línea]. Disponible en: https://www.itu.int/rec/T-REC-G.984.1-200803-I

[21] ITU-T, “ONU management and control interface (OMCI) specification,” Recomendación G.988, nov. 2022. Accedido: 04 de junio de 2025. [En línea]. Disponible en: https://www.itu.int/rec/T-REC-G.988-202211-I/en

[22] Broadband Forum, “Using GPON Access in the context of TR-101 (FAN),” TR-156 Issue 4, nov. 2017. Accedido: 04 de junio de 2025. [En línea]. Disponible en: https://www.broadband-forum.org/technical-reports

[23] ITU-T, “Gigabit-capable Passive Optical Networks (GPON): Transmission convergence layer specification,” Recomendación G.984.3, feb. 2004. Accedido: 04 de junio de 2025. [En línea]. Disponible en: https://www.itu.int/rec/T-REC-G.984.3-200402-S

[24] National Institute of Standards and Technology, “Advanced Encryption Standard (AES),” FIPS PUB 197, nov. 2001. Accedido: 04 de junio de 2025. [En línea]. Disponible en: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf nvlpubs.nist.gov

[25] ITU-T, “Rogue optical network unit (ONU) considerations,” Suplemento 49 a la Serie G, sept. 2020. Accedido: 04 de junio de 2025. [En línea]. Disponible en: https://www.itu.int/rec/T-REC-G.Sup49-202009-I

[26] ITU-T, “Gigabit-capable Passive Optical Networks (GPON): Physical Media Dependent layer specification,” Recomendación G.984.2, ago. 2019. Accedido: 04 de junio de 2025. [En línea]. Disponible en: https://www.itu.int/rec/T-REC-G.984.2-201908-I

[27] W. Mamakos *et al.*, “A Method for Transmitting PPP over Ethernet (PPPoE),” RFC 2516, feb. 1999. Accedido: 04 de junio de 2025. [En línea]. Disponible en: https://www.rfc-editor.org/rfc/rfc2516.html

[28] P. Srisuresh y K. Egevang, “Traditional IP Network Address Translator (Traditional NAT),” RFC 3022, ene. 2001. Accedido: 04 de junio de 2025. [En línea]. Disponible en: https://www.rfc-editor.org/rfc/rfc3022.html

[29] R. Droms, “Dynamic Host Configuration Protocol,” RFC 2131, mar. 1997. Accedido: 04 de junio de 2025. [En línea]. Disponible en: https://www.rfc-editor.org/rfc/rfc2131.html

[30] ITU-T, “Gigabit-Capable Passive Optical Networks (G-PON): Transmission Convergence Layer Specification,” Recomendación G.984.3, mar. 2008. Accedido: 04 de junio de 2025. [En línea]. Disponible en: https://www.itu.int/rec/T-REC-G.984.3

[31] M. Holik, T. Horvath y V. Oujezsky, Application for GPON Frame Analysis, *Electronics*, vol. 8, n. º 6, p. 700, jun. 2019. Accedido: 04 de junio de 2025. [En línea]. Disponible en: https://doi.org/10.3390/electronics8060700

[32] D. Hood y E. Trojer, *Gigabit-Capable Passive Optical Networks*. Hoboken, NJ, EE. UU.: Wiley, 2012. Accedido: 04 de junio de 2025. [En línea]. Disponible en: https://onlinelibrary.wiley.com/doi/book/10.1002/9781118211678

[33] E. R. Moraguez, *“Seguridad en Infraestructuras de Redes Virtuales de Funciones de Red (NFV),”* LovTechnology. Accedido: 06 de junio de 2025. [En línea]. Disponible en: https://lovtechnology.com/seguridad-en-infraestructuras-de-redes-virtuales-de-funciones-de-red-nfv/

[34] I. Maximets, *"[ovs-announce] Open vSwitch 3.5.0 is now Available!",* lista de correo ovs-announce, 17 feb. 2025. Accedido: 06 de junio de 2025. [En línea]. Disponible en: https://mail.openvswitch.org/pipermail/ovs-announce/2025-February/000364.html

[35] M. Gupta, *"Introduction to Open vSwitch with DPDK on Arm,"* Arm Community Blog, 13 ago. 2020. Accedido: 06 de junio de 2025. [En línea]. Disponible en: https://community.arm.com/arm-community-blogs/b/tools-software-ides-blog/posts/introduction-to-open-vswitch-with-dpdk-on-arm

[36] Open vSwitch Project, *"ovs-actions(7) — Open vSwitch Actions Manual Page,"* Open vSwitch 3.5.90 Documentation. Accedido: 06 de junio de 2025. [En línea]. Disponible en: https://docs.openvswitch.org/en/latest/ref/ovs-actions.7/#actions

[37] VyOS Project, *"About — VyOS 1.2.x (crux) documentation,"* 2019. Accedido: 06 de junio de 2025. [En línea]. Disponible en: https://docs.vyos.io/en/1.2/introducing/about.html

[38] C. Luengo, "*La Revolución de IPTV en Chile: Transformando la forma en que vemos televisión*," *El Dínamo*, 19 oct. 2023. Accedido: 06 de junio de 2025. [En línea]. Disponible en: https://www.eldinamo.cl/presentado-por/2023/10/19/la-revolucion-de-iptv-en-chile-transformando-la-forma-en-que-vemos-television/

[39] D. G. Barzola, "*IGMP y MLD Snooping en MikroTik RouterOS: Conceptos, Configuración y Ejemplos*," abcXperts Blog, 7 feb. 2024. Accedido: 06 de junio de 2025. [En línea]. Disponible en: https://abcxperts.com/igmp-y-mld-snooping-en-mikrotik-routeros-conceptos-configuracion-y-ejemplos/

[40] Netelip, "*¿Qué es la Telefonía sobre Voz IP o Voz sobre IP?*," Netelip, 2025. Accedido: 06 de junio de 2025. [En línea]. Disponible en: https://www.netelip.com/pe/que-es-la-voz-ip-telefonia-ip/

[41] International Voice-Over Agency, "*Blog – Page 20*," IVA VoiceoverAgency.es, 2025. Accedido: 06 de junio de 2025. [En línea]. Disponible en: https://www.voiceoveragency.es/blog/page/20/

[42] FasterCapital, "*¿Qué es la pérdida de paquetes y cómo afecta la transmisión de datos?*," FasterCapital, 2025. Accedido: 06 de junio de 2025. [En línea]. Disponible en: https://fastercapital.com/es/tema/%C2%BFqu%C3%A9-es-la-p%C3%A9rdida-de-paquetes-y-c%C3%B3mo-afecta-la-transmisi%C3%B3n-de-datos.html

[43] 3CX, "*Qué es un Central Telefónica PBX •• Beneficios*," 3CX, 2025. Accedido: 06 de junio de 2025. [En línea]. Disponible en: https://www.3cx.es/voip-sip/central-telefonica-pbx/

[44] Aselcom, "*VLANs en Redes GPON: Seguridad y Rendimiento*," Aselcom Blog, 2023. Accedido: 06 de junio de 2025. [En línea]. Disponible en: https://aselcom.com/blog/actualidad/vlans-en-redes-gpon

[45] Encyclopædia Britannica, "Closed-circuit television (CCTV)," Encyclopædia Britannica, 2025. Accedido: 09 de junio de 2025. [En línea]. Disponible en: https://www.britannica.com/technology/closed-circuit-television

[46] Isarsoft, "What is a Network Video Recorder (NVR)?" Isarsoft Knowledge Hub, 2024. Accedido: 09 de junio de 2025. [En línea]. Disponible en: https://www.isarsoft.com/knowledge-hub/nvr

[47] GPON.com, “Why GPON.” GPON Resources, s. f. Accedido: 09 de junio de 2025. [En línea]. Disponible en: https://www.gpon.com/why-gpon

[48] IPCent, “Frequently Asked Questions – ONVIF IP Camera.” IPCent, s. f. Accedido: 09 de junio de 2025. [En línea]. Disponible en: https://www.ipcent.com/home/faq

[49] H. Schulzrinne, S. Casner, R. Frederick y V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” *IETF RFC 3550*, 2003. Accedido: 09 de junio de 2025. [En línea]. Disponible en: https://www.rfc-editor.org/rfc/rfc3550

[50] Cisco Systems, “Understand GPON Technology,” Cisco Support Document ID 216230, 2023. Accedido: 09 de junio de 2025. [En línea]. Disponible en: https://www.cisco.com/c/en/us/support/docs/switches/catalyst-pon-series/216230-understand-gpon-technology.html

[51] Coram AI, “VMS (Video Management System).” Coram AI Glossary, s. f. Accedido: 09 de junio de 2025. [En línea]. Disponible en: https://www.coram.ai/glossary/vms-video-management-system

[52] S. Gorski, “The Truth about VLANs,” *Security Today*, vol. 11, núm. 8, 2013. Accedido: 09 de junio de 2025. [En línea]. Disponible en: https://securitytoday.com/articles/2013/08/01/the-truth-about-vlans.aspx

[53] TP-Link, “Unified GPON Network and Surveillance Solution for Hotel Turia Valencia,” Case Study, 2025. Accedido: 09 de junio de 2025. [En línea]. Disponible en: https://static.tp-link.com/Case%20Study_Hotel%20Turia%20Valencia_2025(EN).pdf

[54] FTTH Council Europe y IDATE DigiWorld, “Market Forecast 15 – Remaining Homes to be Passed with FTTH/B in 2026,” en *FTTH Forecast for Europe: Market Forecasts 2021-2026*, presentación mostrada en la FTTH Virtual Conference 2021, Bruselas, Bélgica, 15-16 sep. 2021. Accedido: 10 jun. 2025. [En línea]. Disponible en: https://www.sipotra.it/wp-content/uploads/2022/02/Forecast-for-Europe-2021-2026.pdf

[55] Open Networking Foundation, “CORD (Central Office Re-architected as a Datacenter),” 2025. Accedido: 10 de junio de 2025. [En línea]. Disponible en: https://opennetworking.org/cord/

[56] Open Networking Foundation, “VOLTHA Component Diagram,” en *VOLTHA Documentation – Architecture Overview*, ver. 2.13.1, 2024. Accedido: 10 de junio de 2025. [En línea]. Disponible en: https://docs.voltha.org/master/overview/architecture_overview.html

[57] W. Braun y M. Menth, “Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices,” *Future Internet*, vol. 6, n. 2, pp. 302-336, mayo 2014, Fig. 1 “A three-layer software-defined networking (SDN) architecture.” Accedido: 10 jun. 2025. [En línea]. Disponible en: https://www.researchgate.net/figure/A-three-layer-software-defined-networking-SDN-architecture_fig1_284696928

[58] L. L. Peterson, C. Cascone, B. O’Connor, T. Vachuska y B. Davie, “Fig. 30 – Three-layer ONOS architecture (Northbound APIs, Distributed Core, Southbound Plugins),” en *Software-Defined Networks: A Systems Approach*, cap. 6 “Network OS”, versión en línea 2.1-dev, Systems Approach LLC, 2024. Accedido: 10 jun. 2025. [En línea]. Disponible en: https://sdn.systemsapproach.org/onos.html

[59] K. S. Sahoo, S. Mohanty, M. Tiwary, B. K. Mishra y B. Sahoo, “A Comprehensive Tutorial on Software Defined Network: The Driving Force for the Future Internet Technology,” en *Proc. 6th Int. Conf. on Advanced Computing & Communication Technologies (AICTC)*, Bikaner, India, ago. 2016. Accedido: 11 de junio de 2025. [En línea]. Disponible en: https://www.researchgate.net/publication/309259552_A_Comprehensive_Tutorial_on_Software_Defined_Network_The_Driving_Force_for_the_Future_Internet_Technology

[60] Y. Luk, “NFV Architectural Framework: The ETSI architectural framework explained,” STL Partners, Network Innovation (artículo en línea). Accedido: 10 de junio de 2025. [En línea]. Disponible en: https://stlpartners.com/articles/network-innovation/nfv-architectural-framework/

[61] TELNET Redes Inteligentes S.A., *GPON OLT SmartOLT 240 – Datasheet*, Ref. 180420, Zaragoza, España, ene. 2018. Accedido: 11 jun. 2025. [En línea]. Disponible en: https://www.telnetfo.es/wpcontent/uploads/2018/01/DS_EQ_GPON_SmartOLT_240_EN.pdf

[62] TELNET Fiber Optic S.L., “GPON Management System (TGMS),” página de producto, 19 mar. 2018. Accedido: 11 jun. 2025. [En línea]. Disponible en: https://www.telnet-fo.es/en/telnet-gpon-management-system/

[63] TELNET Redes Inteligentes S.A., *ONT GPON WaveAccess 4520 – Datasheet*, ref. 180920, Zaragoza, España, sep. 2018. Accedido: 11 jun. 2025. [En línea]. Disponible en: https://telnet-fo.es/wpcontent/uploads/2018/11/DS_EQ_GPON_WaveAccess_4520_EN.pdf

[65] TELNET Redes Inteligentes S.A., “Transporte en el canal descendente,” diapositiva 24 de la presentación *GPON: Introducción y Conceptos Generales*, versión 1.7, Zaragoza, España, oct. 2014. Accedido: 11 jun. 2025. [En línea]. Disponible en: https://telnet-fo.es/wp-content/uploads/2014/10/gpon-introduccion-conceptos.pdf

[66] Open vSwitch Project, “Open vSwitch: a production-quality multilayer virtual switch (overview page),” 2025. Accedido: 11 jun. 2025. [En línea]. Disponible en: https://www.openvswitch.org/

[67] A. Chiao, “Fig. 2.1 – OVS Architecture,” en *OVS Deep Dive 0: Overview*, ArthurChiao’s Blog, 31 dic. 2016. Accedido: 12 jun. 2025. [En línea]. Disponible en: http://arthurchiao.art/blog/ovs-deep-dive-0-overview/

[68] P. Emmerich, D. Raumer, S. Gallenmüller, F. Wohlfart y G. Carle, “Throughput and Latency of Virtual Switching with Open vSwitch: A Quantitative Analysis,” *Journal of Network and Systems Management*, vol. 26, n. 4, pp. 938-966, abr. 2018. Accedido: 12 jun. 2025. [En línea]. Disponible en: https://www.researchgate.net/publication/318604719_Throughput_and_Latency_of_Virtual_Switching_with_Open_vSwitch_A_Quantitative_Analysis

[69] VyOS Project, “VyOS Platform Datasheet: The Universal Networking Platform,” ver. 3.0, 2024. Accedido: 09 de junio de 2025. [En línea]. Disponible en: https://vyos.io/files/vyos-datasheet-v3.pdf

[70] Tonmind, “The Session Initiation Protocol (SIP),” 2 nov. 2021. Accedido: 14 de junio de 2025. [En línea]. Disponible en: https://es.tonmind.com/blog/the-session-initiation-protocol-sip_b11

[71] Docker Inc., “Docker Hub,” 2025. Accedido: 14 de junio de 2025. [En línea]. Disponible en: https://hub.docker.com/

[72] Open vSwitch Project, “Distributions packaging Open vSwitch – Debian/Ubuntu,” 2025. Accedido: 02 de junio de 2025. [En línea]. Disponible en: https://docs.openvswitch.org/en/latest/intro/install/distributions/#debian-ubuntu

[73] IBM Corporation, “Preparing a virtual switch,” 2019. Accedido: 02 de junio de 2025. [En línea]. Disponible en: https://www.ibm.com/docs/en/linux-on-systems?topic=devices-virtual-switch

[74] Open vSwitch Project, “ovs-vsctl(8): utility for querying and configuring ovs-vswitchd,” 2025. Accedido: 02 de junio de 2025. [En línea]. Disponible en: http://www.openvswitch.org/support/dist-docs/ovs-vsctl.8.txt

[75] VyOS, “Installation — VyOS 1.5.x (circinus) documentation,” 2025. Accedido: junio de 2025. [En línea]. Disponible en: https://docs.vyos.io/en/latest/installation/install.html

[76] Canonical, “Get Ubuntu Desktop,” 2025. Accedido: junio de 2025. [En línea]. Disponible en: https://ubuntu.com/download/desktop

[77] VideoLAN, “Stream over HTTP — VLC Desktop User Documentation 3.0,” 2025. Accedido: junio de 2025. [En línea]. Disponible en: https://docs.videolan.me/vlc-user/desktop/3.0/en/advanced/streaming/stream_over_http.html

[78] ZoneMinder Project, “Ubuntu Installation Guide,” 2025. Accedido: junio de 2025. [En línea]. Disponible en: https://zoneminder.readthedocs.io/en/latest/installationguide/ubuntu.html

[79] Suprema Inc., “How to Install Local SIP Server (FreePBX Asterisk) in VirtualBox,” 2022. Accedido: junio de 2025. [En línea]. Disponible en: https://support.supremainc.com/en/support/solutions/articles/24000081864

[80] TP-Link, “How to Setup OpenVPN on TP-Link Routers (Windows),” 2022. Accedido: junio de 2025. [En línea]. Disponible en: https://www.tp-link.com/support/faq/1239/

[81] OpenVPN Inc., “How to Install OpenVPN Connect on Windows,” 2024. Accedido: junio de 2025. [En línea]. Disponible en: https://openvpn.net/connect-docs/installation-guide-windows.html

[82] Ubuntu, “How to install and use OpenVPN,” 2023. Accedido: junio de 2025. [En línea]. Disponible en: https://documentation.ubuntu.com/server/how-to/security/install-openvpn/index.html

[83] Y. K. Yin, “How to setup OpenVPN client,” 2014. Accedido: junio de 2025. [En línea]. Disponible en: https://askubuntu.com/questions/460871/how-to-setup-openvpn-client

[84] ONOS Project, “Appendix B: REST API,” 2025. Accedido: junio de 2025. [En línea]. Disponible en: https://wiki.onosproject.org/display/ONOS/Appendix+B%3A+REST+API

[85] Wireshark Foundation, “7.10.2: Checksum Offloading,” *Wireshark User’s Guide*, 2025. Accedido: junio de 2025. [En línea]. Disponible en: wireshark.org

[86] Wireshark Wiki, “Capture Setup / Offloading,” 2025. Accedido: junio de 2025. [En línea]. Disponible en: wiki.wireshark.org

[87] Stack Overflow, “IGMP Join message gives wrong checksum,” 2013. Accedido: junio de 2025. [En línea]. Disponible en: stackoverflow.com

Pliego de condiciones

PI.1. Introducción

Este apartado trata los elementos relacionados con las autorizaciones de uso, los derechos de propiedad intelectual y las obligaciones correspondientes. Asimismo, se especifican las circunstancias bajo las cuales se ha realizado la elaboración del Trabajo de Fin de Grado y las condiciones que deben cumplirse para la utilización del escenario de red desarrollado.

PI.2. Condiciones *hardware*

Durante la elaboración de este Trabajo de Fin de Grado, se utilizaron los componentes de *hardware* que se detallan en la Tabla 16.

Tabla 16. Componentes hardware utilizados en el TFG.

Denominación	Modelo	Fabricante
OLT	SmartOLT 240	TELNET
ONT	WaveAccess 4520	TELNET
SFP OLT de óptica C+	SFP OLT 1 Gbps	Lopacan
Splitter 1x8 g657a1 2mm 1,5m In:sc/upc -1,5m Out: sc/apc	Electropolis	Electropolis
Splittter PLC 1x8 1x8 sc/apc sc/apc	Electropolis	Electropolis
Acoplador simple SC	Elfcam	Elfcam
Ordenador	64 GB de RAM	TicNOVA
Ordenador	optiplex 7010	Dell
Monitor 19"	LCD HD Dell 1909WB	Dell
Monitor 23,8"	1080p 100Hz IPS Flicker Free Low Blue Light VESA HDMI VGA	Dell
Teclado	Español	Dell
Switch de escritorio de 5 puertos a 10/100/1000Mbps	TL-SG105S	Tp-link
USB 3.0 to Gigabit	UE300	Tp-link
Cámara IP	DCS-5222L	D-Link
Ordenador portátil	Vivobook Pro 15	Asus
Router-VPNServer	Archer AX55	TP-Link

PI.3. Condiciones *software*

Tabla 17. Software utilizado en el TFG.

Aplicación	Versión
Sistema Operativo Windows 11	23H2
Controlador *ONOS*	2.7.0 LTS
Docker Desktop	4.41.2
Sistema Operativo W*indows* 10	22H2
Open Virtual Switch	3.3.5 LTS
Router VyOS	1.3
Asterisk Free PBX	16
ZoneMinder	1.32
OpenVPN en sistema Windows	3.6.0
OpenVPN en sistema Ubuntu	2.6.12
VLC	4.0
Sistema operativo Ubuntu Desktop	24.04.2 LTS
TGMS	V2.1.3
VirtualBox	7.1

PI.4. Condiciones de uso por parte del administrador de red

Para el correcto funcionamiento del sistema, el administrador de red debe cumplir con requisitos técnicos específicos que garanticen la conectividad y seguridad del entorno. Es necesario establecer una conexión VPN para acceder al *Open vSwitch* (OvS), ya que esta proporciona un canal seguro y cifrado para las comunicaciones de control. Adicionalmente, se requiere tener una instancia del controlador *ONOS* instalada localmente en el equipo del administrador. La arquitectura de conexión se establece mediante la VPN, que permite conectar el controlador *ONOS* local con el OvS remoto, creando un enlace de control seguro. Esta configuración asegura que todas las operaciones de administración y monitorización se realicen a través de canales protegidos. El administrador debe mantener activa la conexión VPN durante todas las sesiones de gestión y garantizar que la instancia de *ONOS* esté correctamente configurada para comunicarse con el *switch* virtual.

PI.5. Condiciones de licencia

La infraestructura de red implementada en este TFG pertenece a la ULPGC. Toda persona o institución que requiera utilizar completa o parcialmente dicha infraestructura deberá cumplir con los términos especificados en la presente licencia. La implementación del sistema o su instalación en entornos externos necesitará aprobación explícita del autor, del TFG y de la Escuela de Ingeniería de Telecomunicación y Electrónica de la ULPGC.

PI.6. Derechos de autor

Tanto la configuración fuente como la documentación asociada a la red GPON-SDN están protegidos por la legislación vigente en materia de propiedad intelectual, así como por las disposiciones de los tratados internacionales aplicables. El sistema desarrollado se considera un producto sujeto a derechos de autor. No obstante, se permite su uso o reproducción bajo autorización expresa del autor, del tutor del TFG y de la Escuela de Ingeniería de Telecomunicación y Electrónica de la ULPGC.

PI.7. Restricciones

Queda prohibida la aplicación de técnicas de ingeniería inversa. Se autoriza la cesión de la configuración a terceras partes solamente cuando el cedente no mantenga copias del material transferido, incluidas versiones modificadas, actualizadas o documentación adicional.

PI.8. Garantía

El autor del TFG presenta la red GPON-SDN en su estado actual (tal como está), sin ofrecer garantías de ningún tipo, ya sean expresas o implícitas. No asume responsabilidad por posibles perjuicios directos o indirectos que puedan originarse del uso de la configuración, la documentación o cualquier otro elemento relacionado. No se asegura la precisión, confiabilidad ni aptitud de la red para propósitos particulares, tampoco se garantiza que su operación esté exenta de fallos. La configuración no ha sido concebida para su implementación en ambientes críticos o de alta exposición al riesgo que demanden resistencia a errores. Por consiguiente, se declina de manera categórica cualquier garantía sobre la idoneidad de la red para actividades que conlleven riesgos para equipamiento, infraestructuras o individuos.

PI.9. Limitaciones de responsabilidad

Ni el autor del TFG, ni el tutor, ni la Escuela de Ingeniería de Telecomunicación y Electrónica de la ULPGC asumirán responsabilidad alguna, bajo ninguna circunstancia, por daños directos, indirectos, incidentales, derivados o especiales, incluyendo pérdidas financieras, interrupciones operativas, pérdida de información, beneficios no realizados o cualquier otra repercusión que resulte del uso o de la incapacidad de usar la configuración de la red GPON-SDN y la documentación suministrada. El usuario acepta de forma voluntaria todos los riesgos vinculados a la implementación de la red y consiente las condiciones, términos y limitaciones establecidas. No se otorga ninguna garantía complementaria más allá de las ya claramente especificadas en este documento.

PI.10. Otras consideraciones

En el supuesto de que cualquiera de las cláusulas contenidas en esta licencia sea considerada, completa o parcialmente, nula o inexigible, dicha disposición será ajustada de manera apropiada para hacerla válida y aplicable, sin comprometer la vigencia del resto del contenido de la licencia. Este acuerdo se encuentra bajo el amparo de la normativa legal española, y cualquier disputa que surja de su interpretación o ejecución quedará sometida a la competencia exclusiva de los juzgados españoles. El usuario manifiesta estar informado y dar su conformidad a estas condiciones de forma íntegra.

Presupuesto

P1. Introducción

Para el cálculo del presupuesto se ha dividido el capítulo de la siguiente manera:

- Recursos materiales.
- Trabajo tarifado por tiempo empleado.
- Redacción del documento.
- Derechos de visado del COITT.
- Gastos de tramitación y envío.
- Material fungible.

Después de analizar los cada uno de los puntos anteriores, se calcula el coste final del proyecto incluyendo impuestos.

P2. Recursos materiales

Para la amortización de los recursos materiales se tiene en cuenta el conjunto de recursos *hardware* y *software* empleados en el desarrollo de este TFG. Para calcular el coste, se ha empleado el método de depreciación lineal, el cual distribuye equitativamente la pérdida de valor de estos elementos a lo largo de la vida útil estimada. La vida útil estándar considerada es de cuatro años, pero este trabajo se realizó en un periodo de 4 meses, requiriendo así un ajuste proporcional para reflejar el periodo específico de uso.

Hardware

Como se ha mencionado anteriormente, este TFG se ha realizado en un periodo de cuatro meses, siendo un tiempo mucho menor al de cuatro años, el cual es considerado como base para calcular la depreciación de los equipos físicos. Por esta causa se ha tomado la relación entre los dos periodos para el cálculo de la amortización:

$$\text{Vida útil (meses)} = 4 \text{ años} * 12 \text{ meses}/1 \text{ año} = 48 \text{ meses}$$

$$\text{Valor amortizado} = 4 \text{ meses}/48 \text{ meses} * \text{Valor de adquisición}$$

En la tabla 18, se detalla los elementos *hardware* utilizados en este proyecto, indicando el precio de compra original como la porción de su valor que se ha depreciado durante el periodo desarrollado.

Tabla 18. Amortización de los recursos hardware.

Denominación	Coste/ud.	Cantidad	Total	Amortización
SmartOLT 240 OLT GPON	1.702,35 €	1	1.702,35 €	141,86 €
ONT GPON WaveAccess 4520	60,85 €	2	121,70 €	10,14 €
SFP OLT de óptica C+	38,39 €	1	38,39 €	3,20 €
Splitter 1x8 g657a1 2mm 1,5m In:sc/upc -1,5m Out: sc/apc	17,57 €	1	17,57 €	1,46 €
Splittter PLC 1x8 1x8 sc/apc sc/apc	11,25 €	2	22,50 €	1,88 €
Acoplador simple SC	2,00 €	1	2,00 €	0,17 €
Ordenador	439,98 €	1	439,98 €	36,67 €
Ordenador	562,47 €	2	1.124,94 €	93,75 €
Monitor 19"	150,00 €	1	150,00 €	12,50 €
Monitor 23,8"	145,20 €	2	290,40 €	24,20 €
Teclado	15,00 €	3	45,00 €	3,75 €
TL-SG105S Switch de escritorio de 5 puertos a 10/100/1000Mbps	20,74 €	1	20,74 €	1,73 €
USB 3.0 to Gigabit	17,41 €	1	17,41 €	1,45 €
Cámara ip D-Link	179 €	1	179 €	14,92 €
Portatil Asus Vivobook Pro 15	839,29 €	1	839,29 €	69,94 €

El coste total de amortización de los materiales físicos es de cuatrocientos diecisiete con setenta y dos euros (417,72 €).

Software

Las herramientas *software* utilizadas en este proyecto han sido de carácter gratuito y *open source*:

- ***Open Virtual Switch:*** *Software* de Código Abierto, por lo que no se requirió pago de licencias por tanto los costes de amortización serán nulos.

- ***Windows 10 y 11:*** Las licencias ya venían instaladas en los equipos utilizados, por lo que no se requirió pago de licencias por tanto los costes de amortización serán nulos.

- ***Controlador ONOS:*** *Software* de código abierto descargado de forma gratuita mediate, *Docker Hub*, por lo que no se requirió pago de licencias por tanto los costes de amortización serán nulos.

- ***Docker Desktop:*** *Software* gratuito, por lo que no se requirió pago de licencias por tanto los costes de amortización serán nulos.

- ***Virtual Box:*** *Software* gratuito, por lo que no se requirió pago de licencias por tanto los costes de amortización serán nulos.

- ***Servicios virtualizados:*** *Software* gratuitos, por lo que no se requirió pago de licencias por tanto los costes de amortización serán nulos.

- ***Ubuntu Desktop:*** *Software* gratuito, por lo que no se requirió pago de licencias por tanto los costes de amortización serán nulos.

P3. Trabajo tarifado por tiempo empleado

La elaboración de este proyecto ha requerido aproximadamente 300 horas de trabajo, distribuidas entre las fases de diseño, desarrollo y elaboración de la documentación técnica. Conforme a las directrices establecidas por el COITT (Colegio Oficial de Ingenieros Técnicos en Telecomunicación), la valoración económica del trabajo desarrollado se calcula aplicando la siguiente fórmula:

$$H = Ct * 74{,}88 * Hn + Ct * 96{,}72 * He$$

Donde:

- ***H**:* Importe total de honorarios correspondientes al proyecto
- ***Ct**:* Coeficiente de ajuste en función de las horas empleadas
- ***Hn**:* Horas desarrolladas durante jornada laboral ordinaria
- ***He**:* Horas desarrolladas en horario extraordinario (para este proyecto su valor es 0, al no haberse registrado)

Considerando los baremos establecidos por el COITT, el coeficiente de ajuste correspondiente a las horas empleadas, según se especifica en la Tabla 19, equivale a 0,60.

Tabla 19. Coeficientes para el cálculo de los honorarios.

Horas empleadas	Factor de corrección Ct
$X < 36$	1
$36 < X < 72$	0,90
$72 < X < 108$	0,80
$108 < X < 144$	0,70
$144 < X < 180$	0,65
$180 < X < 360$	0,60
$360 < X < 540$	0,55

De acuerdo con la tabla 19, al completarse este proyecto en 300 horas, le corresponde aplicar el coeficiente de ajuste de 0,60. Con esto, la fórmula anterior queda de la siguiente forma:

$$H = 0{,}6 * 74{,}88 * 300 + 0.6 * 96{,}72 * 0 = 13.478{,}40\,€$$

Por tanto, los honorarios resultantes del tiempo invertido sin incluir presupuesto ascienden a un total de trece mil cuatrocientos setenta y ocho euros con cuarenta céntimos (13.478,40 €).

P4. Redacción del documento

El importe correspondiente a la redacción del TFG se determina conforme a la fórmula establecida por el COITT, como se muestra a continuación:

$$R = 0{,}07 * P * C_n$$

Donde:

- ***R:*** son los honorarios por la redacción del TFG.
- ***P:*** representa el presupuesto parcial acumulado hasta este punto: 13.896,12 €.
- C_n: es el coeficiente de ponderación, que según el COITT tiene un valor de 1 para presupuestos menores de 30.050 €.

Sustituyendo:

$$R = 0{,}07 * 13.896{,}12 * 1$$

Por tanto, el coste estimado libre de impuestos correspondiente a la redacción del TFG asciende a los novecientos setenta y dos con setenta y dos euros (972,72 €).

P5. Derechos de visado del COITT

Los gastos derivados del visado colegial del presente TFG se calculan conforme a las tarifas establecidas por el COITT, mediante la siguiente ecuación:

$$V = 0{,}006 * P * C_v$$

Donde:

- ***V*** es el coste de visado del trabajo.
- ***P*** es el presupuesto del TFG.
- C_v el coeficiente reductor en función del presupuesto del trabajo.

Considerando los valores calculados anteriormente:

- Presupuesto parcial: 13.896,12 €
- Redacción del TFG: 972,72 €
- Presupuesto total acumulado (***P***): 14.868,84 €

Según lo establecido por el COITT, para presupuestos inferiores a 30.050 €, el coeficiente C_v es igual a 1.

$$V = 0{,}006 * 14.868{,}84 * 1 = 89{,}21 €$$

Por tanto, el coste libre de impuestos asociado a los derechos de visado del COITT asciende a ochenta y nueve con veintiún (89,21 €).

P6. Gastos de tramitación y envío

Los gastos de tramitación y documentos están estipulados en seis euros (6,00 €).

P7. Material fungible

En este proyecto no se contempla ningún gasto por edición de documentos ni material de oficina, por lo que el coste asociado al material fungible es de cero euros (0 €)

P8. Coste final del proyecto

El presupuesto total estimado del proyecto, calculado en base a los diferentes recursos materiales y humanos empleados, asciende a 14.964,05 €, importe libre de impuestos. A esta cantidad se le aplica el Impuesto General Indirecto Canario (IGIC), correspondiente al 7% que grava este tipo de servicios técnicos para profesionales en la Comunidad Autónoma de Canarias.

Tabla 20. Cálculo del presupuesto total.

Descripción	Coste (€)
Recursos materiales	417,72
Trabajo tarifado por tiempo empleado	13.478,40
Costes de material fungible	0
Costes asociados a la redacción	972,72
Derechos de visado del COITT	89,21
Gastos de tramitación y envío	6,00
Suma	14.964,05
IGIC (7%)	1047,48
TOTAL	**16.011,53**

Por tanto, el presupuesto final de este TFG asciende a los dieciséis mil once euros con cincuenta y tres céntimos (16.011,53 €), impuestos incluidos.

Las Palmas de Gran Canaria, a 26 de junio de 2025

Fdo.: Gabriel Ernesto Lares Aspera

ANEXOS

A1 Guía de instalación de *Open vSwitch* (OvS) en *Ubuntu 24.04.2 LTS*

A1.1 Introducción

Open vSwitch (OvS) es un conmutador virtual multicapa de código abierto que proporciona funcionalidades de *switching* y *routing* para entornos virtualizados. Esta guía detalla el proceso de instalación y configuración básica en *Ubuntu* 24.04.2 LTS.

A1.2. Instalación

La instalación de *Open vSwitch* se realiza mediante los paquetes provistos en los repositorios oficiales de *Ubuntu*. *Debian* y *Ubuntu* disponen de los paquetes binarios *openvswitch-switch* y *openvswitch-common*, que contienen los componentes principales del conmutador virtual OvS [72].

A1.3. Actualización del sistema e instalación

En una terminal con privilegios de superusuario, se debe actualizar el índice de paquetes e instalar *openvswitch-switch [73]*:

sudo apt update

sudo apt install -y openvswitch-switch

Este comando instalará:

- El demonio de OvS (*ovs-vswitchd*)
- La utilidad de configuración (*ovs-vsctl*)
- Las dependencias necesarias (como *openvswitch-common* y el módulo de *datapath* del *kernel*)

A1.4 Verificación del servicio

Tras la instalación, el servicio *openvswitch-switch* suele iniciarse automáticamente. Se recomienda comprobar su estado [74]:

sudo systemctl status openvswitch-switch

Si el servicio no está activo, iniciarlo manualmente:

sudo systemctl start openvswitch-switch

Para habilitar el inicio automático en el arranque del sistema:

sudo systemctl enable openvswitch-switch

A1.5. Configuración básica

Creación de un puente virtual (*bridge*)

Para crear un nuevo puente virtual se usa el comando *ovs-vsctl add-br*. Por ejemplo, para crear un bridge llamado *ovsbr0*:

sudo ovs-vsctl add-br ovsbr0

Adición de interfaces al puente

Para añadir una interfaz física o virtual al puente se utiliza *ovs-vsctl add-port*. Por ejemplo, para vincular la interfaz de red *eth0* al puente *ovsbr0*:

sudo ovs-vsctl add-port ovsbr0 eth0

Nota importante: Al añadir una interfaz física al bridge, la interfaz perderá su configuración IP. Es recomendable configurar la IP en el bridge en lugar de en la interfaz física.

Configuración de IP en el *bridge*

Para asignar una dirección IP al *bridge*:

sudo ip addr add 192.168.1.100/24 dev ovsbr0

sudo ip link set ovsbr0 up

A1.6. Comandos de administración y monitoreo

Inspección de la configuración

Para visualizar la configuración completa del *switch [73]*:

sudo ovs-vsctl show

Para listar únicamente los bridges configurados:

sudo ovs-vsctl list-br

Para enumerar los puertos de un bridge específico:

sudo ovs-vsctl list-ports ovsbr0

Información de flujos *OpenFlow*

Para mostrar información de bajo nivel (flujos *OpenFlow*) del *bridge*:

sudo ovs-ofctl show ovsbr0

Para ver las tablas de flujos:

sudo ovs-ofctl dump-flows ovsbr0

Eliminación de elementos

Para eliminar un puerto específico del bridge:

sudo ovs-vsctl del-port ovsbr0 eth0

Para eliminar un puente completo y sus puertos asociados:

sudo ovs-vsctl del-br ovsbr0

Información del sistema

Para verificar la versión instalada de OvS:

sudo ovs-vsctl --version

Para mostrar estadísticas de los puertos:

sudo ovs-vsctl list interface

Ejemplo de configuración completa

A continuación, se presenta un ejemplo práctico de configuración [74]:

 # Crear el bridge

sudo ovs-vsctl add-br ovsbr0

 # Añadir interfaces

sudo ovs-vsctl add-port ovsbr0 eth0

sudo ovs-vsctl add-port ovsbr0 eth1

 # Configurar IP en el bridge

sudo ip addr add 192.168.1.100/24 dev ovsbr0

sudo ip link set ovsbr0 up

 # Verificar configuración

sudo ovs-vsctl show

sudo ovs-ofctl show ovsbr0

A1.7. Solución de problemas comunes

El servicio no inicia

Verificar logs del sistema:

sudo journalctl -u openvswitch-switch -f

Bridge sin conectividad

Verificar que el *bridge* esté activo:

ip link show ovsbr0

Verificar configuración de red:

ip addr show ovsbr0

A2 Guía de instalación de servicios virtualizados en VirtualBox

A2.1. Introducción

En este anexo se describe paso a paso el proceso de instalación de cuatro servicios virtualizados sobre máquinas independientes de *VirtualBox*, con anfitrión *Ubuntu*. Cada servicio se instala en una máquina virtual distinta creada en *VirtualBox*. No se detallan configuraciones de red avanzadas ni parámetros internos de cada aplicación; el enfoque está en la descarga de la imagen o paquete oficial, la creación básica de la VM y el arranque inicial del servicio.

Se indican las fuentes oficiales de las imágenes ISO o repositorios utilizados, así como los parámetros mínimos de la VM (tipo de sistema, memoria, disco) necesarios para iniciar la instalación correctamente.

Servicios a instalar

Los cuatro servicios virtualizados que se abordan en esta guía son:

- ***VyOS**:* Sistema operativo *router* para servicio de datos.
- ***VLC Media Player**:* Servidor de *streaming* para servicio IPTV.
- ***ZoneMinder**:* Sistema de videovigilancia para servicio CCTV.
- ***FreePBX/Asterisk**:* Plataforma PBX para servicio VoIP.

Requisitos previos

- *VirtualBox* instalado en el sistema anfitrión *Ubuntu.*
- Conexión a Internet para descargar las imágenes ISO.
- Espacio suficiente en disco (mínimo 60 GB libres para todas las VMs).
- Al menos 8 GB de RAM en el sistema anfitrión.

A2.2. Instalación del *router* virtual (VyOS)

Descarga de la imagen ISO

Para el servicio de datos se emplea VyOS, un sistema operativo *router* de código abierto. Primero se descarga la imagen ISO oficial desde el sitio de descargas de VyOS [75]:

- ***URL oficial**:* https://downloads.vyos.io/rolling/current/amd64/vyos-rolling-latest.iso
- ***Alternativa**:* Portal de soporte de VyOS en docs.vyos.io.

Creación de la máquina virtual

Se crea una nueva máquina virtual en *VirtualBox* con las siguientes especificaciones:

Tabla 21. Especificaciones mínimas de la VM correspondiente al router VyOS. Elaboración propia.

Parámetro	Valor
Tipo de sistema	*Linux*
Versión	*Debian* (64-bit)
Memoria RAM	512 MB (mínimo)
Disco duro	10 GB (VDI dinámico)

***Justificación**:* VyOS se basa en *Debian*, por lo que se selecciona esta versión para una mejor compatibilidad.

Proceso de instalación

- ***Configuración inicial**:* En la configuración de la VM se agrega la ISO de VyOS como unidad óptica virtual.

- ***Arranque**:* Al iniciar la máquina, aparecerá el instalador en modo *"live"*.

- ***Instalación permanente**:* En la consola de VyOS se ejecuta el comando:

  install image

- ***Confirmación**:* El instalador confirmará la operación y realizará el copiado de archivos.

- ***Finalización**:* Tras completar la instalación, se retira la ISO y se reinicia la máquina.

Al primer arranque del sistema, VyOS mostrará la consola CLI, desde donde ya se puede proceder a la configuración de red básica del *router* [75].

A2.3. Instalación del VLC

Preparación del sistema base

El servicio IPTV se basa en VLC *Media Player* configurado como servidor de *streaming*. Se utiliza *Ubuntu Desktop* como sistema operativo base [77].

Descargar de *Ubuntu Desktop*

- ***Fuente**:* Web oficial de *Ubuntu* (www.ubuntu.com) [76].

- ***Versión recomendada**:* Ubuntu 24.04 LTS.

- ***Archivo**:* ubuntu-24.04.2-desktop-amd64.iso.

Configuración de la VM

Tabla 22. Especificaciones mínimas de la VM correspondiente al VLC. Elaboración propia.

Parámetro	Valor
Tipo de sistema	*Linux*
Versión	*Ubuntu* (64-bit)
Memoria RAM	1 GB (mínimo)
Disco duro	10 GB (VDI dinámico)

Instalación del sistema operativo

1. Arrancar la VM con la ISO de *Ubuntu [76]*
2. Instalar el sistema operativo utilizando el instalador por defecto
3. Seleccionar idioma y configurar particiones mínimas
4. Completar la instalación y reiniciar

Instalación y configuración de VLC

Una vez instalado Ubuntu y reiniciada la VM, actualizaremos el sistema mediante el siguiente comando [77].

sudo apt-get update

Posteriormente seguiremos con la instalación del *software* VLC a través de la siguiente instrucción:

sudo apt-get install vlc

Para iniciar VLC como servidor de *streaming*, ejecutaremos la siguiente línea:

vlc video.mp4 --sout="#std{access=http,mux=ts,dst=:8090/video}"

Parámetros de *streaming*

Según la documentación oficial de VLC, los parámetros principales son [77]:

- ***access=http:*** Protocolo de transmisión
- ***mux=ts***: Formato de multiplexación
- ***dst=:8090/video:*** Puerto y ruta de acceso

A2.4. Instalación del *ZoneMinder*

Para el servicio de videovigilancia se utiliza *ZoneMinder*, un *software* de gestión de cámaras IP [78].

Configuración de la VM

Tabla 23. Especificaciones mínimas de la VM correspondiente al ZoneMinder. Elaboración propia.

Parámetro	Valor
Tipo de sistema	Linux
Versión	Ubuntu (64-bit)
Memoria RAM	2 GB (mínimo)
Disco duro	20 GB (VDI dinámico)

***Nota**:* Se requiere más espacio de disco para almacenar grabaciones de vídeo.

Instalación del sistema base

Se utiliza la misma ISO de *Ubuntu Desktop* descargada anteriormente (www.ubuntu.com) [76].

Instalación del software *ZoneMinder*

Primero se ha de actualizar del sistema [78]:

sudo apt-get update

Después de tener el sistema actualizado pasaremos con la instalación del software mediante la siguiente instrucción:

sudo apt-get install zoneminder

El comando anterior instala *ZoneMinder* junto con *Apache*, MySQL y dependencias necesarias para el correcto funcionamiento del *software*.

Después de tener instalado en software, pasaremos a habilitarlo:

sudo systemctl enable zoneminder

sudo systemctl start zoneminder

Acceso a la interfaz web

Una vez iniciado el servicio, se puede acceder a la interfaz web desde el navegador:

http://<IP_VM>/zm

- Donde <*IP_VM*> es la dirección IP de la máquina virtual.

A2.5. Instalación de FreePBX de *Asterisk*

Acceso a la interfaz web

El servicio VoIP se implementa con FreePBX, una plataforma de PBX basada en *Asterisk* [79].

- ***Fuente**:* Sitio oficial de FreePBX (https://www.freepbx.org/downloads/).
- ***Archivo**:* SNG7-FPBX-64bit-xxxx-*.iso (versión más reciente).
- ***Distribución**: Sangoma* con *Asterisk* incluido.

Configuración de la VM

Tabla 24. Especificaciones mínimas de la VM correspondiente al Asterisk FreePBX. Elaboración propia.

Parámetro	Valor
Tipo de sistema	Linux
Versión	Other Linux (64-bit)
Memoria RAM	1 GB (mínimo)
Disco duro	10-20 GB (VDI dinámico)

Proceso de instalación

1. ***Arranque**:* Se inicia la máquina insertando la ISO de FreePBX como unidad óptica virtual.

2. ***Instalador**:* Al arrancar, aparecerá el instalador de FreePBX basado en texto.
3. ***Instalación**:* El instalador realiza una instalación semiautomática:
    - Se confirman las opciones predeterminadas.
    - Se establece una contraseña de *root* (importante anotarla).
    - El sistema procede a copiar archivos y configurar *Asterisk* y FreePBX.
4. ***Finalización**:* Cuando la instalación finaliza, se reinicia la máquina retirando la ISO virtual.

Primer arranque

En el arranque posterior, la VM presentará un *prompt* de *login* de FreePBX:

- ***Usuario**: root*.
- ***Contraseña**:* La establecida durante la instalación.

Se accede a un menú de comandos de FreePBX donde se puede configurar la interfaz de red y otros parámetros.

A3 Instalación y configuración de clientes *OpenVPN*

A3.1. Introducción

El presente anexo tiene como objetivo proporcionar una guía detallada para la instalación y configuración del cliente *OpenVPN* en los sistemas operativos *Windows 10/11* y *Ubuntu Desktop 24.04.* Este procedimiento constituye un elemento fundamental para el establecimiento de conexiones VPN seguras entre equipos cliente y el servidor *OpenVPN* configurado en el *router TP-Link*.

Es importante señalar que este anexo presupone que el archivo de configuración VPN con extensión *.ovpn* ha sido previamente generado en el servidor *OpenVPN* del *router TP-Link*, y que dicho archivo estará disponible para su importación posterior en cada cliente una vez completada la instalación del *software* correspondiente [80][83].

A3.2. Instalación y configuración del cliente *OpenVPN* en *Windows 10/11*

Requisitos previos

Para la implementación del cliente *OpenVPN* en sistemas *Windows*, se recomienda la utilización del cliente oficial *OpenVPN Connect*, desarrollado por *OpenVPN Technologies, Inc*. Este *software* proporciona una interfaz intuitiva y funcionalidades avanzadas que facilitan la gestión de conexiones VPN [81].

Proceso de descarga e instalación

El proceso de instalación del cliente *OpenVPN* en *Windows* comprende las siguientes etapas [81]:

Etapa 1: Obtención del software.

Se debe acceder a la página oficial de descargas de *OpenVPN* (https://openvpn.net/community-downloads/) o al sitio de soporte de *TP-Link* (https://www.tp-link.com) para obtener la versión más reciente compatible con el sistema operativo *Windows*. Es fundamental seleccionar la versión apropiada según la arquitectura del sistema (32 o 64 bits).

Etapa 2: Instalación del cliente.

Una vez descargado el archivo instalador, se debe ejecutar con privilegios de administrador para garantizar la correcta instalación de los componentes del sistema. El asistente de instalación guiará al usuario a través del proceso, incluyendo la aceptación de los términos de licencia y la selección del directorio de instalación.

Configuración del perfil VPN

Tras completar la instalación, se procede a la configuración del perfil VPN mediante la importación del archivo *.ovpn* generado previamente en el servidor. Este proceso se realiza a través de la opción "Importar perfil desde archivo" disponible en la interfaz principal de *OpenVPN Connect.*

Como alternativa, los usuarios que prefieran utilizar la interfaz gráfica clásica de *OpenVPN* (*OpenVPN* GUI) pueden copiar manualmente el archivo *.ovpn* en la carpeta *"config"* ubicada dentro del directorio de instalación de *OpenVPN*.

Establecimiento y verificación de la conexión

El establecimiento de la conexión VPN se realiza seleccionando el perfil importado y activando la opción "Conectar" en la interfaz del cliente. Durante este proceso, el sistema solicitará los permisos necesarios para establecer la conexión de red.

La verificación del estado de la conexión se puede realizar mediante la observación del icono de *OpenVPN* en la bandeja del sistema, el cual debe mostrar un estado activo acompañado de un mensaje confirmando la conexión exitosa. Esta confirmación indica que el túnel VPN hacia la red local del *router* ha sido establecido correctamente.

A3.3. Instalación y configuración del cliente *OpenVPN* en *Ubuntu Desktop 24.04*

Metodología de implementación

En sistemas *Ubuntu Desktop*, la implementación del cliente *OpenVPN* puede realizarse mediante dos enfoques principales: utilizando la interfaz de línea de comandos o a través del gestor gráfico de red. En el presente documento se detalla el procedimiento mediante línea de comandos debido a su mayor flexibilidad y control sobre el proceso de configuración [82].

Instalación del cliente *OpenVPN*

Previo a la instalación, es necesario actualizar la base de datos de paquetes del sistema ejecutando el siguiente comando en una terminal:

sudo apt update

La instalación del cliente *OpenVPN* se realiza mediante el gestor de paquetes APT:

sudo apt install openvpn

Este comando instalará el paquete *openvpn*, que incluye tanto el cliente VPN de línea de comandos como las herramientas auxiliares necesarias para su funcionamiento.

Configuración e importación del perfil

El archivo *.ovpn* generado en el *router* debe ser transferido al sistema *Ubuntu*. Se recomienda ubicarlo en el directorio personal del usuario o, para configuraciones del sistema, en el directorio */etc/openvpn/.*

La conexión VPN se establece ejecutando el siguiente comando en la terminal:

sudo openvpn --config /ruta/al/archivo.ovpn

Como alternativa al método de línea de comandos, es posible utilizar la herramienta gráfica de configuración de red instalando previamente el paquete *network-manager-openvpn:*

sudo apt install network-manager-openvpn

Verificación y diagnóstico de la conexión

Una vez establecida la conexión, se debe verificar la creación de la interfaz de túnel tun0 mediante el comando [82]:

ip addr show tun0

Este comando debe mostrar la interfaz listada con una dirección IP asignada, confirmando la correcta configuración del túnel VPN.

Por otro lado, el estado del servicio *OpenVPN* puede verificarse utilizando *systemctl*:

sudo systemctl status openvpn@client

Un estado *"active (running)"* indica que la VPN está conectada y funcionando correctamente.

Como verificación adicional, se recomienda realizar una prueba de conectividad enviando paquetes ICMP a una dirección de la red remota privada:

ping [dirección_IP_del_router]

La respuesta exitosa a este comando confirma la conectividad a través del túnel VPN.

Consideraciones de seguridad y mantenimiento

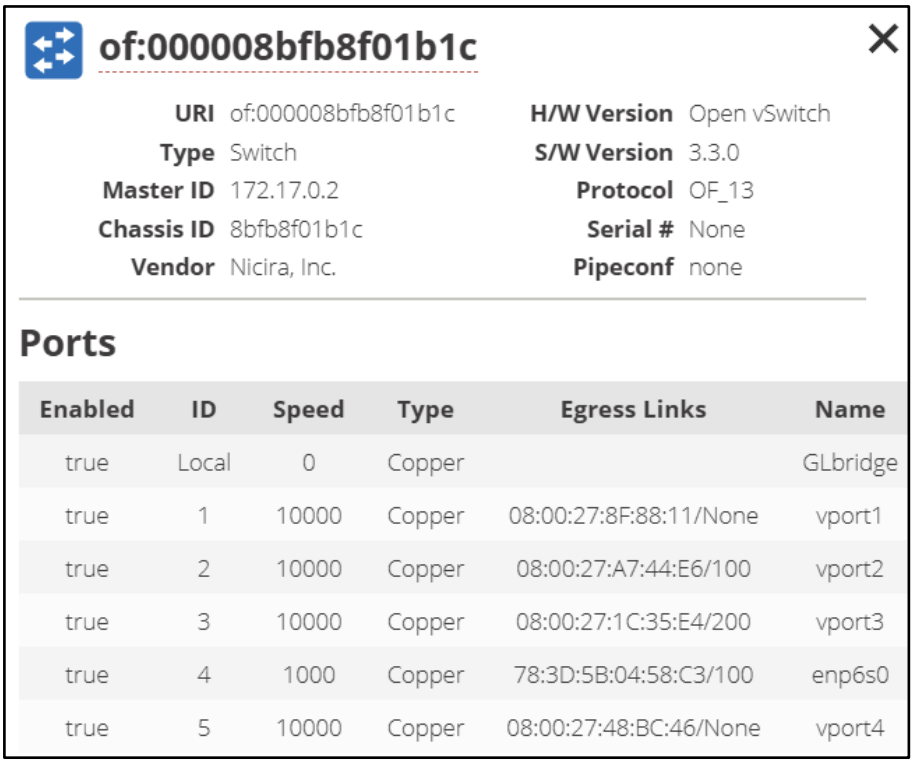
Es importante mantener actualizado el *software OpenVPN* en ambos sistemas operativos para garantizar la seguridad y compatibilidad con las últimas versiones del protocolo. Asimismo, se recomienda realizar verificaciones periódicas del estado de la conexión y mantener copias de seguridad de los archivos de configuración [84].

A4 Pruebas prácticas de gestión SDN en red GPON virtualizada (*ONOS/Swagger*)

A4.1. Introducción

Este anexo describe las pruebas prácticas de gestión SDN (*Software-Defined Networking*) realizadas en la red GPON virtualizada implementada, utilizando el controlador *ONOS* (*Open Network Operating System)* a través de su interfaz de programación de aplicaciones (API) *REST* y la interfaz gráfica *Swagger UI*.

La implementación emplea un *switch* virtual *Open vSwitch* (OvS) con identificador of:000008bfb8f01b1c, que incluye cuatro puertos virtuales (*vport1-vport4*) y un puerto físico (*enp6s0*), tal como se muestra en la Figura X. El controlador *ONOS* permite la gestión programática de reglas de reenvío (*flows*) mediante peticiones HTTP con cuerpos JSON, facilitando la configuración dinámica de la red [84].



of:000008bfb8f01b1c

URI	of:000008bfb8f01b1c	**H/W Version**	Open vSwitch
Type	Switch	**S/W Version**	3.3.0
Master ID	172.17.0.2	**Protocol**	OF_13
Chassis ID	8bfb8f01b1c	**Serial #**	None
Vendor	Nicira, Inc.	**Pipeconf**	none

Ports

Enabled	ID	Speed	Type	Egress Links	Name
true	Local	0	Copper		GLbridge
true	1	10000	Copper	08:00:27:8F:88:11/None	vport1
true	2	10000	Copper	08:00:27:A7:44:E6/100	vport2
true	3	10000	Copper	08:00:27:1C:35:E4/200	vport3
true	4	1000	Copper	78:3D:5B:04:58:C3/100	enp6s0
true	5	10000	Copper	08:00:27:48:BC:46/None	vport4

Figura 107. Resumen de parámetros de red del OvS. Elaboración propia.

Objetivos de las pruebas

Las pruebas implementadas tienen como objetivo validar las siguientes funcionalidades de gestión SDN:

1. ***Aislamiento de abonado**:* Bloqueo selectivo de tráfico desde puertos específicos.
2. ***Supervisión de tráfico**:* Monitoreo en tiempo real de estadísticas de puertos y flujos.
3. ***Redireccionamiento de tráfico**:* Modificación dinámica de rutas entre puertos.

Metodología

Para cada prueba se presenta:

- Marco teórico del concepto implementado.
- Objetivo específico de la prueba.
- Configuración JSON utilizada.
- Resultados obtenidos.
- Análisis de aplicaciones prácticas.

La interfaz *Swagger* de *ONOS*, accesible en *http://localhost:8181/onos/v1/docs*, permite la ejecución directa de peticiones HTTP y la generación automática de documentación de la API *REST*.

A4.2. Aislamiento de abonado

Fundamento teórico

El aislamiento de abonado es una técnica de control de acceso que permite suspender temporalmente la conectividad de un usuario específico sin afectar al resto de la red. Esta funcionalidad es equivalente al concepto de *"client isolation"* en redes LAN/WiFi [84].

Objetivo

Implementar un mecanismo de bloqueo que impida la propagación del tráfico originado desde el puerto virtual vport2, simulando la suspensión temporal de un abonado por motivos de mantenimiento o incumplimiento de políticas.

Configuración implementada

La regla de aislamiento se configura mediante una petición HTTP *POST* al *endpoint /onos/v1/flows/of:000008bfb8f01b1c* con el siguiente cuerpo JSON:

```
{
  "flows": [
    {
      "priority": 40000,
      "timeout": 0,
      "isPermanent": true,
      "deviceId": "of:000008bfb8f01b1c",
      "treatment": {
        "instructions": [
          { "type": "NOACTION" }
        ]
      },
      "selector": {
        "criteria": [
          { "type": "IN_PORT", "port": "2" }
        ]
      }
    }
  ]
}
```

Parámetros de configuración

- ***priority: 40000:*** Prioridad alta para garantizar precedencia sobre otras reglas.
- ***isPermanent: true:*** Regla persistente hasta su eliminación manual.
- ***IN_PORT: 2:*** Criterio de selección para tráfico entrante por puerto 2.

- ***NOACTION:*** Instrucción de descarte de paquetes.

Resultados obtenidos

La implementación de la regla de aislamiento produjo los siguientes resultados:

1. ***Confirmación de instalación***: El controlador *ONOS* confirmó la inserción exitosa del flujo.
2. ***Bloqueo efectivo****:* Todo el tráfico originado desde vport2 fue descartado.
3. ***Pérdida de conectividad****:* Los dispositivos conectados al puerto 2 perdieron comunicación con el resto de la red.
4. ***Contadores de estadísticas****:* Los contadores de paquetes de salida permanecieron en cero para el puerto origen.

Consecuentemente podemos observar la correcta inserción del flujo como podemos observar en la figura X, provocando así el bloqueo de los paquetes para en puerto en cuestión.

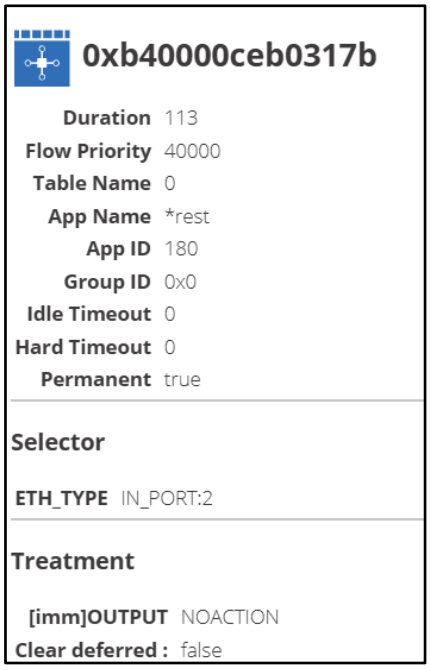


Figura 108. Resumen de regla de flujo-Bloqueo de puerto-implementada por API REST vía Swagger. Elaboración propia.

Aplicaciones prácticas

Esta funcionalidad presenta múltiples aplicaciones en entornos comerciales:

- ***Control de acceso**:* Suspensión temporal de usuarios problemáticos o maliciosos.
- ***Seguridad de red**:* Implementación de controles contra spam o ataques DDoS.
- ***Gestión de cuotas**:* Aplicación de políticas de uso y facturación.
- ***Mantenimiento**:* Aislamiento temporal durante tareas de mantenimiento.

A4.3. Supervisión y monitoreo de tráfico

Fundamento teórico

La supervisión de tráfico es fundamental para las operaciones de red (OAM - *Operations, Administration, and Maintenance*), permitiendo la recolección de métricas en tiempo real para evaluación de rendimiento y detección de anomalías [84].

Objetivo

Implementar un sistema de monitoreo que permita obtener estadísticas detalladas de tráfico tanto a nivel de puertos físicos como de flujos instalados, facilitando el diagnóstico y la optimización de la red.

Configuración implementada

La consulta de estadísticas de flujos se realiza mediante petición HTTP *GET* que permite ejecutarse a través de *Swagger* como podemos ver en la siguiente figura:

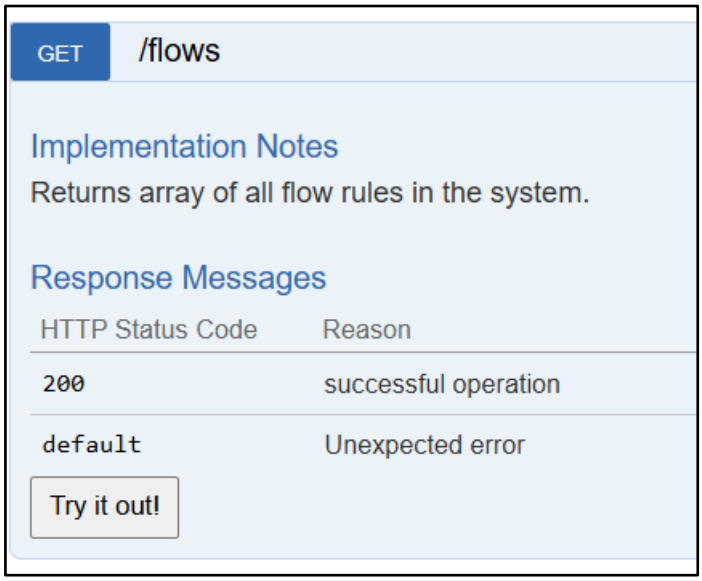


Figura 109.Consulta de estadísticas de los flujos vía Swagger. Elaboración propia.

Y como resultado de la petición anterior tendremos la siguiente respuesta:


```
{
  "flows": [
    {
      "groupId": 0,
      "state": "ADDED",
      "life": 2126,
      "liveType": "UNKNOWN",
      "lastSeen": 1750339456364,
      "packets": 0,
      "bytes": 0,
      "id": "281476057627014",
      "appId": "org.onosproject.core",
      "priority": 40000,
      "timeout": 0,
      "isPermanent": true,
      "deviceId": "of:000008bfb8f01b1c",
      "tableId": 0,
      "tableName": "0",
      "treatment": {
        "instructions": [
          {
            "type": "OUTPUT",
            "port": "CONTROLLER"
          }
        ],
        "clearDeferred": true,
        "deferred": []
      },
      "selector": {
        "criteria": [
          {
            "type": "ETH_TYPE",
            "ethType": "0x88cc"
          }
        ]
      }
    },
    {
      "groupId": 0,
      "state": "ADDED",
      "life": 2126,
      "liveType": "UNKNOWN",
```

```
      "lastSeen": 1750339456364,
      "packets": 0,
      "bytes": 0,
      "id": "281476610891318",
      "appId": "org.onosproject.core",
      "priority": 5,
      "timeout": 0,
      "isPermanent": true,
      "deviceId": "of:000008bfb8f01b1c",
      "tableId": 0,
      "tableName": "0",
      "treatment": {
        "instructions": [
          {
            "type": "OUTPUT",
            "port": "CONTROLLER"
          }
        ],
        "clearDeferred": true,
        "deferred": []
      },
      "selector": {
        "criteria": [
          {
            "type": "ETH_TYPE",
            "ethType": "0x800"
          },
          {
            "type": "IPV4_DST",
            "ip": "224.0.0.0/4"
          }
        ]
      }
    },
    {
      "groupId": 0,
      "state": "ADDED",
      "life": 2126,
      "liveType": "UNKNOWN",
      "lastSeen": 1750339456364,
      "packets": 806,
      "bytes": 51400,
      "id": "281477261216972",
      "appId": "org.onosproject.core",
      "priority": 40000,
```

```
      "timeout": 0,
      "isPermanent": true,
      "deviceId": "of:000008bfb8f01b1c",
      "tableId": 0,
      "tableName": "0",
      "treatment": {
        "instructions": [
          {
            "type": "OUTPUT",
            "port": "CONTROLLER"
          }
        ],
        "clearDeferred": true,
        "deferred": []
      },
      "selector": {
        "criteria": [
          {
            "type": "ETH_TYPE",
            "ethType": "0x806"
          }
        ]
      }
    },
    {
      "groupId": 0,
      "state": "ADDED",
      "life": 2126,
      "liveType": "UNKNOWN",
      "lastSeen": 1750339456364,
      "packets": 0,
      "bytes": 0,
      "id": "281478661736285",
      "appId": "org.onosproject.core",
      "priority": 40000,
      "timeout": 0,
      "isPermanent": true,
      "deviceId": "of:000008bfb8f01b1c",
      "tableId": 0,
      "tableName": "0",
      "treatment": {
        "instructions": [
          {
            "type": "OUTPUT",
            "port": "CONTROLLER"
```

```
        }
      ],
      "clearDeferred": true,
      "deferred": []
    },
    "selector": {
      "criteria": [
        {
          "type": "ETH_TYPE",
          "ethType": "0x8942"
        }
      ]
    }
  },
  {
    "groupId": 0,
    "state": "ADDED",
    "life": 2126,
    "liveType": "UNKNOWN",
    "lastSeen": 1750339456364,
    "packets": 234,
    "bytes": 65025,
    "id": "281478762648302",
    "appId": "org.onosproject.core",
    "priority": 5,
    "timeout": 0,
    "isPermanent": true,
    "deviceId": "of:000008bfb8f01b1c",
    "tableId": 0,
    "tableName": "0",
    "treatment": {
      "instructions": [
        {
          "type": "OUTPUT",
          "port": "CONTROLLER"
        }
      ],
      "clearDeferred": true,
      "deferred": []
    },
    "selector": {
      "criteria": [
        {
          "type": "ETH_TYPE",
          "ethType": "0x800"
```

```
        }
      ]
    }
  },
  {
    "groupId": 0,
    "state": "ADDED",
    "life": 1298,
    "liveType": "UNKNOWN",
    "lastSeen": 1750339456364,
    "packets": 247,
    "bytes": 29816,
    "id": "50665499275571579",
    "appId": "org.onosproject.rest",
    "priority": 40000,
    "timeout": 0,
    "isPermanent": true,
    "deviceId": "of:000008bfb8f01b1c",
    "tableId": 0,
    "tableName": "0",
    "treatment": {
      "instructions": [
        {
          "type": "NOACTION"
        }
      ],
      "deferred": []
    },
    "selector": {
      "criteria": [
        {
          "type": "IN_PORT",
          "port": 2
        }
      ]
    }
  }
]
}
```


Análisis de la respuesta:

1. ***Flows del sistema (org.onosproject.core)**: ONOS* instala automáticamente varios *flows* para:
    - ***LLDP (0x88cc)**:* Protocolo de descubrimiento de enlaces.
    - ***ARP (0x806)**:* Resolución de direcciones con 806 paquetes procesados.
    - ***IPv4 (0x800)**:* Tráfico IP general con 234 paquetes procesados.
    - ***IPv4 Multicast (224.0.0.0/4)**:* Tráfico *multicast* sin actividad.
2. ***Flow personalizado (org.onosproject.rest)**:* Nuestro flow de aislamiento con:
    - ***ID**:* 50665499275571579.
    - ***Criterio**:* IN_PORT puerto 2.
    - ***Acción**:* NOACTION (descarte).
    - ***Estadísticas**:* 247 paquetes bloqueados, 29.816 bytes descartados.
    - ***Tiempo de vida**:* 1.298 segundos desde su instalación.

Resultados obtenidos

Las pruebas de monitoreo con datos reales del sistema demostraron:

1. ***Recolección exitosa de métricas**:* El *endpoint GET* devolvió información completa de 6 *flows* activos en el *switch.*
2. ***Flows automáticos del sistema**: ONOS* instaló automáticamente 5 *flows* de control (LLDP, ARP, IPv4, *multicast*).
3. ***Tracking del flow personalizado**:* Nuestro *flow* de aislamiento aparece correctamente identificado con appId: *"org.onosproject.rest"*
4. ***Estadísticas en tiempo real**:* Los contadores muestran actividad real:

- ***Flow ARP**:* 806 paquetes, 51.400 bytes.
- ***Flow IPv4***: 234 paquetes, 65.025 bytes.
- ***Flow de aislamiento**:* 247 paquetes bloqueados, 29.816 bytes descartados.

Métricas clave observadas:

- ***Tiempo de vida (life)**:* Permite identificar *flows* recientes vs. Establecidos.
- ***Estado (state)**:* Todos los *flows* muestran estado *"ADDED"* (activos).
- ***Último acceso (lastSeen)**: Timestamp* de la última actualización de estadísticas.
- ***Contadores de tráfico**: packets/bytes* proporcionan métricas precisas de uso.

Validación funcional del aislamiento: El *flow* personalizado con *IN_PORT: 2* y *NOACTION* muestra 247 paquetes interceptados y descartados, confirmando que el mecanismo de aislamiento está funcionando correctamente y bloqueando el tráfico del puerto especificado.

Aplicaciones prácticas

El sistema de monitoreo implementado permite:

- ***Medición de ancho de banda**:* Análisis del uso de recursos de red.
- ***Detección de anomalías**:* Identificación de picos de tráfico inesperados.
- ***Generación de reportes**:* Datos para facturación y garantía de servicio.
- ***Integración con sistemas externos**:* Compatibilidad con plataformas de gestión de red.

A4.4. Funcionalidades adicionales de la API REST de ONOS (*Swagger*)

Además de las funciones exploradas en secciones anteriores (aislamiento y supervisión), existen numerosas capacidades adicionales. Por ejemplo, la API *REST* de *ONOS* permite gestionar el inventario de infraestructura: es posible listar todos los dispositivos de red, consultar detalles de cada uno y crear, actualizar o eliminar dispositivos en el inventario [84]. De forma similar, los *endpoints* correspondientes permiten administrar los enlaces físicos de la red, obteniendo listados de enlaces existentes y modificando atributos (como tipo o estado) de enlaces específicos [84].

Asimismo, la API REST de ONOS facilita el descubrimiento y la gestión de hosts finales. A través de la API se pueden listar todos los hosts conectados a la red, obtener información detallada de cada host (por ejemplo, por su dirección MAC y VLAN) e incluso crear o actualizar su configuración o eliminarlo del inventario [84]. En cuanto a la topología, la API ofrece servicios para obtener una vista general de la red y de sus *clústers*, mostrando los dispositivos y enlaces asociados a cada *clúster* topológico [84]. Además, es posible consultar rutas prestablecidas entre dos elementos de red, utilizando los *endpoints* de cálculo de caminos disponibles [84].

Otro componente clave es la gestión de *intents*. La API REST de ONOS proporciona *endpoints* para crear nuevos *intents*, listar todos los *intents* existentes y eliminar *intents* específicos [84]. Estos *intents* representan políticas de conectividad de alto nivel que *ONOS* traduce a flujos subyacentes. De forma complementaria, la interfaz permite configurar *métricas* (*meters*) en los dispositivos, lo que posibilita imponer reglas de calidad de servicio (por ejemplo, limitación de ancho de banda). Para ello existen *endpoints* para crear, listar y borrar entradas de *meter* en cada dispositivo [84].

Más allá de lo anterior, la documentación *Swagger* revela que la API *REST* de *ONOS* cubre prácticamente todos los aspectos del controlador SDN. Por ejemplo, se incluyen *endpoints* para gestionar reglas de flujo y entradas de grupo en los *switches*, así como para administrar aplicaciones de *ONOS* (instalación, activación o desactivación de aplicaciones) [84]. Asimismo, la API ofrece funcionalidades para consultar y modificar la configuración de cada componente interno del sistema, a través de los *endpoints* de configuración [84].

En conjunto, estas capacidades proporcionan una visión integral de las posibilidades de gestión de la red que ofrece *ONOS* a través de su interfaz *Swagger*.

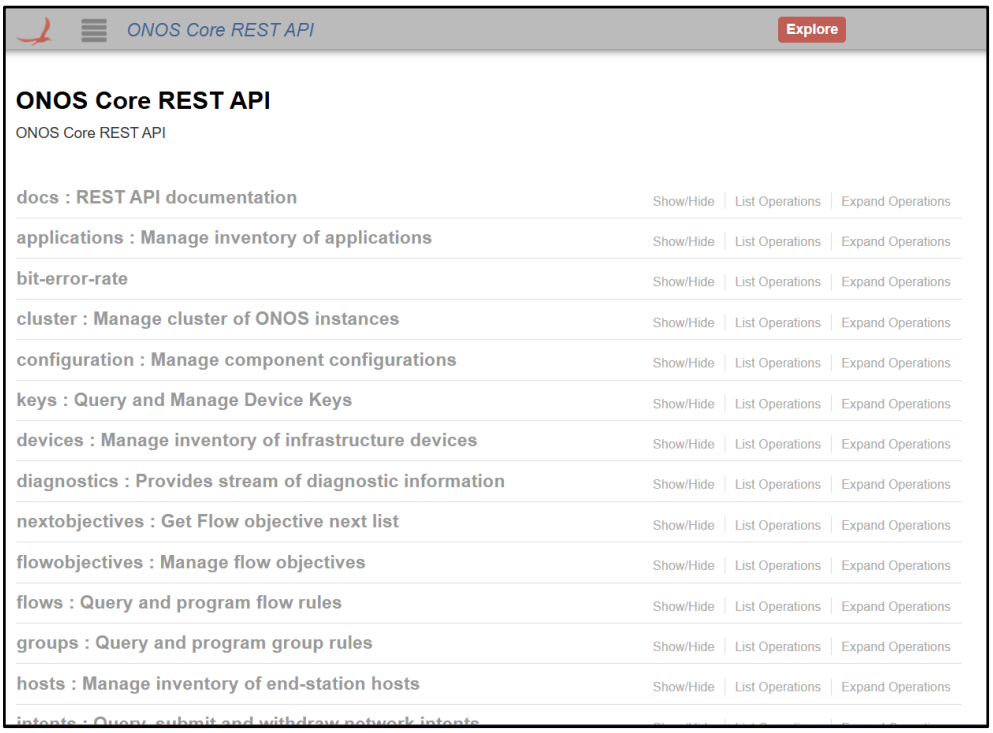


Figura 110. Interfaz de usuario del Swagger de ONOS. Elaboración propia.