



UNIVERSIDAD DE

VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE Máster

Máster EN INGENIERÍA DE TELECOMUNICACIÓN

Implementación de un módulo de control automatizado en Python de una maqueta de red GPON

Autor:

Don Ángel Gómez Aguado

Tutor:

Dña Noemí Merayo Álvarez

TÍTULO: Implementación de un módulo de control automatizado en Python de una maqueta de red GPON

AUTOR: Don Ángel Gómez Aguado

TUTOR: Doña Noemí Merayo Álvarez

DEPARTAMENTO: Teoría de la Señal y Comunicaciones e Ingeniería Telemática

TRIBUNAL

PRESIDENTE: Doña Patricia Fernández del Reguero

VOCAL: Don Ignacio de Miguel Jiménez

SECRETARIO: Don Ramón José Durán Barroso

PRESIDENTE SUPLENTE: Doña Lourdes Pelaz Montes

SECRETARIO SUPLENTE: Doña María Jesús Verdú Pérez

VOCAL SUPLENTE: Don Ramón de la Rosa Steinz

FECHA:

CALIFICACIÓN:

Resumen de TFM

En este trabajo se ha desarrollado una aplicación capaz de configurar servicios de Internet en la red GPON del laboratorio 2L007 de la Escuela Técnica Superior de Ingeniería de Telecomunicación de la Universidad de Valladolid. La configuración será de dos formas diferentes: mediante comandos en el CLI del OLT, y mediante el uso del estándar OpenFlow.

Palabras clave

OpenFlow, ONU, ONT, OLT, CLI, Python, SQL, GPON

Abstract

In this work we have developed an application capable of configuring Internet services in the GPON network of the 2L007 laboratory of the School of Telecommunications Engineering of the University of Valladolid. The configuration will be in two different ways: through commands in the OLT CLI and using the OpenFlow standard.

Keywords

OpenFlow, ONU, ONT, OLT, CLI, Python, SQL, GPON

ÍNDICE

1	Introducción	12
1.1	Motivación	12
1.2	Objetivos.....	14
1.2.1	Objetivo General.....	14
1.2.2	Objetivos Específicos	14
1.3	Fases y Métodos	14
1.3.1	Fase de Análisis	14
1.3.2	Fase de Configuración	15
1.3.3	Fase de Programación.....	16
1.3.4	Fase de Realización de los Informes	16
1.4	Estructura de la memoria del TFM.....	16
2	Metodología y herramientas de trabajo	17
2.1	Introducción.....	17
2.2	Montaje real y modos de gestión de la red GPON	18
2.2.1	Estructura y componentes de la red de acceso GPON.....	18
2.2.2	Modos de gestión de la red GPON	20
2.3	Lenguaje de programación Python	20
2.4	Protocolo OpenFlow	21
2.4.1	Estándar OpenFlow	21
2.4.2	Protocolo OpenFlow para la gestión de servicios en la red GPON	22
2.5	Metodología de trabajo	22
2.5.1	Análisis del modo de configuración por línea de comandos del OLT	23
2.5.2	Análisis del modo de configuración a través de OpenFlow	23
2.5.3	Especificación de requisitos y programación	24
2.6	Conclusiones.....	25

3	Descripción global del programa en Python y su funcionamiento.....	26
3.1	Introducción.....	26
3.2	Requisitos y funcionalidades del programa.....	26
3.3	Estructura lógica del programa.....	27
3.3.1	Lógica del programa.....	27
3.3.2	Estructura de la base de datos.....	38
3.3.3	Conclusiones.....	39
4	Configuración de servicios mediante CLI.....	40
4.1	Introducción.....	40
4.2	Estructura jerárquica de los comandos CLI.....	41
4.3	Configuración de servicios mediante CLI.....	43
4.3.1	Configuración de servicio de Internet.....	43
4.3.2	Configuración de servicio de video.....	49
4.3.3	Implementación en Python de la configuración de servicios mediante CLI.....	50
4.4	Eliminación de servicios mediante CLI.....	54
4.4.1	Comandos CLI para eliminar servicios.....	54
4.4.2	Implementación en Python de la eliminación de servicios.....	55
4.5	Conclusiones.....	57
5	Configuración de servicios mediante OpenFlow.....	58
5.1	Introducción.....	58
5.2	Integración de OpenFlow en la red GPON.....	59
5.2.1	Conceptos básicos de OpenFlow.....	59
5.2.2	Integración de OpenFlow a la red GPON del laboratorio.....	60
5.3	Flujos OpenFlow.....	62
5.3.1	Configuración de flujos OpenFlow.....	62
5.3.2	Configuración de los medidores OpenFlow.....	64

5.4	Implementación en Python de la configuración de servicios mediante OpenFlow	64
5.4.1	Métodos de configuración de flujos OpenFlow y medidores	64
5.4.2	Implementación en Python de la configuración de servicios	65
5.5	Conclusiones	69
6	Conclusiones y Líneas Futuras	71
6.1	Conclusiones	71
6.2	Líneas Futuras	72
7	Bibliografía	73

ÍNDICE DE FIGURAS

Figura 1: Esquema general de la red de acceso GPON desplegada en el laboratorio	19
Figura 2: Clase Servicio y su constructor	28
Figura 3: Menú principal de la aplicación	29
Figura 4: Menú de configuración de servicios	29
Figura 5: Creación de un servicio	29
Figura 6: Listado de servicios existentes en la base de datos	30
Figura 7: Mofidicación de servicios	30
Figura 8: Borrado de un servicio	31
Figura 9: Menú CLI	31
Figura 10: Asociación de un servicio con una ONU/ONT mediante CLI.....	32
Figura 11: Creación de objeto servicio	32
Figura 12: Llamada al método servicio_Internet_Video	33
Figura 13: Resumen del servicio configurado	34
Figura 14: Desasociación de un servicio y una ONU/ONT mediante CLI	34
Figura 15: Llamada al método borrar_configuracion.....	35
Figura 16: Menú de configuración OpenFlow.....	36
Figura 17: Flujo encargado de la subida.....	37
Figura 18: Ejemplo escenario GPON	40
Figura 19: Estructura jerárquica del CLI	43
Figura 20: Relación entre T-CONT, alloc-id y GEM Port	45
Figura 21: Declaración del método servicio_Internet_Video.....	51
Figura 22: Creación de listas de parámetros y fichero de log.....	51
Figura 23: Código de recuperación de datos de servicios anteriores.....	52
Figura 24: Introducción de los datos del nuevo servicio	52
Figura 25: Asignación de números a parámetros de configuración del OLT para cada servicio.....	53

Figura 26: Inicio de la introducción de comandos en el CLI del OLT.....	53
Figura 27: Ejemplo de un comando que solo se introduce cuando se quiere configurar un servicio de vídeo.....	54
Figura 28: Introducción del servicio configurado en la base de datos	54
Figura 29: Llamada al método de borrar_configuracion()	56
Figura 30: Conexión a la base de datos e introducción de algunos comandos.....	57
Figura 31: Borrado de instancias pertenecientes a servicio de vídeo	57
Figura 32: Red GPON sin OpenFlow	60
Figura 33: Integración de un escenario OpenFlow en la red GPON (SDN-GPON)	61
Figura 34: Campos de un flujo OpenFlow	62
Figura 35: Campos de configuración de un flujo OpenFlow en la interfaz web del controlador ODL.....	65
Figura 36: Declaración del método de configuración mediante OpenFlow	66
Figura 37: Selección del ID de OVS y MAC destino en función de MAC_ONU	66
Figura 38: Dirección MAC de destino para los flujos del COVS	67
Figura 39: Cabecera HTTP junto a URLs a las que se enviarán los datos	67
Figura 40: Flujo de upstream.....	68
Figura 41: Medidor para upstream	68
Figura 42: Inserción de flujos y medidores en el controlador mediante petición HTTP PUT.....	69
Figura 43: Inserción del servicio configurado en la base de datos	69
Figura 44: Código principal del borrado de un servicio OpenFlow	69

ÍNDICE DE TABLAS

Tabla 1: Tabla Services	38
Tabla 2: Tabla Onts	38
Tabla 3: Tabla ont_service.....	39

1

Introducción

1.1 Motivación

En este Trabajo de Fin de Máster se ha llevado a cabo el diseño e implementación de un programa en Python capaz configurar servicios de red y perfiles de abonado en una red real GPON (*Gigabit-capable Passive Optical Network*) de forma automatizada bien a través del interfaz de comandos CLI (*Command Line Interface*) del OLT (*Optical Line Termination*), o haciendo uso del estándar OpenFlow. Esta red de acceso está situada en el laboratorio de Comunicaciones Ópticas (2L007) de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Valladolid.

La red GPON es una red de acceso óptica pasiva con capacidad para el estándar Gigabit. Gracias a su simplicidad, debida a la no necesidad de elementos activos entre el operador y las dependencias del abonado (20-25 km), las redes GPON son las redes de acceso más desplegadas a nivel mundial en la actualidad. Estas redes se basan en una topología en árbol, donde hay un elemento central que da servicio a varios usuarios utilizando el protocolo GPON, descrito en G.984.x del ITU-T (*International Telecommunication Union*). Dicho protocolo permite una tasa máxima de 2,5 Gbps en el sentido de bajada (*downstream*), desde el OLT hasta las ONUs/ONTs y 1,25 Gbps en el subida (*upstream*), esto es, desde las ONUs/ONTs.

Como se ha mencionado, la red está desplegada en una topología en árbol. El componente central es el OLT, que da servicio a varias ONUs/ONTs (*Optical Network Unit/Terminal*) que tendrían típicamente los usuarios en sus casas. En el caso de la red de acceso en la que se ha trabajado, dichos componentes han sido diseñados por la empresa

TELNET Redes Inteligentes [1]. Otros componentes importantes son los divisores ópticos, que permiten dividir la señal para poder llegar a más usuarios y, cómo no, la fibra óptica monomodo que conecta el resto de componentes.

Para configurar diferentes servicios en las ONUs/ONTs conectadas al OLT hay dos métodos diferentes: TGMS (*Telnet GPON Management System*), y acceder directamente al CLI (*Command Line Terminal*) del OLT. El primero de ellos, TGMS, consiste en una máquina virtual configurada por TELNET Redes Inteligentes que da acceso a una interfaz web a través de la cual se pueden configurar y ver el estado de los servicios de red que se deseen en las diferentes ONUs/ONTs. El otro método, el CLI del OLT, consiste en conectarse directamente al puerto de gestión del OLT y configurar los servicios a base de introducir comandos a más bajo nivel cercanos al estándar GPON, lo que permite mayor nivel de control que TGMS sobre la configuración de la red.

Por otro lado, OpenFlow es un estándar abierto que permite la creación de diferentes protocolos experimentales. En concreto, en routers y switches clásicos, el *datapath* (envío de datos de un lugar a otro) y el *control path* (toma de decisiones de encaminamiento) ocurrían en el mismo equipo [2]. Utilizando OpenFlow, el *control path* se puede mover a un controlador que se comunicará con el switch donde reside el *datapath* utilizando mensajes OpenFlow. Una de las grandes ventajas de este diseño es que un solo controlador podría operar sobre diferentes switches simultáneamente, permitiendo que las conexiones e interacciones entre ellos puedan variar en función del estado de la red de forma más dinámica. Con esta filosofía, uno de los objetivos de este trabajo ha sido utilizar OpenFlow para que un controlador central configure o modifique los servicios de una red GPON a través de switches virtuales OVS (*Open Virtual Switch*) que se encargarán de limitar el ancho de banda en los canales ascendente y descendente en la ONU/ONT en la que se quiera configurar uno o varios servicios.

Para la implementación del programa central que automatiza la gestión de la red GPON se ha utilizado el lenguaje Python. Se ha elegido este lenguaje por compatibilidad con otros programas relacionados desarrollados con anterioridad, y por lo sencillo que resulta programar cosas relativamente complejas haciendo uso de sus diferentes y variadas librerías.

1.2 Objetivos

A continuación, se explicarán los objetivos que se han seguido para desarrollar el Trabajo de Fin de Máster.

1.2.1 *Objetivo General*

El objetivo principal de este trabajo ha sido desarrollar un programa capaz de configurar una red GPON de forma automatizada utilizando el CLI del OLT o el estándar OpenFlow, a elección del usuario que utilice el programa.

1.2.2 *Objetivos Específicos*

Durante la realización de este trabajo se han cubierto los siguientes objetivos específicos:

1. Análisis de la maqueta de red de acceso GPON: topología, arquitectura y componentes principales que la forman.
2. Diseño de la estructura global del programa en Python.
3. Análisis de los comandos CLI para configurar servicios de internet y video.
4. Análisis del estándar OpenFlow para la configuración de servicios haciendo uso de este protocolo.
5. Puesta en marcha de la programación.

1.3 Fases y Métodos

La metodología seguida para el desarrollo de los objetivos del Proyecto de Fin de Máster ha constado fundamentalmente de las fases que se explicarán a continuación.

1.3.1 *Fase de Análisis*

En esta fase se encuentran los análisis de diferentes aspectos relacionados con el Trabajo de Fin de Máster.

- *Análisis del montaje real de la red GPON*: estudio de la topología y estructura de la red GPON, así como sus diferentes modos de configuración.
- *Análisis de comandos CLI del OLT*: estudio de los comandos necesarios para introducir en el OLT configuraciones de diferentes servicios y perfiles de abonado en la red GPON.
- *Análisis del estándar OpenFlow*: estudio de adecuación del estándar para nuestra red GPON así como de su funcionalidad para limitar el tráfico, y con ello configurar servicios y perfiles de abonado en la red GPON de la forma más análoga posible.

1.3.2 Fase de Configuración

Esta fase tiene como objetivo la configuración de servicios de internet y vídeo, así como perfiles de abonado en una red GPON. Una vez se ha adquirido el suficiente manejo de la red a través de los análisis anteriores, la red se ha de poner en marcha de forma que los usuarios puedan disponer de estos servicios. Se utilizarán los dos modos de gestión introducidos en el programa:

- *Configuración de servicios utilizando el modo de gestión CLI*: una vez analizados los comandos que admite el CLI del OLT, se pasará a hacer configuraciones de diferentes perfiles para los clientes (ONUs/ONTs) mediante dichos comandos de línea.
- *Configuración de servicios utilizando OpenFlow*: se pondrán en marcha las funcionalidades de OpenFlow que permiten configurar dinámicamente los servicios de la red GPON y la gestión de los diferentes parámetros de los mismos, tales como el ancho de banda de un servicio asociado a un cliente (ONU/ONT).

Una vez hechas varias configuraciones mediante CLI y OpenFlow, se pasará automatizar el proceso de configuración con el objetivo de que sea rápido y amigable para el usuario desde un programa global en Python.

1.3.3 Fase de Programación

En esta fase se llevará a cabo la programación en Python. En primer lugar, se hará un análisis de los requisitos que debe cumplir el programa. Una vez se tienen los requisitos, se buscará la mejor manera de llevarlos todos a cabo para implementar las funcionalidades requeridas por el programa. Para desarrollar el programa se han utilizado trabajos de programación anteriores realizados sobre la maqueta, y se ha utilizado el lenguaje de programación Python.

1.3.4 Fase de Realización de los Informes

En la última fase, se ha realizado el informe del Trabajo de Fin de Máster.

1.4 Estructura de la memoria del TFM

En el Capítulo 2 se describirán los principales componentes y metodología utilizados en este Trabajo de Fin de Máster.

En el Capítulo 3 se describirán los requisitos que debe cumplir la aplicación desarrollada, y su estructura.

En el Capítulo 4 se describirá el modo de configuración CLI tanto a nivel teórico como a nivel de programación.

En el Capítulo 5 se describirá el modo de configuración OpenFlow, así como una breve descripción de la integración de SDN (*Software Defined Networks*) mediante OpenFlow y cómo se ha implementado en la red del laboratorio.

En el Capítulo 6 se tratarán las conclusiones a las que se ha llegado tras hacer este Trabajo de Fin de Máster, así como las líneas futuras que se pueden seguir.

2

Metodología y herramientas de trabajo

2.1 Introducción

Este capítulo está dedicado a la realización de un análisis descriptivo de los componentes principales utilizados en este Trabajo de Fin de Máster, esto es, la red de acceso GPON, los diferentes modos de gestión y configuración de la red y finalmente las funcionalidades que se utilizarán del protocolo OpenFlow. Además, también se describirá la metodología de trabajo seguida.

La red de acceso GPON está situada en el laboratorio de Comunicaciones Ópticas de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Valladolid. Los principales elementos de esta red son el OLT (*Optical Line Termination*) y las ONUs/ONTs (*Optical Network Unit/Terminal*) a las que el OLT da servicios. Estos servicios pueden ser configurados de dos formas diferentes: mediante el uso del TGMS (*TELNET GPON Management System*) o, directamente, accediendo al CLI (*Command Line Interface*) del OLT mediante su puerto de configuración. De esos dos modos de configuración va a utilizarse el CLI del OLT en este trabajo, dado que por el momento no es posible integrar el TGMS con programas ajenos a él.

Por su parte, OpenFlow es un estándar que permite la creación de protocolos experimentales a partir de la separación que hace entre *datapath* (envío de un lugar a otro) y *control path* (toma de decisiones). OpenFlow cuenta con varias versiones y controladores, siendo la versión 1.3 y el controlador OpenDayLight los que se van a utilizar en este trabajo.

En cuanto a la metodología, se explicará paso a paso los pasos que se han ido siguiendo; desde los análisis de los diferentes modos de configuración, pasando por la definición de requisitos y, por último, la programación.

2.2 Montaje real y modos de gestión de la red GPON

En esta sección del capítulo se van a describir tanto la estructura y los componentes de la red de acceso como los diferentes modos de gestión con los que cuenta la red GPON desplegada en el laboratorio.

2.2.1 Estructura y componentes de la red de acceso GPON

Una red de acceso es un conjunto de elementos que permiten a usuarios finales conectarse con los proveedores de servicio, de forma que estos puedan darles a dichos usuarios los servicios que han contratado.

Nuestra red de acceso óptica GPON presenta una topología en árbol típica, donde un OLT da servicio a varias ONUs/ONTs. En el caso de la red de nuestro laboratorio, el OLT da servicio a un total de 4 ONUs en el puerto 0, siendo ampliable hasta un máximo de 64 ONUs por puerto. Como el OLT tiene un total de 4 puertos, el OLT puede soportar un total de hasta 256 ONUs. Otros elementos de la red son, por supuesto, la fibra óptica monomodo que conecta los diferentes componentes y los *splitters* (divisores ópticos), responsables de dividir la señal para que cada uno de los puertos del OLT de servicio a varias ONUs. La fibra óptica desplegada en este testbed puede llegar a 25 km (acorde con el estándar GPON), y los divisores ópticos son de razón 1:8 (se dispone de dos, aunque realmente está conectado uno actualmente). El aspecto general que presenta la red de acceso GPON es el que se muestra en la imagen de la Figura 1.

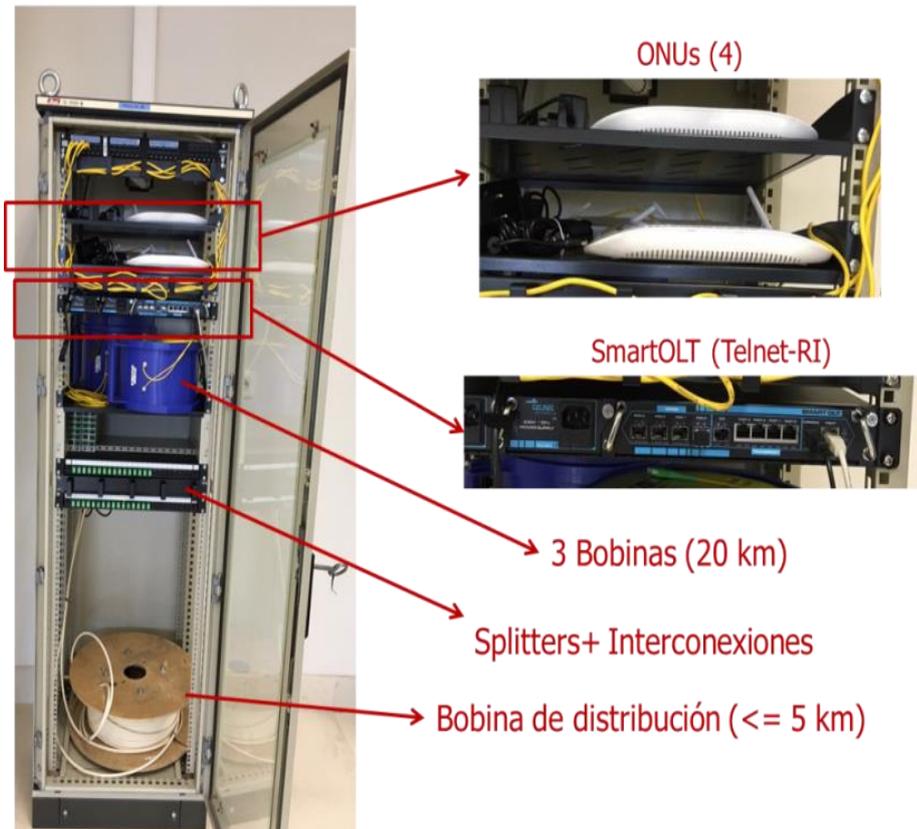


Figura 1: Esquema general de la red de acceso GPON desplegada en el laboratorio

Los componentes de la red de acceso han sido fabricados por la empresa TELNET Redes Inteligentes, de forma que la compatibilidad entre ellos está garantizada. Dependiendo de la dirección que lleven los datos, se puede hablar de dos flujos diferentes:

- **Flujo descendente**: conocido más comúnmente como *downstream*, es el flujo que lleva los datos con las ONUs como destino, esto es, desde fuera de la red hasta los usuarios finales. Según el estándar GPON, la tasa máxima permitida en el *downstream* es de 2.5 Gbps y la longitud de onda de 1490 nm.
- **Flujo ascendente**: conocido más comúnmente como *upstream*, es el flujo que lleva los datos con las ONUs como origen, es es, desde los usuarios finales hacia fuera de la red de acceso. Según el estándar GPON, la tasa máxima permitida en el *upstream* es de 1,25 Gbps y la longitud de onda está situada en segunda ventana 1310 nm.

Es importante tener en cuenta la tasa máxima de estos dos flujos, ya que la suma de las tasas de *upstream* y *downstream* dadas a todos los servicios proporcionados a las diferentes ONUs no pueden exceder los máximos impuestos por el estándar GPON en cada uno de los canales. En caso contrario, la red GPON hará saltar un error en la configuración y puesta en marcha de la misma.

2.2.2 Modos de gestión de la red GPON

A la hora de configurar servicios, existen dos métodos diferentes: el TGMS (*TELNET GPON Management Sstem*) y el CLI (*Command Line Interface*). El TGMS es una máquina virtual que, ejecutándola en Virtualbox con la configuración adecuada, permite al ordenador que la contiene acceder a una página web desde la que se pueden configurar los servicios a cada una de las ONUs conectadas al OLT. Además, también muestra información que puede ser de utilidad, como la potencia recibida en el OLT y las ONUs. Este método está diseñado por el fabricante para que los operadores y proveedores de servicio puedan configurar la red de un modo más amigable.

El otro método, denominado CLI consiste en conectarse mediante telnet a la dirección IP (*Internet Protocol*) y puerto de configuración del OLT para poder acceder a su línea de comandos CLI y configurarlo manualmente a base de comandos propios del estándar GPON. Este método es mucho más complicado que utilizar el TGMS, pero no requiere de la carga computacional que implica ejecutar la máquina virtual del TGMS y permite la creación de programas externos que configuren el OLT de forma automatizada introduciendo los comandos adecuados. Como nuestro objetivo es modificar los servicios que el OLT ofrece a las ONUs de forma automática y sencilla (de cara al usuario), este va a ser uno de los métodos que se van a utilizar.

2.3 Lenguaje de programación Python

Python es un lenguaje de programación interpretado de alto nivel, y ha sido el lenguaje utilizado para desarrollar la aplicación. Su principal ventaja es la versatilidad que ofrece, existiendo una gran variedad de librerías de todo tipo que facilitan mucho el desarrollo e implementación de diferentes funciones y características [3].

Python es un lenguaje multiparadigma, es decir, que admite tanto orientación a objetos como otros paradigmas, como son la programación por procedimientos o la programación funcional. Esto hace que sea un lenguaje de programación muy permisivo y que facilita mucho el desarrollo rápido de código, siendo perfecto para la tarea que se pretende realizar.

Otra de las razones por las que se ha elegido Python es que existen trabajos anteriores desarrollados en este lenguaje, lo que permite una rápida adaptación de las funcionalidades desarrolladas en ellos al nuevo sistema. La versión utilizada de Python para garantizar la compatibilidad entre las diferentes librerías utilizadas ha sido Python 3.5.

2.4 Protocolo OpenFlow

El otro modo de configuración de nuestra maqueta de red GPON que implementará el programa es utilizando OpenFlow.

2.4.1 Estándar OpenFlow

El estándar OpenFlow es un protocolo desarrollado para la implementación de redes SDN (*Software Defined Networking*). Las redes SDN son redes en las que el plano de control está separado del plano de enrutamiento, y en las que el plano de control controla varios dispositivos [4]. Gracias a esta separación, se pueden crear redes directamente programables y ágiles en tanto a que una orden al controlador central es capaz de hacer cambios en la red de una forma muy rápida. Otra ventaja de las redes SDN es que pueden ser configuradas a través de programas, que es lo que se ha hecho en este Trabajo de Fin de Máster.

El estándar OpenFlow consiste en un controlador que opera sobre varios switches OpenFlow, configurando en ellos diferentes flujos OpenFlow capaces de clasificar el tráfico que entra en el switch y actuar sobre él de distintas maneras a través de diferentes mensajes del propio estándar que se estarán intercambiando el controlador con los switches. Típicamente se usan switches virtuales, tales como OVS (*Open Virtual Switch*) [5], que emulan un switch con la capa OpenFlow necesaria para trabajar con este protocolo. El estándar permite una gran variedad de acciones con el tráfico que pasa por

sus flujos, por lo que OpenFlow se puede utilizar para muchos propósitos diferentes, tales como ofrecer Calidad de Servicio (QoS, *Quality of Service*) o modificar la topología de una red en función del tráfico que esté llegando.

Dado que el objetivo de este Trabajo de Fin de Máster es el desarrollo de un programa capaz de configurar servicios en una red GPON, la inclusión de OpenFlow de forma experimental para configurar servicios en ONUs/ONTs es una buena forma de aproximar la filosofía SDN a las redes GPON.

2.4.2 Protocolo OpenFlow para la gestión de servicios en la red GPON

En nuestro caso, nos interesa limitar dinámicamente el tráfico downstream y upstream de una o varias ONUs/ONTs, y de esta forma, configurar nuevos servicios de internet o video con diferentes tasas de bajada y subida en ambos canales de comunicación. La principal ventaja de este tipo de configuración es la flexibilidad que ofrece, ya que es capaz de dar diferentes servicios a un usuario sin modificar en el OLT y en su ONU/ONT ningún tipo de configuración.

Para la limitación del ancho de banda se utilizarán los “medidores” (*meters*) OpenFlow. Los medidores OpenFlow consisten en una entidad que mide el tráfico del flujo OpenFlow al que están asociados en el interior de un OVS (*Open Virtual Switch*). Cuando el tráfico que pasa por el flujo supera una cierta tasa de bits por segundo, es posible conseguir que se descarte el tráfico sobrante, de forma que se limita el tráfico que pasa por un flujo a una determinada tasa.

La limitación de ancho de banda y la configuración de servicios en la red de acceso utilizando OpenFlow se explicará más detalladamente en su capítulo correspondiente.

2.5 Metodología de trabajo

En esta sección del capítulo se describirá la metodología llevada a cabo para desarrollar este Trabajo de Fin de Máster.

2.5.1 Análisis del modo de configuración por línea de comandos del OLT

Como se ha comentado anteriormente, uno de los modos de configurar servicios de red es mediante el CLI del OLT. Para esta configuración, es necesario conectarse remotamente al puerto de gestión del OLT, e introducir los comandos determinados.

Con los scripts proporcionados por TELNET Redes Inteligentes hay varios tipos de configuraciones posibles: servicio de internet, servicio de vídeo y servicio de voz. En este Trabajo Fin de Máster se llevará a cabo la configuración de los servicios de internet y de vídeo, ya que no tenemos un servidor de voz para emular ese servicio. En este sentido, la configuración de ambos tipos de servicio es muy similar, variando solamente en la integración de tres comandos adicionales en el caso de querer configurar un servicio de internet o un servicio de video.

Cada uno de los servicios irá en una VLAN (*Virtual Local Area Network*) diferente, tal y como lo llevan a cabo típicamente los operadores y proveedores de servicio. Además, la configuración de estos servicios se plasmará en la red del usuario de tal modo que en el router de la ONU/ONT correspondiente se configurarán de una o varias interfaces WAN (*Wide Area Network*) de tal modo que cada WAN se corresponda con cada uno de los servicios que se quieren proporcionar a ese cliente. Otros parámetros importantes a la hora de configurar un servicio mediante CLI son el ancho de banda de bajada garantizado, el ancho de banda de bajada en exceso, el ancho de banda de subida garantizado y el ancho de banda de subida en exceso.

En el capítulo correspondiente se profundizará más en el modo de configuración por CLI de servicios y perfiles de abonado en la red GPON.

2.5.2 Análisis del modo de configuración a través de OpenFlow

Utilizando los medidores OpenFlow en el interior de varios OVS se pretende limitar el tráfico que atraviesa el switch y, con ello, poder crear servicios dinámicos de forma que seamos capaces de controlar y cambiar el ancho de banda del cliente sin tener que modificar la configuración del servicio que se le está dando en el OLT y ONU/ONT. Además, mediante este protocolo se pueden controlar otros parámetros de los servicios,

tales como la VLAN de los servicios proporcionados por el operador o proveedor de servicios.

Dado que OpenFlow es un estándar que requiere de un controlador que se encarga de gestionar el tráfico de varios switches OpenFlow, se ha utilizado el controlador OpenDayLight del laboratorio. Existen otros controladores, pero se escogió OpenDayLight debido a que es de código abierto, es compatible con OVS y es uno de los controladores más utilizados en la actualidad al ser un proyecto de Linux Foundation. Además, su análisis comparativo de prestaciones con otros controladores y el estudio más profundo de sus características fue realizado en un Trabajo Fin de Grado anterior [6].

Este controlador opera en una máquina virtual dentro del ordenador central, y tendrá conectividad con varios OVS encargados de limitar el tráfico en el canal de bajada y de subida de la red GPON. El OVS encargado del tráfico del sentido de bajada (downstream) estará operando en el ordenador central, pues controlará el tráfico de entrada hacia dentro de la propia red de acceso. Mientras, el OVS encargado del tráfico del canal de subida (upstream) estará situado entre la ONU/ONT y los equipos conectados a ella, de modo que controle el tráfico que sale de la red de la casa del usuario hacia dentro de la red de acceso.

En el capítulo correspondiente se detallará más sobre el funcionamiento de este tipo de configuración.

2.5.3 Especificación de requisitos y programación

Para desarrollar el programa se ha llevado a cabo una especificación de requisitos que debe cumplir para, posteriormente, llevar a cabo la programación.

En primer lugar, para mantener la compatibilidad con ficheros y scripts desarrollados en trabajos anteriores, se ha utilizado Python para el desarrollo del programa. Dado que Python admite programación orientada a objetos, el programa se ha orientado de forma que cada servicio es un objeto que consta de varios métodos para poder operar con él. Estos métodos son, por ejemplo, crear un servicio, modificarlo, borrarlo o suscribirlo a una ONU/ONT.

Dado que era necesario almacenar algunos parámetros de configuración de los servicios, se ha optado por utilizar una base de datos SQL (*Structured Query Language*) que se encargará de relacionar servicios y ONUs/ONTs. Se ha escogido SQL debido a su simplicidad, pero podría haberse utilizado cualquier tipo de base de datos.

Con todo esto en cuenta, se ha confeccionado una lista de requisitos y un esquema sobre qué y cómo tiene que hacer las cosas el programa. Los detalles sobre los requisitos y el desarrollo aparecerán en los capítulos correspondientes.

2.6 Conclusiones

El laboratorio 2L007 de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Valladolid cuenta con una red de acceso GPON capaz de dar diferentes servicios a las ONUs/ONTs conectadas al OLT.

Los servicios se pueden configurar de dos formas diferentes: CLI del OLT, o TGMS. Se utilizará el modo CLI para poder hacer un programa que configure los servicios de forma personalizada. Sin embargo, en este Trabajo Fin de Máster se plantea además la posibilidad de configurar servicios mediante el estándar OpenFlow colocando switches virtuales (OVS) a la entrada de los elementos activos (OLT y ONUs/ONTs) para limitar el tráfico *upstream* y *downstream* de las ONUs/ONTs de forma dinámica y sin reconfigurar el OLT.

Como objetivo final, se pretende desarrollar un programa global en Python capaz de configurar de forma automatizada tanto por CLI como por OpenFlow diferentes servicios y perfiles de abonado en nuestra maqueta de red GPON. Para ello, se han analizado los scripts CLI proporcionados por TELNET Redes Inteligentes, y el estándar OpenFlow. Una vez llevados a cabos los análisis, se especificarán qué requisitos es necesario que cumpla el programa, y se pondrá en marcha su desarrollo en Python.

3

Descripción global del programa en Python y su funcionamiento

3.1 Introducción

En este capítulo se describirá el programa desarrollado en este Trabajo de Fin de Máster, esto es, los requisitos que debe cumplir para las funcionalidades que se desean implementar, y la estructura que seguirá.

Debido a que se va a utilizar Python, es posible seguir una programación orientada a objetos que implemente los requisitos como métodos a los que se llamará periódicamente en el programa principal.

En cuanto a la estructura del programa, habrá dos partes diferenciadas: la configuración de la red en el modo CLI y el modo OpenFlow. Cada una de estas partes será independiente, aunque compartirán el tipo de servicio que se desee configurar.

3.2 Requisitos y funcionalidades del programa

Antes de hacer cualquier programa, es importante hacer un listado de requisitos que debe cumplir para su correcto funcionamiento:

1. El programa debe ser capaz de crear, ver, modificar o eliminar servicios. La información relativa a estos servicios debe almacenarse en una base de datos. Si se modifica un servicio asociado a una o varias ONUs/ONTs, ese servicio tiene que ser reconfigurado en dichas ONUs/ONTs. Si se elimina

un servicio asociado a una o varias ONUs/ONTs, dicho servicio debe desasociarse de esas ONUs/ONTs.

2. El programa debe ser capaz de asociar uno o varios servicios con una o varias ONUs/ONTs. La información de la asociación deberá almacenarse en una base de datos. Además, no debe ser posible asociar a una ONU/ONT un servicio que ya está asociado a dicha ONU/ONT.
3. El programa debe de ser capaz de eliminar la subscripción de un servicio a una o varias ONU/ONT.
4. La asociación y desasociación de los servicios a una ONU/ONT es independiente del resto de servicios que estén configurados en esa ONU/ONT.
5. Se debe poder configurar la red a través de dos modos de configuración: CLI y OpenFlow. Los servicios se comparten, es decir, son análogos en ambos modos de configuración, pero el modo de asociación es diferente y esto se reflejará en la base la datos.
6. El programa deberá ser robusto y no quedarse bloqueado frente a entradas no válidas o inesperadas.

3.3 Estructura lógica del programa

En esta sección se describirá con más detalle la estructura de la lógica del programa para llevar a cabo los requisitos listados, y de la base de datos que se utilizará.

3.3.1 Lógica del programa

La aplicación puede dividirse en dos modos de funcionamiento: CLI y OpenFlow. Aunque las operaciones que se pueden realizar en ambos modos son similares, la forma de llevarlas a cabo es totalmente diferente. Sin embargo, aparte de estos dos tipos de subscripción de los servicios a los usuarios (ONTs/ONUs), existe una parte importante de la lógica del programa destinada a la gestión de servicios en la red (creación, modificación y/o eliminación) que posteriormente se subscribirán a las ONTs/ONUs.

3.3.1.1 Gestión de Servicios

Los servicios son la entidad principal de la aplicación. Para los servicios se ha creado una clase *Servicio* con diferentes métodos, tales como la creación de un servicio, su modificación o su eliminación. De esta forma, cuando se quiere operar con un servicio, basta con crear un objeto de tipo *Servicio* y llamar al método con el que se quiera operar. En la Figura 2 se muestra la clase *Servicio* con su constructor.

```
class Servicio:
    def __init__(self, gDownstream, gUpstream, excessDownstream, excessUpstream, VLAN, VLANpriority, typeService):
        self.gDownstream=gDownstream
        self.gUpstream=gUpstream
        self.excessDownstream=excessDownstream
        self.excessUpstream=excessUpstream
        self.VLAN=VLAN
        self.VLANpriority=VLANpriority
        self.typeService=typeService
```

Figura 2: Clase Servicio y su constructor

Cada servicio tiene asociado diferentes parámetros: el identificador único del servicio (*id_service*)¹, el ancho de banda downstream garantizado (*gDownstream*), el ancho de banda downstream en exceso (*excessDownstream*), el ancho de banda upstream garantizado (*gUpstream*), el ancho de banda upstream en exceso (*excessUpstream*), la VLAN, la prioridad de la VLAN (*VLANPriority*) y si se trata de un servicio de vídeo o de internet (*typeService*). Todos estos parámetros se almacenarán, modificarán o borrarán en una base de datos que se explicará posteriormente con más detenimiento.

Los servicios son independientes del tipo de configuración que se vaya a utilizar (CLI u OpenFlow) a la hora de asociarlos a los usuarios finales (ONTs/ONUs).

3.3.1.1.1 Configuración de servicios

Para operar con los servicios, hay que introducir la opción 1 (*Service configuration*) en el menú principal mostrado en la Figura 3.

¹ Como se puede observar, *id_service* no se encuentra en el constructor de la Figura 2. Esto se debe a que el identificador único del servicio es proporcionado automáticamente por la base de datos, y no es configurable.

```
Welcome to the service management system.
1. Service configuration
2. CLI Configuration
3. OpenFlow configuration
4. ONT Router configuration
5. Exit
What do you want to do?: 
```

Figura 3: Menú principal de la aplicación

Si se selecciona dicha opción de configurar servicios, el programa llevará al usuario al menú que se muestra en la Figura 4. Tal y como se observa todas las opciones del menú están en inglés para darle mayor potencialidad y uso en diferentes entornos.

```
You have chosen: Service configuration
1. List of configured services
2. Create new service
3. Modify service
4. Delete service
5. Go back
Please, select what do you want to do: 
```

Figura 4: Menú de configuración de servicios

Dentro de este menú, el usuario puede seleccionar si quiere visualizar por pantalla todos los servicios existentes (*List of configured services*), crear un servicio nuevo, modificar un servicio, borrar un servicio o volver al menú anterior.

Si se selecciona la opción 2, crear un servicio (*Create new service*), la aplicación llevará al usuario al menú de la Figura 5. En él, se irán introduciendo todos los parámetros que necesita el servicio uno a uno (enumerados anteriormente).

```
You have chosen: Create new service
Type of service (Internet: 0, Video: 1): 0
VLAN [0-4095]: 806
VLAN Priority [0-7]: 0
Guaranteed downstream (Kbps) [0-2488000]: 30000
Excess downstream (Kbps) [0-2488000]: 0
Guaranteed upstream (Mbps) [0-1244]: 20
Excess upstream (Mbps) [0-1244]: 
```

Figura 5: Creación de un servicio

Una vez creado el servicio, se guarda en la base de datos mediante una petición SQL y se vuelve al menú principal de la Figura 4. Si ahora se desea visualizar todos los servicios (*List of configured services*), como se ve en la Figura 6, el nuevo servicio creado en la Figura 5 aparece listado junto al resto de servicios. En este caso concreto, se trata del servicio con *ID* (identificador de servicio) 22. En este punto, no hay que olvidar que estos identificadores de servicio los da la base de datos, por lo que no se introducen en la creación de servicio de la Figura 5.

ID	Guaranteed Downstream	Excess Downstream	Guaranteed Upstream	Excess Upstream	VLAN	VLAN Priority	Type of service
15	24960.0 Kbps	960.0 Kbps	17.0 Mbps	1.0 Mbps	833	7	Internet
16	19968.0 Kbps	0.0 Kbps	20.0 Mbps	0.0 Mbps	833	0	Internet
17	19968.0 Kbps	0.0 Kbps	20.0 Mbps	0.0 Mbps	806	0	Internet
19	14976.0 Kbps	0.0 Kbps	10.0 Mbps	0.0 Mbps	806	0	Video
20	24960.0 Kbps	0.0 Kbps	60.0 Mbps	0.0 Mbps	833	0	Video
21	19968.0 Kbps	0.0 Kbps	10.0 Mbps	0.0 Mbps	833	0	Video
22	29952.0 Kbps	0.0 Kbps	20.0 Mbps	10.0 Mbps	806	0	Video

Press enter to continue...

Figura 6: Listado de servicios existentes en la base de datos

De vuelta al menú de la Figura 4, si se selecciona la opción 3 (*Modify service*) será posible modificar uno de los servicios, eso es, alguno de los parámetros del propio servicio. En la Figura 7 puede verse la funcionalidad de modificación de servicios.

```
New configuration:
```

Guaranteed Downstream	Excess Downstream	Guaranteed Upstream	Excess Upstream	VLAN	VLAN Priority	Type of service
29952.0 Kbps	0.0 Kbps	20.0 Mbps	10.0 Mbps	806	0	Video

What parameter do you want to modify?

1. Guaranteed downstream
2. Excess downstream
3. Guaranteed upstream
4. Excess upstream
5. VLAN
6. VLAN Priority
7. Modify!

Please, select what do you want to do:

Figura 7: Modificación de servicios

En este menú, el usuario primero selecciona qué parámetros quiere modificar uno a uno, y una vez hechos todos los cambios deseados se selecciona la opción 7 (*Modify!*), que hará que la nueva configuración con todos los cambios realizados se guarde en la base de datos con una petición SQL.

La última opción del menú de la Figura 4 es borrar un servicio (*Delete service*). Al seleccionar esta opción, se irá al menú de la Figura 8.

```
You have chosen: Delete service
```

ID	Guaranteed Downstream	Excess Downstream	Guaranteed Upstream	Excess Upstream	VLAN	VLAN Priority	Type of service
15	24960.0 Kbps	960.0 Kbps	17.0 Mbps	1.0 Mbps	833	7	Internet
16	19968.0 Kbps	0.0 Kbps	20.0 Mbps	0.0 Mbps	833	0	Internet
17	19968.0 Kbps	0.0 Kbps	20.0 Mbps	0.0 Mbps	806	0	Internet
19	14976.0 Kbps	0.0 Kbps	10.0 Mbps	0.0 Mbps	806	0	Video
20	24960.0 Kbps	0.0 Kbps	60.0 Mbps	0.0 Mbps	833	0	Video
21	19968.0 Kbps	0.0 Kbps	10.0 Mbps	0.0 Mbps	833	0	Video
22	29952.0 Kbps	1984.0 Kbps	20.0 Mbps	10.0 Mbps	806	0	Video

```
Select the ID of the service which you want to delete: 21
```

Figura 8: Borrado de un servicio

Desde esta funcionalidad, simplemente se muestran todos los servicios configurados y se selecciona el que se quiere borrar utilizando su ID. Una vez seleccionado, el programa manda la petición SQL para borrar dicho servicio de la base de datos y ninguna ONT/ONU podrá subscribirse a este servicio.

3.3.1.2 Subscripción de servicios mediante CLI

Para utilizar el modo de configuración CLI, el programa debe de ser capaz de conectarse al puerto de gestión del OLT al realizar las operaciones de asociar o subscribir un servicio con una ONU/ONT, o desasociarlo. Una vez introducidos los comandos necesarios para configurar o desconfigurar un servicio, se guardan en un fichero de log y se modifica la base de datos para que quede constancia de la operación que se acaba de realizar. Mediante CLI es posible tanto asociar como desasociar servicios a una ONU/ONT.

3.3.1.2.1 Asociación de un servicio con una ONU/ONT

Para asociar o desasociar un servicio mediante CLI a una ONU/ONT, basta con seleccionar la opción 2 en el menú principal (Figura 3). Al seleccionar esta opción, el usuario irá al menú mostrado en la Figura 9.

```
You have chosen: CLI Config
1. Attach service to ONT
2. Detach service from ONT
3. Go back
What do you want to do?:
```

Figura 9: Menú CLI

En este menú, se selecciona la opción 1 y se llegará al menú de la Figura 10.

```
You have chosen: attach service to ONT
```

ID	Guaranteed Downstream	Excess Downstream	Guaranteed Upstream	Excess Upstream	VLAN	VLAN Priority	Type of service
15	24960.0 Kbps	960.0 Kbps	17.0 Mbps	1.0 Mbps	833	7	Internet
16	19968.0 Kbps	0.0 Kbps	20.0 Mbps	0.0 Mbps	833	0	Internet
17	19968.0 Kbps	0.0 Kbps	20.0 Mbps	0.0 Mbps	806	0	Internet
19	14976.0 Kbps	0.0 Kbps	10.0 Mbps	0.0 Mbps	806	0	Video
20	24960.0 Kbps	0.0 Kbps	60.0 Mbps	0.0 Mbps	833	0	Video
22	29952.0 Kbps	1984.0 Kbps	20.0 Mbps	10.0 Mbps	806	0	Video

```
Select the ID of the service which you want to attach to the ONT: 16
```

ID	ONT
0	54-4c-52-49-5b-01-f7-30
1	54-4c-52-49-5b-01-f6-90
2	54-4c-52-49-5b-01-f7-28

```
Select ID: 2
```

Figura 10: Asociación de un servicio con una ONU/ONT mediante CLI

Como se puede observar, primero aparece la lista de servicios configurados en la base de datos. De esa lista, se escoge la ID del servicio que se quiere asociar a una ONT/ONU. Posteriormente, se muestran las ONUs/ONTs conectadas al OLT, junto al identificador (ID de ONT/ONU) que les ha dado el OLT. Tras escoger una ONU/ONT con su ID, el servicio que se ha elegido en el paso anterior se configura y se guarda en la base de datos esta asociación mediante una petición SQL.

Para realizar dicha subscripción del servicio a la ONT/ONU, se recogen de la base de datos los parámetros del servicio a configurar y se crea un objeto con ellos. El código de esta operación se puede ver en la Figura 11.

```
options=Service.getService(id_service)
service = Service(options[1], options[3], options[2], options[4], options[5], options[6],options[7])
```

Figura 11: Creación de objeto servicio

El método de la clase `Service.getService()` se encarga de cargar los datos del servicio en la lista de cadenas de caracteres `options`, que luego pasan a ser los parámetros de entrada del constructor para el objeto `service`. Tras crear este objeto, se configura el servicio mediante CLI llamando al método de configuración desde el objeto que se acaba de crear, denominado `service.servicio_Internet_Video()`. En la lógica de este método, se realiza en primer lugar una conexión telnet al puerto de gestión del OLT, y a continuación se introducen automáticamente los comandos CLI necesarios para configurar el servicio, es decir, para asociar dicho servicio a la ONU/ONT elegida. En la Figura 12 se muestra la llamada al método de configuración desde el objeto creado anteriormente.

```
service.servicio_Internet_Video(int(id_onu),MAC[int(id_onu)],id_service,n_service[0],WANinst)
```

Figura 12: Llamada al método servicio_Internet_Video

Tal y como se observa en la Figura 12, los parámetros de entrada al método de asociación de un servicio a una ONU/ONT *service.servicio_Internet_Video()* son:

1. La ID de la ONU/ONT asignada por el OLT y a la que se pretende suscribir un servicio
2. El número de serie de identificación de la ONU/ONT (llamado MAC en el código por su similitud con una dirección MAC)
3. El identificador del servicio (ID del servicio) que se pretende configurar en la ONU/ONT
4. El número de servicios que hay configurados en la ONU/ONT
5. El número de la interfaz WAN de la ONU/ONT (creada en el router) a la que se asociará el servicio

Dentro del método, los comandos CLI introducidos crean varias instancias a nivel GPON en el interior del OLT capaces de configurar el servicio en la ONU/ONT. Los números utilizados para la creación de estas instancias se guardarán en la tabla de la base de datos que relaciona las ONUs/ONTs con los servicios, ya que serán necesarios para poder borrar dichas instancias a la hora de eliminar la suscripción del servicio a una o varias ONUs/ONTs. Es importante destacar que este método descrito sirve tanto para asociar servicios de vídeo como para servicios de internet, pero que la introducción de comandos CLI varía en función del tipo de servicio que se desea suscribir.

En la Figura 13 se muestra un resumen del servicio que se acaba de configurar en una ONU/ONT, y que aparece tras realizar la suscripción a través del método implementado.

```

| 1 | 54-4c-52-49-5b-01-f6-90 |
| 2 | 54-4c-52-49-5b-01-f7-28 |
+-----+
Select ID: 2
Numero de servicios ya configurados: 0
El identificador VLAN para el servicio 1 es: 833

La prioridad VLAN para el servicio 1 es: 0

El ancho de banda Downstream garantizado en Kbps para el servicio 1 es: 19968
El ancho de banda Downstream en exceso en Kbps para el servicio 1 es: 0

El ancho de banda Upstream garantizado en Mbps para el servicio 1 es: 20

El ancho de banda Upstream BE en Mbps para el servicio 1 es: 20

Servicio de Internet configurado.
Enter to continue...
    
```

Figura 13: Resumen del servicio configurado

3.3.1.2.2 Desasociación de un servicio a una ONU/ONT

La desasociación de un servicio configurado mediante CLI se realiza seleccionando la opción 2 del menú CLI del programa (Figura 9). En la Figura 14 se muestra el menú en la que se elimina la suscripción de un servicio, así como lo que se explica a continuación.

```

You have chosen: detach service from ONT
+-----+
| ID | ONT |
+-----+
| 0 | 54-4c-52-49-5b-01-f7-30 |
| 1 | 54-4c-52-49-5b-01-f6-90 |
| 2 | 54-4c-52-49-5b-01-f7-28 |
+-----+
Select ID: 2
+-----+
| ONT | ID | Guaranteed Downstream | Excess Downstream | Guaranteed Upstream | Excess Upstream | VLAN | VLAN Priority | Type of service |
+-----+
| 54-4c-52-49-5b-01-f7-28 | 16 | 19968.0 Kbps | 0.0 Kbps | 20.0 Mbps | 0.0 Mbps | 833 | 0 | Internet |
+-----+
Select the ID of the service which you want to detach from the ONT:
    
```

Figura 14: Desasociación de un servicio y una ONU/ONT mediante CLI

En este caso, al contrario que en el caso de la asociación de servicio mediante CLI, primero se muestran las ONUs/ONTs conectadas al OLT con su correspondiente identificador (ID de la ONUs/ONTs). De esta forma, se usa este identificador para seleccionar la ONU/ONT de la que se quiere desasociar un servicio.

Tras seleccionar la ONU/ONT, se hace una petición SQL para saber qué servicios están configurados actualmente en esa ONU/ONT. Dichos servicios se muestran listados para que el usuario sea capaz de seleccionar la ID del servicio que pretende desasociar.

En el caso de la desasociación de un servicio se llama al método de la clase *service.borrar_configuracion()* capaz de introducir los comandos necesarios en el OLT para eliminar todas las instancias creadas en la configuración de un servicio (Figura 15).

```
Service.borrar_configuracion(int(id_onu),MAC[int(id_onu)],id_service)
```

Figura 15: Llamada al método borrar_configuracion

Los parámetros de entrada a este método son:

1. El identificador de la ONU/ONT (ID) asignada por el OLT y a la que se pretende dar servicio
2. El número de serie de identificación de la ONU/ONT
3. El identificador del servicio (ID del servicio) que se pretende desasociar de la ONU/ONT

En el interior del método, se cargan los parámetros CLI de las instancias del OLT creadas al configurar el servicio desde la base de datos. Después, utilizando estos parámetros, se procede a eliminar todas las instancias del OLT pertenecientes al servicio a eliminar en la ONU/ONT concreta.

Una vez se realiza la desasociación, se hace una sentencia SQL para quitar la asociación de la base de datos, pero no se borrará el servicio a nivel global.

3.3.1.3 Suscripción de servicios OpenFlow

El modo de suscripción de servicios a una o varias ONUs/ONTs mediante OpenFlow es totalmente ajeno al OLT, como evidencia su opción de configuración propia en el menú principal de la aplicación de la Figura 3 (*OpenFlow Configuration*). Al seleccionar la opción de OpenFlow, se entrará en un nuevo menú que se muestra en la Figura 16.

```
You have chosen: OpenFlowConfig
1. Attach OpenFlow service to ONT
2. Detach OpenFlow service from ONT
3. Go back
What do you want to do?: 
```

Figura 16: Menú de configuración OpenFlow

Dentro de este nuevo menú (*OpenFlowConfig*) las opciones 1 y 2, suscribir y desuscribir respectivamente, funcionan exactamente igual que en el caso de configuración CLI de cara al usuario, aunque la configuración en sí se realice de forma diferente.

En este caso, el programa mandará varios flujos y medidores mediante peticiones HTTP PUT o DELETE al controlador OpenDayLight en función de si se quiere crear o eliminar un servicio respectivamente. En la Figura 17 se puede ver el flujo para el OVS encargado de la subida.

```

flowUp='' {
  "flow": [
    {
      "id": "flow'+str(40+int(id_service))+'",
      "match": {
        "ethernet-match": {
          "ethernet-destination": {
            "address": "'+mac_upstream+'
          }
        }
      },
      "instructions": {
        "instruction": [
          {
            "order": 5,
            "meter": {
              "meter-id": "'+str(int(self.gupstream)+'
            }
          },
          {
            "order": "1",
            "apply-actions": {
              "action": [
                {
                  "order": "1",
                  "output-action": {
                    "output-node-connector": "NORMAL"
                  }
                }
              ]
            }
          }
        ]
      },
      "flow-name": "flow'+str(40+int(id_service))+'",
      "priority": "8000",
      "idle-timeout": "0",
      "hard-timeout": "0",
      "cookie": "478478457845784600",
      "table_id": "0"
    }
  ]
}
'''

```

Figura 17: Flujo encargado de la subida

Como se puede observar, se trata de una cadena en formato JSON (*JavaScript Object Notation*) que contiene todos los datos de configuración del flujo. El otro flujo y los medidores siguen un esquema similar.

Los datos enviados en esos flujos y medidores variarán en función de algunos parámetros del servicio a borrar o configurar, como se ve en la Figura 15.

Una vez enviada la orden al controlador, este será el encargado de mandar los flujos y medidores a los OVS para bajada y subida que se encargarán de limitar el tráfico y, en definitiva, de configurar el nuevo servicio.

Tras realizar cualquier operación, se actualizará la base de datos en consecuencia. La configuración, borrado y subscripción de servicios a las diferentes ONTs/ONUs mediante OpenFlow se explicará con más detenimiento en capítulos posteriores.

3.3.2 Estructura de la base de datos

Para guardar la configuración de servicios, los identificadores de las ONUs/ONTs conectadas al OLT, y las asociaciones servicio-ONU/ONT se ha utilizado una base de datos SQL. La base de datos consta de tres tablas diferentes: en una se almacenarán los identificadores de las ONUs/ONTs (que se corresponden con el número de serie de estas y son unívocos) que se han conectado alguna vez al OLT, en otra se almacenarán cada uno de los servicios y sus parámetros, y en la última se relacionarán las ONUs/ONTs con los servicios que tengan configurados. Además, la tabla que relaciona las ONUs/ONTs con los servicios también almacenará diferentes parámetros de configuración necesarios si el servicio ha sido asociado mediante CLI.

Los identificadores de las ONUs/ONTs y de los servicios se han configurado como PRIMARY KEY, de forma que no puedan duplicarse. El resto de los parámetros se han configurado como cadenas de caracteres, números enteros y números float.

3.3.2.1 Esquema de la base de datos

En la Tabla 1 se muestran todos los parámetros de la tabla *services*, así como el tipo de dato de cada entrada.

id_service	gDownstream	excessDownstream	gUpstream	excessUpstream	VLAN	typeService	VLANPriority
Primary key, int	float	Float	Float	float	int	Varchar (255)	int

Tabla 1: Tabla Services

En la Tabla 2 se muestra la tabla *onts*. Esta tabla es muy sencilla, pues solo almacena el número de serie de cada una de las ONUs/ONTs que se han conectado alguna vez al OLT.

id_ont
Varchar (255)

Tabla 2: Tabla Onts

Por último, en la Tabla 3 se muestra la tabla que relaciona las ONUs/ONTs de la tabla *Onts* con los servicios de la tabla *Services*.

id_ont	id_service	alloc_port	pointer	profile	instance	WANinstance	configuration
int	int	Int	int	Int	Int	int	Varchar (255)

Tabla 3: Tabla ont_service

Los parámetros *alloc_port*, *pointer*, *profile* e *instance* son parámetros que hacen referencia a las diferentes instancias que se crean en el OLT al configurar un servicio mediante CLI, y en el siguiente capítulo se especificará más qué es lo que significan. Estos campos no se rellenan al configurar un servicio mediante OpenFlow, puesto que en ese caso no son necesarios. Por su parte, *WANinstance* es el número de la interfaz WAN que usará el servicio *id_service* configurado en la ONU/ONT *id_ont*, mientras que *configuration* indicará si se trata de una configuración CLI o una configuración OpenFlow.

3.3.3 Conclusiones

Para llevar a cabo el desarrollo del programa se ha comenzado pensando en los requisitos que debe cumplir. Después, se ha pensado en una estructura orientada a objetos donde el centro de todo son los servicios que se desea configurar. Estos servicios admitirán operaciones como creación o eliminación, y posteriormente se podrán asociar a una o varias ONUs/ONTs mediante configuración CLI u OpenFlow. Se ha utilizado una base de datos SQL para almacenar los parámetros necesarios para el correcto funcionamiento de la aplicación.

4

Configuración de servicios mediante CLI

4.1 Introducción

El OLT, u *Optical Line Termination*, es el encargado de ofrecer los servicios en el tramo final de la fibra óptica en el que se encuentran las ONUs/ONTs, como se muestra en la Figura 16. A través de un OLT pueden darse diferentes tipos de servicios, como pueden ser servicios de internet, de vídeo o de voz.

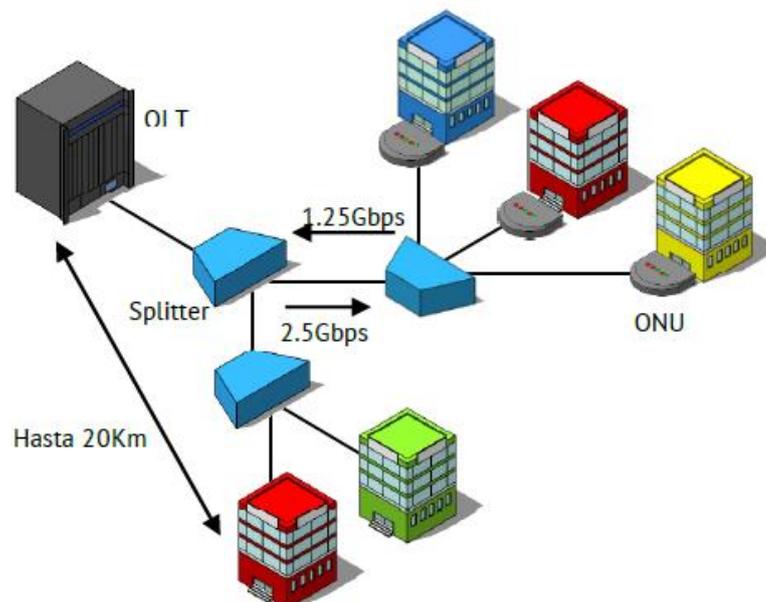


Figura 18: Ejemplo escenario GPON

Para que los operadores puedan controlar y gestionar estos servicios, es necesario que haya herramientas intuitivas y capaces de configurar diferentes aspectos del OLT. En el caso del OLT del laboratorio 2L007, hay dos modos de hacer esto: TGMS y CLI.

El TGMS, o *TELNET GPON Management System* es un sistema de gestión desarrollado por Telnet Redes Inteligentes. En él, se permite la creación de servicios o paquetes de servicios que posteriormente son asignables a las ONUs/ONTs conectadas al OLT. Aunque es una herramienta muy útil y completa, lo cierto es que su funcionamiento interno entre ella y el OLT es totalmente opaco. Debido a ello, no pueden estudiarse adecuadamente las operaciones que se realizan en el OLT cuando se pretende configurar o desconfigurar un servicio mediante TGMS.

Por otro lado, la otra opción es el CLI del OLT. En este modo, se realiza una conexión telnet al puerto de gestión del OLT y es posible realizar diferentes configuraciones introduciendo comandos navegando por los diferentes menús de los que consta el OLT. La ventaja de este modo de funcionamiento es que los comandos indican qué se está haciendo en el OLT, por lo que son muy útiles para el estudio de las operaciones internas del OLT y el protocolo GPON. Su desventaja es, sin embargo, que la complejidad de la gestión de servicios mediante CLI es mucho mayor que en TGMS.

Sin embargo, CLI tiene otra ventaja, ya que hace posible automatizar la introducción de comandos con facilidad mediante scripts en diferentes lenguajes de programación. Debido a ello, como el objetivo de este Trabajo de Fin de Máster es el desarrollo de una aplicación capaz de gestionar la red del laboratorio, se utilizará el modo de gestión mediante CLI para ser integrado de forma transparente mediante Python.

4.2 Estructura jerárquica de los comandos CLI

El CLI del OLT sigue una estructura jerárquica de menús en los que se irán introduciendo los comandos. En la Figura 19 se muestra gráficamente esta jerarquía. Para configurar los servicios habrá que navegar por los siguientes menús:

- **Initial Menu:** este es el menú inicial, en el que se estará nada más hacer la conexión remota con el puerto de gestión del OLT.

- **Privileged Mode:** introduciendo “enable” en el menú anterior, se entra en el modo con privilegios.
- **Configuration:** introduciendo “configure” en el menú anterior, se accede al menú de configuración. A partir de aquí pueden empezar las configuraciones.
- **OLT device configuration mode:** introduciendo “olt-device OLT_DEVICE_INDEX” en el menú Configuration se utiliza OLT_DEVICE_INDEX para seleccionar el puerto del OLT en el que está la ONU/ONT con la que se va a trabajar.
- **OLT channel configuration:** introduciendo en el menú anterior “olt-channel OLT_CHANNEL_INDEX” se utiliza OLT_CHANNEL_INDEX para seleccionar el canal al que se conectará la ONU/ONT con la que se va a trabajar.
- **Local ONU configuration mode:** aquí se introducen los comandos que configuran aspectos de la ONU/ONT. Se accede a este menú introduciendo “onu-local ONU_INDEX” en el menú OLT channel configuration, seleccionando la ONU/ONT que interesa a través de ONU_INDEX.
- **OMCI ONU configuration mode:** aquí se introducen los comandos que configuran aspectos que tienen que ver con el canal OMCI (*ONT Management and Control Interface*) entre ONU/ONT y OLT. Se accede a este menú desde OLT channel configuration introduciendo “onu-omci ONU_INDEX”, seleccionando la ONU/ONT que interesa a través de ONU_INDEX.
- **PON interface configuration mode:** se accede introduciendo “pon” desde el menú OLT device configuration mode. En este menú pueden gestionarse los aspectos que tienen que ver con las interfaces PON (*Passive Optical Network*).
- **DBA configuration mode:** se accede introduciendo “dba pythagoras 0” en el menú PON interface configuration mode. En este menú pueden gestionarse aspectos que tienen que ver con el DBA (*Dynamic Bandwidth Allocation*).

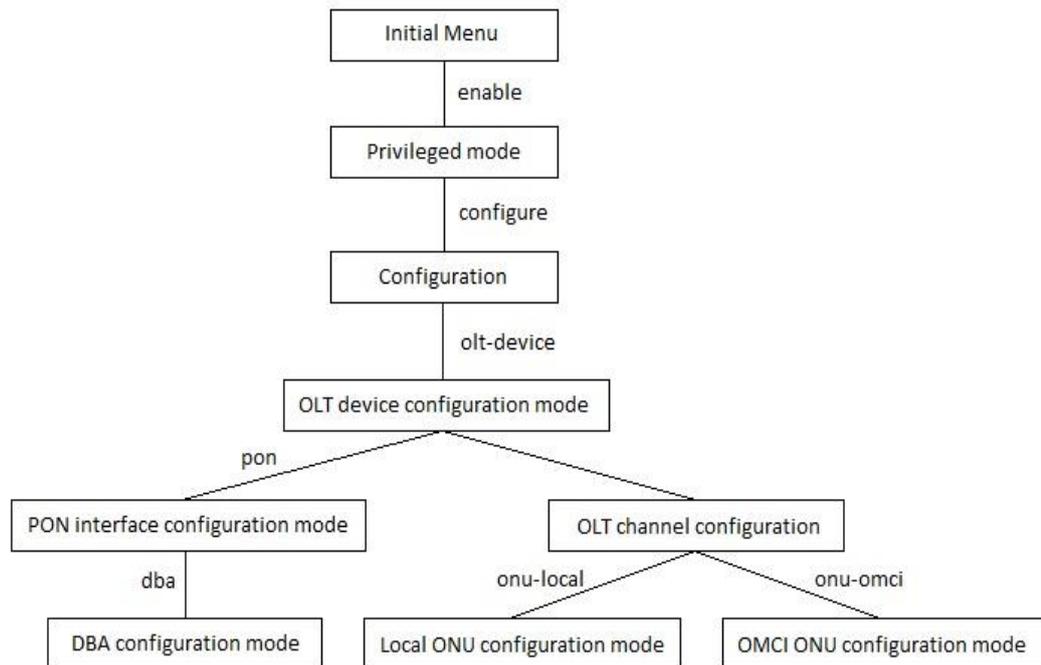


Figura 19: Estructura jerárquica del CLI

4.3 Configuración de servicios mediante CLI

4.3.1 Configuración de servicio de Internet

A continuación, se describirán los comandos utilizados para configurar un servicio de internet.

Lo primero que hay que hacer es ir a *OLT channel configuration*, seleccionado “OLT_DEVICE_INDEX = 0” y “OLT_CHANNEL_INDEX = 0”. Es decir, la primera secuencia de comandos será:

- *enable*
- *configure*
- *olt-device 0*
- *olt-channel 0*

Desde aquí, el siguiente paso es activar el canal OMCI con la ONU/ONT en la que se quiere configurar el servicio. Para ello, se introducen los siguientes comandos:

- *onu-local ID_ONU*
- *omci-port ID_ONU*

- *exit*

Donde *ID_ONU* es un número entero con el que el OLT está identificando a la ONU/ONT a la que se quiere dar servicio. Es importante señalar que este identificador no es invariable, y que la misma ONU/ONT puede tener asignado diferente *ID_ONU* al encender el OLT.

Como se ha usado el comando *exit*, se ha vuelto a *OLT channel configuration*. Desde aquí, se va a *OMCI ONU configuration mode* para reiniciar las entidades MIB (*Management Information Base*). Desde ahí, se vuelve a *OLT channel configuration*, se activa FEC (*Forward Error Connection*) en el flujo ascendente y se vuelve al menú *Local ONU configuration mode*, esto es, según la siguiente secuencia:

- *onu-omci ID_ONU*
- *ont-data mib-reset*
- *exit*
- *fec direction uplink*
- *onu-local ID_ONU*

El siguiente paso es crear el *alloc-id* perteneciente al servicio de internet que se va a configurar. El *alloc-id*, o *Allocation Identifier*, es un número que se asigna entre OLT y ONU. Identifica a una entidad portadora de tráfico (T-CONT) receptora de asignaciones de ancho de banda ascendentes dentro de esa ONU/ONT, es decir, la cola en la que se almacena el tráfico de dicha prioridad:

- *alloc-id ALLOC_ID*
- *exit*

El número entero *ALLOC_ID* introducido puede ser cualquiera, y se utilizará uno diferente por cada servicio que se configure. Tras crear el *alloc-id*, se sale al menú anterior (*OLT Channel Configuration*).

A continuación, hay que asociar un *GEM Port* (*GPON Encapsulation Method Port*) al *alloc-id* que se acaba de crear. Para ello, se introduce el siguiente comando:

- *port PORT-ID alloc-id ALLOC-ID*

Para evitar problemas, se tomará como *PORT-ID* el mismo número que *ALLOC-ID*. De esta forma, cada servicio tiene un *alloc-id* asociado a un *GEM Port*.

Por otra parte, el T-CONT (*Transmission Container*) es un objeto de la ONU/ONT que representa un grupo de conexiones lógicas que aparecen como una sola entidad con el propósito de que se le asigne ancho de banda ascendente en la red óptica. La asignación de ancho de banda ascendente, por su parte, se realiza gracias al algoritmo DBA (*Dynamic Bandwidth Allocation*). Este algoritmo dará ancho de banda ascendente en franjas de tiempo definidas en el T-CONT, y de esta forma se hará la asignación de ancho de banda. El T-CONT se crea en el menú *OMCI ONU configuration mode*, y se asocia a un *alloc-id* que se encargará de identificarlo:

- *onu-omci ID_ONU*
- *t-cont set slot-id 128 t-cont-id TCONT_ID alloc-id ALLOC_ID*

Para entender mejor la relación entre T-CONT, *alloc-id* y GEM Port se puede observar la Figura 20.

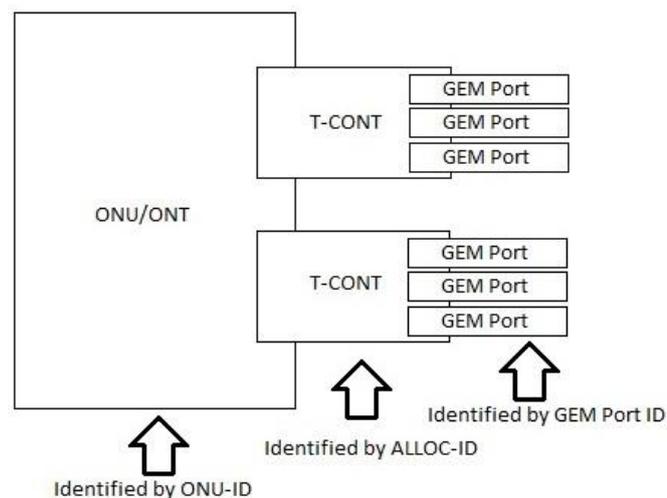


Figura 20: Relación entre T-CONT, *alloc-id* y GEM Port

Cabe destacar que el *alloc-id*, que puede tener uno o más *GEM Ports* asociados, identifica al TCONT. No obstante, para no aumentar la complejidad de las configuraciones, siempre se harán configuraciones 1:1 entre *alloc-id*, *GEM Port* y T-CONT por servicio.

A continuación, deben crearse el *MAC Bridge Service Profile* y los *MAC Bridge Port Configuration Data*. A través de *bridge-group-id* se relaciona el *MAC Bridge Service Profile* con los *MAC Bridge Port Configuration Data*, de modo que los comandos pertinentes a introducir en orden son los siguientes:

- *mac-bridge-service-profile create slot-id 0 bridge-group-id 1 spanning-tree-ind true learning-ind true atm-port-bridging-ind true priority 32000 max-age 1536 hello-time 256 forward-delay 1024 unknown-mac-address-discard false mac-learning-depth 255 dynamic-filtering-ageing-time 1000*
- *mac-bridge-pcd create instance 1 bridge-id-ptr 1 port-num 1 tp-type lan tp-ptr 257 port-priority 2 port-path-cost 32 port-spanning-tree-ind true encap-method llc lanfcs-ind forward*
- *mac-bridge-pcd create instance NUM_INST bridge-id-ptr 1 port-num NUM_INST tp-type gem tp-ptr NUM_INST port-priority 0 port-path-cost 1 port-spanning-tree-ind true encap-method llc lanfcs-ind forward*

Tal y como se observa, siempre se crean varias instancias para los *MAC Bridge PCD*: la primera de ellas aparece siempre, mientras que la segunda es relativa al servicio particular que se esté configurando. Cada *NUM_INST* identificará un servicio en una ONU/ONT en particular.

El siguiente paso es la creación del *GEM Port Network CTP*, asociado al Puerto GEM asociado al servicio, y con un puntero que será diferente por cada servicio configurado.

- *gem-port-network-ctp create instance NUM_INST port-id PORT_ID t-cont-ptr PUNTERO direction bidirectional traffic-mgmt-ptr-ustream 0 traffic-descriptor-profile-ptr 0 priority-queue-ptr-downstream 0 traffic-descriptor-profile-ds-ptr 0 enc-key-ring 0*

A continuación, hay que crear los *GEM Interworking Termination Point*, encargados de transformar las tramas GEM en flujos de bytes y viceversa. A través de *gem-port-nwk-ctp-conn-ptr* se vinculará al *GEM Port Network CTP* utilizando el mismo número que la instancia creada en el comando anterior (*NUM_INST*). Además, poniendo *service-profile-ptr* a 1 e *interwork-option mac-bridge-lan*, junto a un número de instancia igual al de los *MAC Bridge PCD* se consigue asociar los *GEM Interworking Termination Point* con los *MAC Bridge PCD*:

- *gem-interworking-termination-point create instance NUM_INST gem-port-nwk-ctp-conn-ptr NUM_INST interwork-option mac-bridge-lan service-profile-ptr 1 interwork-tp-ptr 0 gal-profile-ptr 0*

A continuación, para poder usar la VLAN con la prioridad que requiera el servicio, hay que introducir el siguiente comando:

- *vlan-tagging-filter-data create instance NUM_INST forward-operation h-vid-a vlan-tag1 VLAN vlan-priority1 PRIORITY vlan-tag2 null vlan-priority2 null vlan-tag3 null vlan-priority3 null vlan-tag4 null vlan-priority4 null vlan-tag5 null vlan-priority5 null vlan-tag6 null vlan-priority6 null vlan-tag7 null vlan-priority7 null vlan-tag8 null vlan-priority8 null vlan-tag9 null vlan-priority9 null vlan-tag10 null vlan-priority10 null vlan-tag11 null vlan-priority11 null vlan-tag12 null vlan-priority12 null*

Los próximos comandos que hay que introducir son la creación de *Extended VLAN Tagging Operation Config Data* y su configuración. El número de instancia debe coincidir con el *tp-ptr* del primer *MAC Bridge PCD*. El segundo comando sirve para configurar el primero:

- *extended-vlan-tagging-operation-config-data create instance 257 association-type ptp-eth-uni associated-me-ptr 257*
- *extended-vlan-tagging-operation-config-data set instance 257 operations-entry filter-outer-prio filter-prio-no-tag filter-outer-vid none filter-outer-tpid none filter-inner-prio filter-prio-none filter-inner-vid VLAN filter-inner-tpid none filter-ethertype none treatment-tag-to-remove 1 treatment-outer-prio none treatment-outer-vid copy-from-inner treatment-outer-tpid tpid-de-copy-from-outer treatment-inner-prio 0 treatment-inner-vid VLAN treatment-inner-tpid tpid-de-copy-from-inner*
- *exit*

Tras poner los comandos relativos a la VLAN, se usa *exit* para salir al menú anterior *OLT Channel configuration*. Allí, se introducen los siguientes comandos:

- *vlan uplink configuration port-id PORT_ID min-cos 0 max-cos 7 de-bit disable primary-tag-handling false*
- *vlan uplink handling port-id PORT_ID primary-vlan none destination datapath c-vlan-handling no-change s-vlan-handling no-change new-c-vlan 0 new-s-vlan 0*

Estos comandos sirven para configurar las reglas VLAN en el OLT asociadas al Puerto GEM que se usará para configurar el servicio.

A continuación, se crean los perfiles de ancho de banda en sentido descendente. Aquí es donde se introducirán el ancho de banda de bajada y el ancho de banda de bajada en exceso del servicio a través del siguiente comando (Los anchos de banda del flujo descendente van en Kbps):

- *policing downstream profile committed-max-bw G_DBW committed-burst-size 1023 excess-max-bw E_DBW excess-burst-size 1023*

Al introducir este comando, el OLT devolverá un número (*downstream_profile_index*). A continuación, hay que asociar ese número del perfil devuelto con con el puerto GEM del servicio usando el siguiente comando:

- *policing downstream port-configuration entity port-id PORT_ID ds-profile-index DS_PROFILE_INDEX*

Finalmente, solo queda terminar de configurar el ancho de banda en el sentido ascendente. Para ello, hay que ir al menú *dba configuration* y asociar un perfil de subida al *alloc-id* utilizado por el servicio con el ancho de banda de subida garantizado y ancho de banda de subida en exceso que se requiera. Para ellos se introducen los siguientes comandos:

- *exit*
- *pon*
- *dba pythagoras 0*

- *sla ALLOC_ID service data status-report nsr gr-bw G_UBW gr-fine 0 be-bw E_UBW be-fine 0 \n*

En este caso, el ancho de banda va en Mbps, y en el caso del ancho de banda de subida en exceso que se introduce en el comando se trata de la suma del garantizado más el ancho de banda de subida “extra” que se pretende permitir al servicio.

4.3.2 Configuración de servicio de video

La configuración de un servicio de video es esencialmente igual que en uno de internet, pues en realidad se está configurando un servicio de internet más uno de video. La diferencia entre esta configuración y la anterior está en tres comandos.

El primero de ellos crea un segundo *MAC Bridge PCD* independiente de los servicios. Tomando como referencia los comandos de creación de servicio de internet, este iría entre la creación del primer *MAC Bridge PCD* y la creación de los *MAC Bridge PCD* asociados a los servicios. El comando es el siguiente:

- *mac-bridge-pcd create instance 2 bridge-id-ptr 1 port-num 2 tp-type mc-gem tp-ptr 2 port-priority 0 port-path-cost 1 port-spanning-tree-ind true encaps-method llc lanfcs-ind forward*

Tras la creación de los *MAC Bridge PCD* específicos de cada servicio, sería el turno del *GEM Port Network CTP* asociado al servicio multicast:

- *gem-port-network-ctp create instance 2 port-id 4094 t-cont-ptr 0 direction ani-to-uni traffic-mgmt-ptr-ustream 0 traffic-descriptor-profile-ptr 0 priority-queue-ptr-downstream 0 traffic-descriptor-profile-ds-ptr 0 enc-key-ring 0*

Este comando está asociado al *GEM Port* 4094, que es donde va el servicio multicast. Además, la dirección no es bidireccional como en los *GEM Port Network CTP* de servicios de internet, sino que es ani-to-uni. Esta entidad está asociada al último comando específico del servicio de internet y video, *Multicast GEM Interworking Termination Point*, a través del número de instancia, que estará en el *gem-port-nwk-ctp-conn-ptr* del siguiente comando:

- *multicast-gem-interworking-termination-point create instance 2 gem-port-nwk-ctp-conn-ptr 2 interwork-option mac-bridge service-prof-ptr 65535 interwork-tp-ptr 0 gal-prof-ptr 65535 gal-lpbk-config 0*

Este último comando se introduce después de *GEM Port Network CTP*. Con esto, y siguiendo el resto de los comandos tal cual el servicio de internet, se tendría un servicio de internet y video.

4.3.3 Implementación en Python de la configuración de servicios mediante CLI

4.3.3.1 Declaración del método y parámetros de entrada

La configuración de los servicios de internet y vídeo se realiza en un método desarrollado llamado *servicio_Internet_Video()*. Como se ha mencionado en el capítulo anterior, los parámetros de entrada a este método son:

1. La ID de la ONU/ONT asignada por el OLT y a la que se pretende suscribir un servicio
2. El número de serie de identificación de la ONU/ONT (llamado MAC en el código por su similitud con una dirección MAC)
3. El identificador del servicio (ID del servicio) que se pretende configurar en la ONU/ONT
4. El número de servicios que hay configurados en la ONU/ONT
5. El número de la interfaz WAN de la ONU/ONT (creada en el router) a la que se asociará el servicio

El método debe ser llamado desde un objeto de tipo *Servicio*, cuyos atributos serán los que se muestran en la Figura 2, y que se explican en el capítulo anterior. Dado que el método pertenece a ese objeto, los atributos del objeto pueden entrar “por defecto” en el método para que éste pueda trabajar con ellos. Para ello, la declaración del método debe incluir “*self*” entre los parámetros de entrada al método, como se muestra en la Figura 21.

```
def servicio_Internet_Video(self, ID_ONU, MAC_ONU, id_service, n_service, WANinstance):
```

Figura 21: Declaración del método `servicio_Internet_Video`

4.3.3.2 Lógica de implementación

En primer lugar, se crean unas listas vacías en las que se introducirán los diferentes parámetros discutidos a lo largo de este capítulo que sirven para configurar un servicio, y también se abre un fichero en el que se guardarán todos los comandos introducidos en el CLI del OLT correctamente al finalizar la ejecución del método con el objetivo de detectar fallos cuando los haya. En la Figura 22 se muestran las líneas de código.

```
# Creación de las listas en las que se almacenarán los parámetros
# internos de configuración
port_ID = []
alloc_ID = []
tcont_ID = []
num_instancia = []
puntero = []
ds_profile_index = []

# Creación de las listas en las que se almacenarán los parámetros
# de configuración correspondientes al servicio
VLAN_ID = []
VLAN_PRIORITY = []
BW_Downstream_GR = []
BW_Downstream_Excess = []
BW_Upstream_GR = []
BW_Upstream_BE = []
Type_Service = []

#Aquí se guarda el log
nombre_fichero = 'servicio_Internet_ONU_MAC_' + MAC_ONU + '.txt'
outfile = open(nombre_fichero, 'a')
```

Figura 22: Creación de listas de parámetros y fichero de log

A continuación, se utiliza el parámetro de entrada `n_service` para saber si ya hay servicios configurados en la ONU/ONT a la que se quiere suscribir a un servicio. De haber ya servicios configurados, es necesario hacer una conexión a la base de datos y cargar los parámetros de configuración de todos esos servicios. La razón para hacer esto es que la única forma de abonar la ONU/ONT a varios servicios de forma simultánea es introduciendo el comando perteneciente a cada servicio de forma iterativa. Es decir, si una ONU/ONT ya está asociada a uno o varios servicios y se quiere añadir otro, es necesario borrar la configuración perteneciente a los servicios anteriores, y después reconfigurar el OLT con esos servicios junto al nuevo servicio que se quiere añadir.

Todos los parámetros de servicios anteriores se sacan de la base de datos y se introducen en las listas que se han mostrado en la Figura 22. En la Figura 23 se muestra el código de la recuperación de datos de servicios anteriores.

```

if n_service>0:
    cnx = mysql.connector.connect(user='root', password='tfg_2017', host='127.0.0.1', database='gponServices')
    cursor = cnx.cursor()
    cursor.execute("select gDownstream, excessDownstream, gUpstream, excessUpstream, VLAN, \
                    VLANpriority, typeService from services where \
                    id_service in (select id_service from ont_service where id_ont='"+MAC_ONU+"' and configuration='CLI')")
    old_service_data=cursor.fetchall()
    i=0
    while i<n_service:
        BW_Downstream_GR.append(str(int(old_service_data[i][0])))
        BW_Downstream_Excess.append(str(int(old_service_data[i][1])))
        BW_Upstream_GR.append(str(int(old_service_data[i][2])))
        BW_Upstream_BE.append(str(int(float(old_service_data[i][2])+float(old_service_data[i][3])))
        VLAN_ID.append(str(old_service_data[i][4]))
        VLAN_PRIORITY.append(str(old_service_data[i][5]))
        Type_Service.append(old_service_data[i][6])
        i=i+1
    cursor.close()
    cnx.close()

```

Figura 23: Código de recuperación de datos de servicios anteriores

Una vez recuperados los datos de servicios anteriores, si es que los había, se introducen en las listas los datos del nuevo servicio que se pretende configurar. Esto se muestra en la Figura 24.

```

#Se añaden datos del nuevo servicio
VLAN_ID.append(str(self.VLAN))
VLAN_PRIORITY.append(str(self.VLANpriority))
BW_Downstream_GR.append(str(int(self.gDownstream)))
BW_Downstream_Excess.append(str(int(self.excessDownstream)))
BW_Upstream_GR.append(str(int(self.gUpstream)))
BW_Upstream_BE.append(str(int(float(self.gUpstream)+float(self.excessUpstream)))
Type_Service.append(self.typeService)

```

Figura 24: Introducción de los datos del nuevo servicio

Por último, antes de la introducción de los comandos, se asigna a cada uno de los servicios que se van a configurar números de GEM Port (*port_ID*), Alloc ID (*alloc_id*), instancia (*num_instancia*), T-Cont (*tcont_ID*) y puntero. Para ello, se usa un bucle con un número de iteraciones igual al número total de servicios que se van a configurar (*n_service* servicios anteriores + 1 del servicio nuevo) y se van asignando números de forma que no se solapen las instancias nunca. Esto se muestra en la Figura 25.

```

while i<(n_service+1):
    # Para que no se solapen puertos y allocs-ID, se asignan en función del identificador
    # de la ONU y del número de servicio en cuestión
    port_ID.append(600+100*ID_ONU+i)
    alloc_ID.append(600+100*ID_ONU+i)
    # Estos valores se asignan de este modo también para evitar solapamientos
    num_instancia.append(i+3)
    tcont_ID.append(i)
    puntero.append(32768+i)
    i=i+1

```

Figura 25: Asignación de números a parámetros de configuración del OLT para cada servicio

A continuación, hay que introducir los comandos de configuración a través del CLI. Para ello, se hace una conexión telnet al CLI del OLT y se introducen los comandos explicados en el apartado anterior de forma iterata, repitiendo cada comando específico de un servicio según el número de servicios que se quieren configurar. En la Figura 26 se muestra muestra la conexión telnet, así como la introducción de algunos comandos.

```

# Host y puerto al que se hace la conexión Telnet para acceder al CLI
host = "172.26.128.38"
port = "4551"

# Claves de acceso al CLI
password1 = "TLNT25"
password2 = "TLNT145"
enable = "enable"

# Acceso al CLI: conexión Telnet al host y puerto indicados anteriormente
tn = telnetlib.Telnet(host,port,1)
# Mediante la función write de telnetlib, escritura de los comandos que permiten
# acceder al menú de privilegios del CLI
tn.write(password1.encode('ascii') + b"\n")
time.sleep(0.1)
tn.write(enable.encode('ascii') + b"\n")
time.sleep(0.1)
tn.write(password2.encode('ascii') + b"\n")
time.sleep(0.1)

# Se pasa a String ID_ONU
ID_ONU = str(ID_ONU)

# A continuación, se definen todos los comandos necesarios para dar el servicio de Datos.
# Posteriormente serán ejecutados en el CLI con la función write.
# Primero se crea el canal OMCI de comunicación (con el mismo identificador que el de la ONU por convención)
# Se resetean las entidades MIB que pudiera haber y se activa fec en uplink (pasos opcionales)
inicio = "configure \n olt-device 0 \n olt-channel 0 \n onu-local " + ID_ONU + " \n omci-port " + ID_ONU +
" \n exit \n onu-omci " + ID_ONU + " \n ont-data mib-reset \n exit \n fec direction uplink " + ID_ONU + " \n onu-local " + ID_ONU + " \n"
tn.write(inicio.encode('ascii') + b"\n")
time.sleep(0.2)

# Se crean, dentro del menú de la ONU a configurar, tantos Alloc-ID como servicios
i=0
while i<(n_service+1):
    allocID = "alloc-id " + str(alloc_ID[i]).strip('[]') + " \n"
    tn.write(allocID.encode('ascii') + b"\n")
    time.sleep(0.2)
    i=i+1

```

Figura 26: Inicio de la introducción de comandos en el CLI del OLT

Como se puede ver en la Figura 26, la conexión telnet se realiza mediante el método *Telnet* de la librería *telnetlib*. Esta conexión telnet crea un objeto, *tn*, que representa la conexión telnet. A partir de este objeto, y llamando al método *write*, se van introduciendo todos los comandos codificados en ASCII (*American Standard Code for Information Interchange*).

Dado que la configuración de un servicio de internet y de un servicio de vídeo es similar salvo por unos comandos extras que se introducen en el caso de servicio de vídeo, si el nuevo servicio que se va a crear es de vídeo, se introducirán estos comandos extras como se muestra en la Figura 27.

```
if "Video" in Type_Service:
    # Creación del GEM Port Network CTP vinculado al servicio multicast. Está asociado al puerto
    # 4094 (en el que va el servicio multicast). Se diferencia de la entidad que forma el servicio Ethernet
    # en que direction ya no es de tipo bidireccional sino de tipo ani-to-uni. Asimismo, esta entidad está
    # asociada a la entidad Multicast GEM Interworking Termination Point: el nº de instancia debe coincidir con
    # el campo gem-port-nwk-ctp-conn-ptr de la otra entidad.
    gemport_multicast = "gem-port-network-ctp create instance 2 port-id 4094 t-cont-ptr 0 \
    direction ani-to-uni traffic-mgmt-ptr-ustream 0 traffic-descriptor-profile-ptr 0 \
    priority-queue-ptr-downstream 0 traffic-descriptor-profile-ds-ptr 0 enc-key-ring 0 \n"
    tn.write(gemport_multicast.encode('ascii') + b"\n")
    time.sleep(0.2)
```

Figura 27: Ejemplo de un comando que solo se introduce cuando se quiere configurar un servicio de vídeo

Una vez realizada la configuración, se actualiza la base de datos de servicios configurados en una ONT, tal como se muestra en la Figura 28.

```
##Meto en la base de datos la configuracion
cnx = mysql.connector.connect(user='root', password='tfg_2017', host='127.0.0.1', database='gponServices')
cursor = cnx.cursor()
query="insert into ont_service (id_ont, id_service, alloc_port, pointer, profile, \
instance, WANInstance, configuration) values ('"+str(MAC_ONU)+"', '"+str(id_service)+"', '"+str(port_ID[n_service])+
"', '"+str(puntero[n_service])+"', '"+str(ds_profile_index[n_service])+"', '"+str(num_instancia[n_service])+"', '"+str(WANInstance)+"', 'CLI')
cursor.execute(query)
cnx.commit()

cursor.close()
cnx.close()
```

Figura 28: Introducción del servicio configurado en la base de datos

4.4 Eliminación de servicios mediante CLI

4.4.1 Comandos CLI para eliminar servicios

Borrar un servicio consiste en eliminar todas las entidades que lo forman en el interior del OLT. En el menú *OLT Channel configuration* se quita la asociación *profile index-GEM Port*, y se elimina el *GEM Port* introduciendo los siguientes parámetros:

- *no policing downstream port-configuration entity port-id PORT_ID*
- *no port PORT_ID*

Tras esto, hay que ir al menú *ONU OMCI configuration* y eliminar todas las entidades GEM y VLAN según los siguientes comandos:

- *onu-omci ID_ONU*
- *gem-port-network-ctp delete instance NUM_INST*
- *gem-interworking-termination-point delete instance NUM_INST*
- *vlan-tagging-filter-data delete instance NUM_INST*
- *extended-vlan-tagging-operation-config-data delete instance 257*

Si se trata de un servicio de video, se deben eliminar también las instancias extras formadas en ese servicio introduciendo los comandos:

- *gem-port-network-ctp delete instance 2*
- *multicast-gem-interworking-termination-point delete instance 2*

Por último, se vuelve al menú *OLT Channel configuration* y se elimina el *profile index*; y, desde el menú *Local ONU configuration* se elimina el *alloc-id* utilizado para el servicio:

- *exit*
- *no policing downstream profile ds-profile-index DS_PROFILE_INDEX*
- *onu-local ID_ONU*
- *no alloc-id ALLOC_ID*

Y con esto, el servicio queda totalmente eliminado del OLT.

4.4.2 Implementación en Python de la eliminación de servicios

4.4.2.1 Declaración del método y parámetros de entrada

El borrado de los servicios de internet y vídeo se realiza en un método llamado *borrar_configuracion()*. Como se ha mencionado en el capítulo anterior, los parámetros de entrada a este método son:

1. La ID de la ONU/ONT asignada por el OLT y a la que se pretende suscribir un servicio

2. El número de serie de identificación de la ONU/ONT (llamado MAC en el código por su similitud con una dirección MAC)
3. El identificador del servicio (ID del servicio) que se pretende borrar de la ONU/ONT

A diferencia del método de configuración, este no requiere la creación de un objeto tipo servicio para ser invocado. Esto se debe a que este método saca de la base de datos de servicios configurados los parámetros que necesita, por lo que no es necesario pasar dichos parámetros como atributos de un objeto como en el caso anterior. En la Figura 29 se puede ver la llamada a este método.

```
Service.borrar_configuracion(int(id_onu),MAC[int(id_onu)],id_service)
```

Figura 29: Llamada al método de `borrar_configuracion()`²

4.4.2.2 Lógica de implementación

La implementación del método de borrado de servicios es más sencilla que la del método de suscripción pues, a diferencia de la suscripción, es posible borrar uno a uno los servicios de forma independiente y sin afectar al resto.

Al empezar a ejecutarse, se sacan los parámetros del servicio que se quiere borrar mediante una conexión a la base de datos, tal y como se muestra en la Figura 30. En la Figura 30 también se puede ver la introducción de los comandos CLI descritos en el apartado anterior, que se hace mediante las mismas librerías Python que en el caso de la configuración de servicios.

² Como se puede ver en la imagen, la llamada al método se hace directamente desde la clase *Service*, no desde un objeto de tipo *Service*

```
cnx = mysql.connector.connect(user='root', password='tfg_2017', host='127.0.0.1', database='gponServices')
cursor = cnx.cursor(buffered=True)
cursor.execute("select alloc_port, profile, instance from ont_service where id_ont='*MAC_ONU*' and id_service='*id_service*' and configuration='CLI'")
port_ID=cursor.fetchone()
cursor.execute("select typeService from services where id_service='*id_service*'")
typeService=cursor.fetchone()

inicio = "configure \n olt-device 0 \n olt-channel 0 \n"
tn.write(inicio.encode('ascii') + b"\n")
time.sleep(0.2)

#Borra asociacion profile-port
delprofileport = "no policing downstream port-configuration entity port-id " + str(port_ID[0]) + " \n"
tn.write(delprofileport.encode('ascii') + b"\n")
time.sleep(0.2)
```

Figura 30: Conexión a la base de datos e introducción de algunos comandos

Si el servicio que se quiere eliminar es un servicio de vídeo, se borrarán las instancias pertenecientes a este tipo de servicio, como se muestra en la Figura 31.

```
if typeService[0] == "Video":
    gemportvideo = "gem-port-network-ctp delete instance 2 \n"
    tn.write(gemportvideo.encode('ascii') + b"\n")
    time.sleep(0.2)

    multicast = "multicast-gem-interworking-termination-point delete instance 2 \n"
    tn.write(multicast.encode('ascii') + b"\n")
    time.sleep(0.2)
```

Figura 31: Borrado de instancias pertenecientes a servicio de vídeo

4.5 Conclusiones

Tal y como se ha descrito en este capítulo, se ha diseñado y desarrollado un programa en Python capaz de configurar servicios en las ONUs/ONTs del laboratorio mediante el modo de gestión CLI.

Aunque tiene sus desventajas en cuanto a complejidad, ofrece una buena aproximación al funcionamiento interno del OLT y además es posible introducir este tipo de configuración en diferentes tipos de programas y llevar a cabo una configuración de la red GPON de forma transparente al usuario final.

5

Configuración de servicios mediante OpenFlow

5.1 Introducción

En el capítulo anterior se describió cómo se configuraban servicios accediendo al CLI del OLT, y la complejidad que ello implica. Para reducir esta complejidad y ser capaces de configurar o modificar servicios de internet y vídeo de una forma más ágil y eficiente, se ha propuesto realizar las configuraciones mediante OpenFlow.

Usando CLI, si se quería reconfigurar el ancho de banda que ofrece un servicio, había que borrarlo y configurarlo de nuevo con el ancho de banda actualizado. OpenFlow, por otro lado, ofrece la posibilidad de modificar ese ancho de banda sin tocar la configuración del OLT, con un ahorro computacional y de tiempo muy importante.

Es importante señalar que, dado que no se va a tocar la configuración del OLT con OpenFlow, ya que los dispositivos PON actuales no soportan ningún estándar SDN, en realidad lo que se estará haciendo es configurar nuevos servicios sobre un servicio ya existente configurado mediante CLI o mediante TGMS, que son los dos tipos de gestión actualmente desarrollados por los fabricantes PON. No obstante, la posibilidad de poder modificar de una forma rápida y sencilla la configuración de servicios y perfiles de abonado mediante OpenFlow abren un mundo nuevo de posibilidades en el que, por ejemplo, el usuario podría ser capaz de modificar su ancho de banda en tiempo real y sin interferir con las delicadas configuraciones que se hacen en los OLTs.

5.2 Integración de OpenFlow en la red GPON

5.2.1 Conceptos básicos de OpenFlow

El estándar OpenFlow permite implementar SDN en cualquier tipo de red. Como se comentó en uno de los capítulos anteriores, las redes SDN se caracterizan por separar el plano de datos y el plano de control. Esta separación se traduce en la práctica a un controlador que actuará sobre diversos dispositivos de la red, permitiendo un control automatizado de ésta de forma rápida y simultánea. Dentro de este paradigma SDN, OpenFlow es un protocolo que permite la comunicación entre un controlador central OpenFlow (plano del control) y los dispositivos que controla (plano de datos), así como las diferentes funcionalidades que derivan esta comunicación.

Hay diferentes motivos por los que es interesante añadir una capa SDN a redes convencionales. Uno de los beneficios que ofrece SDN es el despliegue rápido y automatizado de nuevas aplicaciones. Esto es posible gracias a que, trabajando con el controlador, se puede trabajar con toda la red que este controla. Otra ventaja derivada de la arquitectura de este tipo de redes es una mayor flexibilidad y utilización de los recursos de la red, pues es posible hacer configuraciones capaces de modificar rutas de datos en función del volumen de tráfico que haya en la red en un momento dado para evitar problemas de congestión. Por último, otra ventaja de las redes SDN es su facilidad para aumentar el tamaño de la red añadiendo más controladores y dispositivos sin interferir con los que ya había configurados [7].

Los dispositivos de la red que se controlarán serán switches OpenFlow. Estos switches son capaces de operar con el tráfico de entrada y salida que pasa por ellos en función de las instrucciones que reciben del controlador OpenFlow. Para operar con el tráfico, los switches utilizan unas entidades enviadas por el controlador conocidas como tablas OpenFlow. En estas tablas habrá una serie de flujos OpenFlow con los que se comparará el tráfico de entrada o salida del switch. Cuando los paquetes cumplan con los requisitos fijados por el flujo, el switch OpenFlow se encargará de actuar con dicho tráfico en función de las órdenes del controlador. Una de estas funcionalidades es limitar la tasa del tráfico que pasa por un flujo, que es lo que interesa en este trabajo.

5.2.2 Integración de OpenFlow a la red GPON del laboratorio

En la Figura 32 se muestra el esquema básico de la red GPON del laboratorio sin ningún tipo de integración SDN, esto es, la configuración inicial de la red.

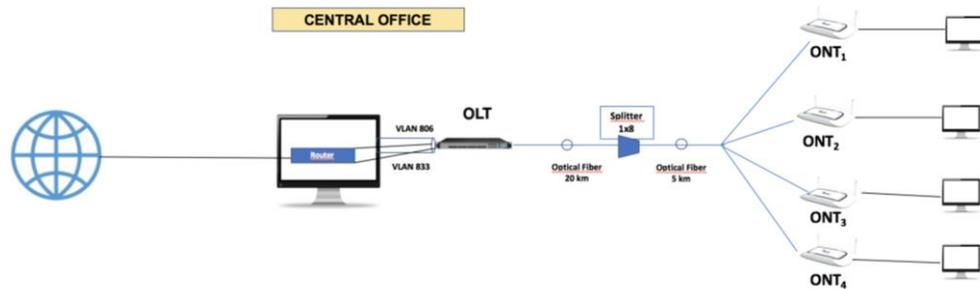


Figura 32: Red GPON sin OpenFlow

Como se puede observar, el camino *downstream* es muy sencillo: desde fuera de la red del laboratorio, los datos van directamente al ordenador central. Desde el ordenador central, se enrutan hacia el OLT añadiendo las VLANs encargadas de diferenciar servicios³ (806 vídeo, 833 internet). Por último, el OLT se conecta directamente mediante fibra óptica a las ONTs/ONUs a las que da servicio, y luego ya cada ONT/ONU permitirá el acceso a los servicios para los dispositivos que tenga conectados.

Para poder utilizar las herramientas descritas en el apartado anterior, necesitamos integrar a la red GPON convencional los siguientes elementos:

1. Añadir el controlador OpenFlow (OpenDayLight)
2. Añadir un switch en el OLT (simulando capa SDN OpenFlow) que limite el tráfico *downstream* de los servicios
3. Añadir un switch en la ONT/ONU (simulando capa SDN OpenFlow) que limite el tráfico *upstream* de los servicios

³ La diferenciación de servicios mediante VLANs es una práctica común en los ISP (*Internet Service Provider*)

4. Asegurar la comunicación entre el controlador OpenFlow y los switches OpenFlow a través de mensajes de dicho estándar

Dado que no se dispone de switches OpenFlow reales, se utilizará una aplicación llamada *Open Virtual Switch (OVS)* capaz de emular el funcionamiento de un switch OpenFlow. En la Figura 33 se muestra el nuevo escenario SDN-GPON (desarrollado en un TFG anterior [8]) donde se han introducido los nuevos componentes de la red para soportar OpenFlow. En primer lugar, el controlador irá en una máquina virtual en el interior del ordenador central. En el ordenador central también estará uno de los dos OVS, denominado COVS (*Central Open Virtual Switch*). Este switch se encargará de limitar el ancho de banda *downstream* del servicio que se quiera configurar, ya que limita todo el tráfico que viene desde fuera de la red de acceso. El otro OVS se denomina ROVS (*Remote Open Virtual Switch*) y será introducido en el interior de una raspberry entre la ONT/ONU y el resto de los dispositivos conectados a su red privada. El ROVS se encargará de limitar el ancho de banda *upstream* del servicio, pues limita todo lo que envían los dispositivos conectados a la ONT/ONU, es decir, limita el ancho de banda que envía el usuario final.

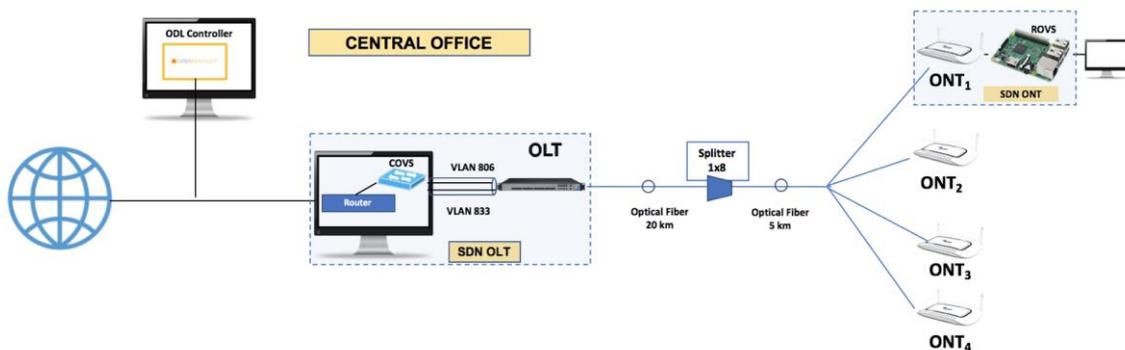


Figura 33: Integración de un escenario OpenFlow en la red GPON (SDN-GPON)

Con este nuevo escenario de red, se tendrán que configurar los enrutamientos de forma adecuada para asegurar la comunicación entre los switches virtuales (ROVS, COVS) y el controlador OpenDayLight (ODL). Para ello, se configurarán en los switches el controlador ODL como controlador principal para que se comuniquen con él al ser lanzados. Cabe destacar que todo este proceso de configuración y gestión de la comunicación entre el controlador y los OVS's fue realizada en un Trabajo Fin de Grado

anterior [8]. Una vez configurada la red GPON-SDN y comprobado que todos los dispositivos se conectan correctamente mediante OpenFlow, el siguiente paso es configurar los flujos OpenFlow que harán posible la configuración de los diferentes servicios asociados a las ONTs/ONUs así como la limitación del tráfico de dichos servicios en los canales de subida y bajada.

5.3 Flujos OpenFlow

Los flujos OpenFlow se encargarán de crear servicios y de limitar el ancho de banda a las tasas que requiera el nuevo servicio. Para ello, harán uso de los medidores o *meters* OpenFlow, que limitan el tráfico que pasa por un flujo OpenFlow a una tasa determinada cuando la tasa de tráfico que atraviesa ese flujo supera cierto umbral.

5.3.1 Configuración de flujos OpenFlow

Los flujos OpenFlow tienen varios campos, mostrados en la Figura 34. De todos ellos, el más importante es “*Match Fields*”.

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags
--------------	----------	----------	--------------	----------	--------	-------

Figura 34: Campos de un flujo OpenFlow

El campo *Match Fields* define qué características deben de tener los paquetes que entran en el switch para pasar por el flujo OpenFlow. Estas características pueden ser muy variadas: dirección MAC de origen o destino, IP de origen o destino y VLAN, entre otros. Además, puede ser una combinación de varias características en concreto.

En nuestro caso, para limitar el ancho de banda de un servicio en la dirección *downstream*, el flujo que se configurará en el COVS deberá clasificar todo el tráfico con una dirección MAC de destino igual a la dirección MAC de la interfaz WAN de la ONT/ONU en la que se quiere configurar el servicio. Por otro lado, el flujo encargado de limitar el ancho de banda de un servicio en el canal de subida *upstream* en el interior del ROVS deberá clasificar todo el tráfico de salida con una dirección MAC de destino igual a la dirección MAC de la interfaz interna de la ONT/ONU a la que envían los datos todos los dispositivos conectados a dicha ONT/ONU, esto es, datos de salida de la casa del usuario.

El campo *priority* define la prioridad del flujo OpenFlow cuando pasan paquetes que cumplen requisitos para pasar por varios flujos. Se ha decidido que los flujos OpenFlow de esta aplicación funcionen siempre con la misma prioridad (8000), pero es posible definir varias prioridades en función del hipotético modelo de negocio asociado a la aplicación (servicios premium temporales, por ejemplo). En este punto, cabe indicar que en un OVS solamente funcionará el flujo con mayor prioridad.

Por otro lado, el campo *instructions* es muy importante, pues dicta qué se hace con el tráfico que pasa por el flujo OpenFlow. Las instrucciones disponibles son muy variadas: enviar el tráfico a otra tabla OpenFlow, descartar el tráfico o enviar el tráfico por un puerto determinado del switch, entre otras opciones. También es posible realizar varias instrucciones en un orden determinado. Además, una instrucción puede contener en su interior varias acciones.

Para el caso que nos ocupa, se deben hacer dos cosas: que el tráfico atraviese el switch como si fuese un switch normal y corriente, y que limite el ancho de banda utilizando los medidores. Esto se consigue añadiendo dos instrucciones: la primera (*order=1*) será del tipo *apply-actions*, con una acción de salida de los paquetes (*output-action*) configurada para que funcione como un switch normal ("*output-node-connector*": "*NORMAL*"). Esta acción se encargará de que el tráfico que entre en el flujo salga del router por todos sus puertos de salida, emulando el funcionamiento básico de un switch y permitiendo que el tráfico pase por el OVS sin problemas. La segunda instrucción (*order > 1*) será la encargada de accionar el medidor para limitar el tráfico. En ella simplemente se indica el *meter-id* del medidor que el flujo tiene que utilizar. El funcionamiento y configuración de los medidores asociados a un flujo concreto para limitar el ancho de banda máximo se explicarán con detalle en el apartado siguiente.

El resto de los campos de los flujos OpenFlow no son tan relevantes, aunque merece la pena mencionar que el campo *timeouts* puede contener un tiempo limitado en el que un flujo concreto estará operativo, y puede servir para configurar servicios con un límite de duración, esto es, servicios Premium activos durante un cierto tiempo.

5.3.2 Configuración de los medidores OpenFlow

Los medidores OpenFlow miden el tráfico que pasa por un flujo OpenFlow con el objetivo de limitarlo a una tasa fija si supera cierto umbral. Los principales campos de un medidor OpenFlow son:

1. **Meter-id:** identificador del medidor. Un flujo indica qué medidor va a utilizar utilizando este número.
2. **Flags:** indica las unidades de bits en los que se medirá y limitará el tráfico. En este caso, se usará en kbps (kilobits por segundo).
3. **Meter-band-headers:** en este campo se configura la funcionalidad principal del medidor. Para limitar el ancho de banda, es necesario seleccionar “*ofpmbt-drop*”. Por su parte, en “*band-rate*” se selecciona el umbral de ancho de banda en el que se activará el medidor, y en “*drop-rate*” la tasa a la que se limitará el ancho de banda. Para la funcionalidad que se quiere implementar, ahí en ambos campos se colocará el ancho de banda máximo del servicio al que irá asociado el medidor configurado.

5.4 Implementación en Python de la configuración de servicios mediante OpenFlow

5.4.1 Métodos de configuración de flujos OpenFlow y medidores

Los flujos y medidores OpenFlow se configuran en el controlador ODL, y posteriormente el controlador se encarga de enviarlos al switch OpenFlow para que se pongan en funcionamiento. La configuración se puede realizar de varias maneras en función de las necesidades del usuario. La primera de ellas, y más directa, es el uso de la interfaz web proporcionada por el controlador ODL, desde la que es posible introducir configuraciones de flujos y medidores rellenando diferentes campos. Un ejemplo de esta interfaz se muestra en la Figura 35.

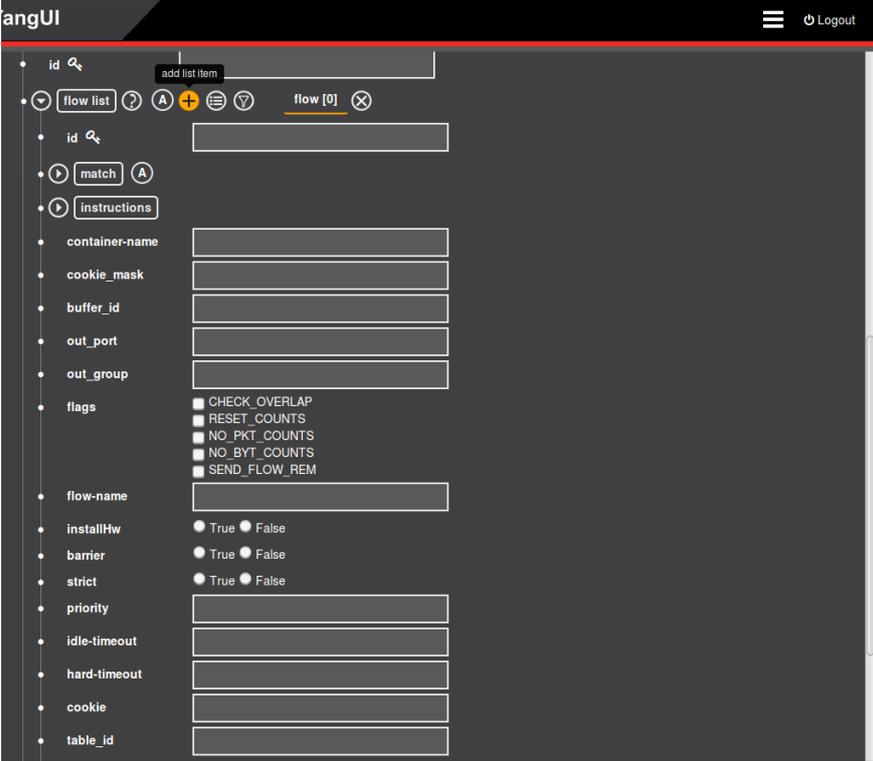
The image shows a web interface for configuring an OpenFlow flow. The interface is titled 'angUI' and has a 'Logout' button in the top right corner. It features a search bar for 'id' and a 'flow list' section with a '+ add list item' button. The main configuration area is for a 'flow [0]' and includes the following fields: 'id' (text input), 'match' (button), 'instructions' (button), 'container-name' (text input), 'cookie_mask' (text input), 'buffer_id' (text input), 'out_port' (text input), 'out_group' (text input), 'flags' (checkboxes for CHECK_OVERLAP, RESET_COUNTS, NO_PKT_COUNTS, NO_BYT_COUNTS, SEND_FLOW_REM), 'flow-name' (text input), 'installHw' (radio buttons for True/False), 'barrier' (radio buttons for True/False), 'strict' (radio buttons for True/False), 'priority' (text input), 'idle-timeout' (text input), 'hard-timeout' (text input), 'cookie' (text input), and 'table_id' (text input).

Figura 35: Campos de configuración de un flujo OpenFlow en la interfaz web del controlador ODL

Para llevar a cabo la configuración de un flujo bastaría con rellenar los campos imprescindibles para el funcionamiento del flujo, cómo su identificador (*id*) o su prioridad (*priority*).

La otra forma de configurar flujos y medidores OpenFlow es enviando las configuraciones en formato JSON al controlador mediante métodos HTTP PUT, como se mostró en la Figura 17. Esta segunda forma es la más adecuada para automatizar las configuraciones de los flujos, y es la que se ha utilizado en esta aplicación.

5.4.2 Implementación en Python de la configuración de servicios

5.4.2.1 Declaración del método y parámetros de entrada

La configuración de los servicios mediante OpenFlow se realiza en un método llamado *servicio_Internet_OpenFlow()*. Los parámetros de entrada a este método son:

1. La ID de la ONU/ONT asignada por el OLT y a la que se pretende suscribir un servicio
2. El número de serie de identificación de la ONU/ONT (llamado MAC en el código por su similitud con una dirección MAC)
3. El identificador del servicio (ID del servicio) que se pretende configurar en la ONU/ONT

El método debe ser llamado desde un objeto de tipo *Servicio* y sus atributos entran en este método de la misma forma que ocurría con su homólogo mediante CLI. En la Figura 36 se muestra la declaración del método. El borrado de servicios es exactamente igual.

```
def servicio_Internet_OpenFlow(self, ID_ONU, MAC_ONU, id_service):
```

Figura 36: Declaración del método de configuración mediante OpenFlow

5.4.2.2 Lógica de implementación

5.4.2.2.1 Creación de un servicio

El código de este método es mucho más sencillo que el de la configuración mediante CLI. El objetivo de este método es crear dos flujos OpenFlow y sus correspondientes medidores para limitar el ancho de banda downstream (COVS), y el ancho de banda upstream (ROVS).

En primer lugar, hay que saber qué ROVS hay que utilizar en función de la ONT/ONU en la que se configurará el servicio (*ovs_upstream*), junto a la dirección MAC de destino para el flujo del canal *upstream* (*mac_upstream*). Para ello, simplemente se ha utilizado un *if* para seleccionar el identificador de ROVS y la MAC destino en función del número de serie de identificación de la ONT/ONU (*MAC_ONU*), tal y como se muestra en la Figura 37.

```
elif MAC_ONU == "54-4c-52-49-5b-01-f7-30":  
    ovs_upstream="346653127080" #Id del OVS para el upstream en los servicios del ONT f7-30  
    mac_upstream="78:3d:5b:01:f7:30"  
    ipQuery="192.168.0.104"
```

Figura 37: Selección del ID de OVS y MAC destino en función de MAC_ONU

Por otro lado, también hay que saber la dirección MAC de destino que se configurará en el flujo del COVS. En la Figura 37 se puede observar una variable llamada “*ipQuery*”. Como se tiene un servidor DHCP que da IPs a las interfaces WAN externas de las ONTs/ONUs, es posible saber de antemano qué IP externa tiene la interfaz de la ONT/ONU dedicada al servicio que se quiere configurar. A través de esta IP, mediante el comando “*arp -a*”, es posible sacar la dirección MAC de destino para el flujo del COVS. El código de esto se muestra en la Figura 38.

```
proc = subprocess.Popen('arp -a', stdout=subprocess.PIPE, shell=True)
(out, err) = proc.communicate()
out=out.decode()

posicion=out.find(ipQuery)
mac_downstream=out[(posicion+18):(posicion+35)]
```

Figura 38: Dirección MAC de destino para los flujos del COVS

El siguiente paso es enviar los mensajes HTTP para introducir los flujos y medidores en el controlador OpenFlow. Para ello, primero se define la cabecera HTTP junto a la URL a la que se enviarán los mensajes. En la Figura 39 es posible ver estas declaraciones.

```
head = {'Content-type': 'application/yang.data+json', 'Accept': 'application/json, text/plain'}

urlDownstream="http://10.0.103.45:8181/restconf/config/opendaylight-inventory:nodes/node \
/openflow:159303472399026/flow-node-inventory:table/0/flow/flow"+str(40+int(id_service))
urlUpstream="http://10.0.103.45:8181/restconf/config/opendaylight-inventory:nodes/node \
/openflow:"+ovs_upstream+"/flow-node-inventory:table/0/flow/flow"+str(40+int(id_service))
urlMeterDown="http://10.0.103.45:8181/restconf/config/opendaylight-inventory:nodes/node \
/openflow:159303472399026/flow-node-inventory:meter/"+str(int(int(self.gDownstream)/1024))
urlMeterUp="http://10.0.103.45:8181/restconf/config/opendaylight-inventory:nodes/node \
/openflow:"+ovs_upstream+"/flow-node-inventory:meter/"+str(int(self.gUpstream))
```

Figura 39: Cabecera HTTP junto a URLs a las que se enviarán los datos

Por último, hay que definir el cuerpo del mensaje HTTP y enviarlo al controlador. Las Figuras 40 y 41 muestran un flujo y el medidor asociado con un ancho de banda máximo para el canal de subida (*upstream*), respectivamente. Se puede observar que aparecen todos los parámetros nombrados en las secciones de configuración de flujos y medidores. Para el caso de la configuración del flujo para configurar el servicio en el canal de bajada *downstream* el código es muy similar.

```

flowUp='' '{
  "flow": [
    {
      "id": "flow'+str(40+int(id_service))+'",
      "match": {
        "ethernet-match": {
          "ethernet-destination": {
            "address": "''+mac_upstream+'''"
          }
        }
      },
      "instructions": {
        "instruction": [
          {
            "order": 5,
            "meter": {
              "meter-id": "''+str(int(self.gUpstream))+'''"
            }
          },
          {
            "order": "1",
            "apply-actions": {
              "action": [
                {
                  "order": "1",
                  "output-action": {
                    "output-node-connector": "NORMAL"
                  }
                }
              ]
            }
          }
        ]
      },
      "flow-name": "flow'+str(40+int(id_service))+'",
      "priority": "8000",
      "idle-timeout": "0",
      "hard-timeout": "0",
      "cookie": "478478457845784600",
      "table_id": "0"
    }
  ]
}'' ]
}'''

```

Figura 40: Flujo de upstream

```

meterUp='' '{
  "meter": [
    {
      "flags": "meter-kbps",
      "meter-id": "''+str(int(self.gUpstream))+'''",
      "meter-name": "mymeter",
      "container-name": "mymeter",
      "meter-band-headers": {
        "meter-band-header": [
          {
            "band-id": "0",
            "meter-band-types": {
              "flags": "ofpmbt-drop"
            },
            "band-rate": "''+str(int(self.gUpstream)*1024)+'''",
            "band-burst-size": "0",
            "drop-rate": "''+str(int(self.gUpstream)*1024)+'''",
            "drop-burst-size": "0"
          }
        ]
      }
    }
  ]
}'' ]
}'''

```

Figura 41: Medidor para upstream

Por último, utilizando la librería *requests* [9], se mandan los mensajes HTTP utilizando el método PUT. Esto se muestra en la Figura 42.

```
requests.put(urlMeterDown,auth=HTTPBasicAuth('admin', 'admin'),headers=head,data=meterDown)
requests.put(urlDownstream,auth=HTTPBasicAuth('admin', 'admin'),headers=head,data=flowDown)
requests.put(urlMeterUp,auth=HTTPBasicAuth('admin', 'admin'),headers=head,data=meterUp)
requests.put(urlUpstream,auth=HTTPBasicAuth('admin', 'admin'),headers=head,data=flowUp)
```

Figura 42: Inserción de flujos y medidores en el controlador mediante petición HTTP PUT

Además, se actualiza la base de datos para indicar la creación del nuevo servicio. Esta configuración requiere muchos menos campos, tal y como se observa en la Figura 43.

```
cnx = mysql.connector.connect(user='root', password='tfg_2017',host='127.0.0.1',database='gponServices')
cursor = cnx.cursor()
cursor.execute("insert into ont_service (id_ont, id_service, configuration) values ('"+str(MAC_ONU)+"', '"+str(id_service)+"', 'OpenFlow')")
cnx.commit()
cursor.close()
cnx.close()
```

Figura 43: Inserción del servicio configurado en la base de datos

5.4.2.2 Borrado de un servicio

El borrado de un servicio mediante OpenFlow es totalmente trivial, pues simplemente hay que hacer una petición HTTP DELETE a las mismas URLs que en el caso de la configuración, y actualizar la base de datos. En la Figura 44 se muestra el código principal.

```
head = {'Content-type': 'application/yang.data+json', 'Accept': 'application/json, text/plain'}
urlDownstream="http://10.0.103.45:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:159303472399026/flow-node-inventory:table/0/flow/flow-"+str(40+int(id_service))
urlUpstream="http://10.0.103.45:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:"ovs_upstream"/flow-node-inventory:table/0/flow/flow-"+str(40+int(id_service))
urlMeterDown="http://10.0.103.45:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:159303472399026/flow-node-inventory:meter/"+str(int(int(self.gDownstream)/1024))
urlMeterUp="http://10.0.103.45:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:"ovs_upstream"/flow-node-inventory:meter/"+str(int(self.gUpstream))
requests.delete(urlMeterDown,auth=HTTPBasicAuth('admin', 'admin'))
requests.delete(urlDownstream,auth=HTTPBasicAuth('admin', 'admin'))
requests.delete(urlMeterUp,auth=HTTPBasicAuth('admin', 'admin'))
requests.delete(urlUpstream,auth=HTTPBasicAuth('admin', 'admin'))
cnx = mysql.connector.connect(user='root', password='tfg_2017',host='127.0.0.1',database='gponServices')
cursor = cnx.cursor()
cursor.execute("delete from ont_service where id_ont='"+MAC_ONU+"' and id_service='"+id_service+"' and configuration = 'OpenFlow'")
cnx.commit()
cursor.close()
cnx.close()
print("Se ha configurado de la ONU con MAC " + MAC_ONU + " y el servicio '"+id_service+"' ha sido borrado.\n")
input("Press enter to continue...")
```

Figura 44: Código principal del borrado de un servicio OpenFlow

5.5 Conclusiones

Aunque la configuración mediante OpenFlow siga siendo dependiente de una configuración anterior mediante CLI o TGMS, es muy interesante ser capaces de modificar anchos de banda de un servicio en tiempo real de una forma tan sencilla como es enviar peticiones HTTP al controlador, sin tocar el OLT. Esta es una muestra más del poderío que pueden tener las redes SDN de cara a un futuro en el que la flexibilidad y

velocidad de configuración serán clave debido al aumento de dispositivos que está habiendo desde hace unos años.

Por lo tanto, en este capítulo se ha descrito la propuesta e implementación de una solución SDN en redes GPON (SDN-GPON) a partir de la integración del estándar OpenFlow, siendo capaces de emular una capa SDN tanto en el OLT como en la ONTs/ONUs con la finalidad de gestionar y configurar servicios en tiempo real a diferentes usuarios de un modo flexible, rápido y eficiente a través de dicho estándar.

6

Conclusiones y Líneas Futuras

6.1 Conclusiones

En este Trabajo de Fin de Máster se ha llevado a cabo el desarrollo de una aplicación capaz de configurar una red GPON utilizando dos métodos diferentes: de la forma clásica mediante una configuración de comandos en el CLI del OLT, y de una forma novedosa utilizando el estándar OpenFlow.

Para llevar a cabo el trabajo, se hizo un esquema de cómo y qué tenía que hacer la aplicación para poder integrar ambos modos de configuración sin que hubiera interferencias entre ellos. Una vez hecho el esquema y elegido el lenguaje de programación, se llevó a cabo la implementación en Python.

La configuración mediante CLI es lo más parecido a lo que hacen los proveedores de servicio en la actualidad. Sin embargo, tiene claras limitaciones en cuanto a complejidad, tiempo y flexibilidad. En este sentido, actualmente las redes controladas por software SDN surgen como una posibilidad muy interesante a medio plazo, por lo que parte de este Trabajo Fin de Máster se ha centrado en continuar con el planteamiento e implementación de una solución SDN en una red real GPON convencional añadiendo una capa OpenFlow que permita configuraciones dinámicas de servicios y perfiles de abonado de forma transparente. Con la aplicación terminada, se ha comprobado que la integración de OpenFlow en una red GPON para modificar los servicios configurados en las ONTs/ONUs puede ser llevada a cabo de un modo eficiente.

Sin embargo, la integración de OpenFlow en una red GPON convencional ha resultado tediosa. Aunque funcione bien cuando la red es estable, los OVS en la

actualidad dan muchos problemas, sobretodo cuando se pretende trabajar con varios conectados a un mismo controlador debido a que están en una etapa de desarrollo. En este sentido, aún hace falta mucho trabajo para que sean una opción viable y estable.

Por último, para llevar a cabo este proyecto han sido de gran ayuda los trabajos anteriores de Carlos Marcos Miguel [10] y de Javier Néstor Azofra Ovejero [8], que desarrollan cada tipo de configuración de servicios en la red GPON de forma independiente y que son la base sobre la que se ha desarrollado la aplicación de este Trabajo de Fin de Máster.

6.2 Líneas Futuras

Uno de los siguientes pasos que se pueden dar con la red del laboratorio es la configuración interna a nivel CLI de las ONTs/ONUs con el objetivo de automatizar aún más la configuración de servicios en el laboratorio.

Por otro lado, TELNET Redes Inteligentes está desarrollando una API REST con el TGMS para poder integrar TGMS con otros programas. En el futuro, puede ser interesante sustituir la configuración mediante CLI por configuración mediante TGMS. La principal ventaja de este cambio sería ganar en comodidad y ahorrarse problemas de configuración de servicios derivados de la complejidad del modo CLI.

También sería interesante crear un algoritmo integrado de forma transparente en el programa de Python capaz de modificar dinámicamente el ancho de banda de cada usuario y/o servicio en función del ancho de banda disponible y el uso real que le estén dando los usuarios. De este modo, se podría aprovechar que un usuario no está utilizando todo lo que ha contratado para ceder el ancho de banda sobrante a otro usuario que lo necesite más durante un tiempo limitado, dando lugar a redes y servicios más aprovechados.

Por último, sería interesante probar cómo funciona el programa con unos nuevos modelos de ONTs/ONUs de nivel 2 y nivel 3 que se acaban de adquirir.

7 Bibliografía

«Página web de TELNET Redes Inteligentes,» [En línea]. Available:
1] <http://www.telnet-ri.es/>.

«Antigua página web de OpenFlow,» [En línea]. Available:
2] <http://archive.openflow.org/wp/learnmore/>.

«Web oficial de Python,» [En línea]. Available: <https://www.python.org/>.
3]

«Web oficial de Open Networking,» [En línea]. Available:
4] <https://www.opennetworking.org/sdn-definition/>.

«Web oficial de Open Virtual Switch,» [En línea]. Available:
5] <https://www.openvswitch.org/>.

Á. G. Aguado, «Diseño de un agente OpenFlow en una maqueta Red de
6] Acceso Óptica real,» Valladolid, Valladolid, España, 2017.

«Web de CISCO, Software-Defined Networking,» [En línea]. Available:
7] <https://www.cisco.com/c/en/us/solutions/software-defined-networking/overview.html>.

J. N. A. Ovejero, «Evolución en la integración del estándar OpenFlow en una
8] maqueta de red GPON,» Valladolid, Valladolid, España, 2018.

«Sitio web de la librería Python Requests,» [En línea]. Available:

9] <http://docs.python-requests.org/en/master/>.

C. M. Miguel, «Configuración de Servicios por un Proveedor en una
10] maqueta Red de Acceso Óptica real,» Valladolid, Valladolid, España.

«GPON: VLANs and GEM Ports,» [En línea]. Available:
11] <https://sites.google.com/site/amitsciscozone/home/gpon/gpon-vlans-and-gem-ports>.