



UNIVERSIDAD DE

VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

## TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

# **Configuración de Servicios por un Proveedor en una maqueta Red de Acceso Óptica real**

Autor:

**Don Carlos Marcos Miguel**

Tutor:

**Doña Noemí Merayo Álvarez**

---

**TÍTULO: Configuración de Servicios por un Proveedor en una maqueta  
Red de Acceso Óptica real**

**AUTOR: Don Carlos Marcos Miguel**

**TUTOR: Doña Noemí Merayo Álvarez**

**DEPARTAMENTO: Teoría de la Señal y Comunicaciones e Ingeniería  
Telemática**

---

### **TRIBUNAL**

---

**PRESIDENTE: Don Ignacio de Miguel Jiménez**

**VOCAL: Don Juan Carlos Aguado Manzano**

**SECRETARIO: Doña Noemí Merayo Álvarez**

**SUPLENTE: Don Ramón J. Durán Barroso**

**SUPLENTE: Don Rubén M. Lorenzo Toledo**

---

---

**FECHA: 19 de Junio de 2017**

**CALIFICACIÓN:**

---

### **Resumen de TFG**

El estudio realizado en este Trabajo de Fin de Grado se basa en la configuración de la red de acceso GPON situada en el laboratorio de Comunicaciones Ópticas de la Escuela Técnica Superior de Ingenieros de Telecomunicación.

En primer lugar, se analizó a nivel físico la estructura de la red y sus elementos principales, el OLT (*Optical Line Termination*) y las ONUs/ONTs (*Optical Network Unit/Terminal*). Se realizaron pruebas en los dos canales de comunicación (canal de bajada -*Downstream*- y canal de subida - *Upstream*), para lo cual se utilizaron

dispositivos como el OTDR (*Optical Time Domain Reflectometer*), la fuente de luz y el medidor de potencia óptica.

Tras esto, se llevó a cabo la configuración de servicios, tanto de Internet como de vídeo, utilizando los dos modos de gestión de red desarrollados por la empresa TELNET Redes Inteligentes. Así pues, el TGMS (*TELNET GPON Management System*), es un sistema de gestión web muy sencillo que abstrae de la complejidad del protocolo GPON y el CLI (*Command Line Interface*), una interfaz más compleja basada en comandos relacionados con el protocolo GPON. Finalmente, se realizaron una serie de programas en lenguaje de programación Python, junto con XML, para automatizar el control de la red y facilitar su gestión.

### **Palabras clave**

GPON (red óptica pasiva con capacidad de gigabit), OLT (terminación óptica de línea), ONU (unidad de red óptica), TGMS (sistema de gestión GPON de TELNET), servicio de Internet, servicio de vídeo, CLI (interfaz de línea de comandos), Python.

### **Abstract**

The research developed in this project is based on the configuration of a real GPON testbed located at the Optical Communications Laboratory of the School of Telecommunication Engineers.

First of all, we carried out a physical analysis focus on the network structure and its main components, the OLT (Optical Line Termination) and the ONUs/ONTs (Optical Network Unit/Terminal). We did different tests at both communication channels, that is, the downstream and the upstream channels. For this analysis, we used devices such as the OTDR (Optical Time Domain Reflectometer), a PON light source and an optical power meter.

After that, we configured the services on the GPON (Internet and video services) using two management modes developed by the enterprise *TELNET Redes Inteligentes*. The TGMS (TELNET GPON Management System) is a very simple web system which abstracts the configuration from the complexity of the GPON protocol and CLI (Command Line Interface), a more complex interface based on commands and very

closed to the GPON protocol. Finally, we developed several programs using the languages Python and XML in order to automate the network control and make easier its management and configuration.

## **Keywords**

GPON (*Gigabit-capable Passive Optical Network*), OLT (*Optical Line Termination*), ONU (*Optical Network Unit*), TGMS (*TELNET GPON Management System*), Internet service, video service, CLI (*Command Line Interface*), Python.

## **Agradecimientos**

Me gustaría aprovechar esta sección para agradecer a todas las personas que me han apoyado y ayudado a lo largo de estos años de carrera y con las que he compartido mi paso por la universidad.

En primer lugar, a mis padres, Jesús y Pilar, por su cariño y su apoyo en todo momento así como por inculcarme el valor del esfuerzo. También me gustaría dar las gracias al resto de mis familiares, en especial a mi abuela Eugenia. Asimismo, quiero acordarme de mis compañeros de clase y amigos, con los que he compartido grandes momentos y que han hecho que esta época universitaria haya sido única.

Finalmente, me gustaría dar las gracias a mi tutora Noemí, por la oportunidad de realizar este proyecto y por su apoyo y ayuda así como al grupo de Comunicaciones Ópticas. También me gustaría mostrar mi agradecimiento a Carlos Alonso y Carlos Brizuela así como a los trabajadores de la empresa TELNET Redes Inteligentes por su inestimable ayuda.



## ÍNDICE

<b>1</b>	<b>Introducción .....</b>
1.1	Motivación .....
1.2	Objetivos .....
1.2.1	Objetivo General .....
1.2.2	Objetivos Específicos .....
1.3	Fases y Métodos .....
1.3.1	Fase de Análisis .....
1.3.2	Fase de Configuración .....
1.3.3	Fase de Programación .....
1.3.4	Fase de Realización de los Informes .....
1.4	Estructura de la Memoria del TFG .....
<b>2</b>	<b>Análisis de la red GPON a nivel físico .....</b>
2.1	Introducción .....
2.2	Montaje real de la red GPON en el laboratorio .....
2.3	Análisis de potencias y pruebas con OTDR en sentido Downstream .....
2.4	Análisis de potencias y pruebas con OTDR en sentido Upstream .....
2.5	Conclusiones .....
<b>3</b>	<b>Configuración de servicios en la red GPON utilizando el modo de gestión TGMS .....</b>
3.1	Introducción .....
3.2	Interfaz TGMS: Telnet GPON Management System .....
3.2.1	Menú Devices .....
3.2.2	Menú Profiles .....
3.2.3	Menú Subscribers .....
3.2.4	Menú Alarms .....
3.2.5	Menú Log .....

3.2.6	Menú Configuration.....	
3.3	Configuración del servicio de Internet .....	
3.3.1	Configuración del servicio de Internet en la interfaz del TGMS.....	
3.3.2	Configuración del servicio de Internet en los routers de las ONUs .....	
3.3.3	Pruebas y resultados de la configuración de un servicio de Internet.....	
3.4	Configuración del servicio de Internet y Vídeo.....	
3.4.1	Configuración del servicio de Internet y vídeo en la interfaz del TGMS.....	
3.4.2	Configuración del servicio de Internet y vídeo en los routers de las ONUs.....	
3.4.3	Pruebas y resultados de la configuración de un servicio de Internet y vídeo ...	
3.5	Conclusiones.....	
<b>4</b>	<b>Configuración de servicios en la red GPON utilizando el modo de gestión CLI.....</b>	
4.1	Introducción.....	
4.2	Descripción del interfaz CLI .....	
4.3	Configuración del servicio de Internet .....	
4.3.1	Configuración del servicio de Internet en los routers de las ONUs .....	
4.3.2	Configuración del servicio de Internet en la interfaz del CLI .....	
4.3.3	Pruebas y resultados de la configuración de un servicio de Internet.....	
4.4	Configuración del servicio de Internet y Vídeo.....	
4.4.1	Configuración del servicio de Internet y vídeo en los routers de las ONUs.....	
4.4.2	Configuración del servicio de Internet y vídeo en la interfaz del CLI .....	
4.4.3	Pruebas y resultados de la configuración de un servicio de Internet y vídeo ...	
4.5	Conclusiones.....	
<b>5</b>	<b>Gestión de la red GPON utilizando Python y XML.....</b>	
5.1	Introducción.....	
5.2	Acceso al CLI desde Python.....	

5.3	Programa de configuración general de la red GPON.....	
5.3.1	Estructura general del programa .....	
5.3.2	Configuración del servicio de Internet .....	
5.3.3	Configuración del servicio de Internet y Vídeo .....	
5.3.4	Ver configuración.....	
5.3.5	Borrar configuración .....	
5.3.6	Modificar configuración .....	
5.3.7	Cargar configuración utilizando XML .....	
5.4	Otros scripts para la gestión de la red GPON.....	
5.4.1	Acceso al CLI.....	
5.4.2	Activación de los modos de gestión TGMS / CLI.....	
5.4.3	Mostrar modo de gestión activo .....	
5.4.4	Mostrar ONUs conectadas .....	
5.5	Conclusiones .....	
<b>6</b>	<b>Conclusiones y Líneas Futuras.....</b>	
6.1	Conclusiones .....	
6.2	Líneas Futuras .....	
<b>7</b>	<b>Bibliografía.....</b>	

## **ÍNDICE DE FIGURAS**

Figura 1. Vista general y por partes de la red de acceso GPON.....	9
Figura 2. Esquema de la red de acceso GPON. Topología en árbol.....	10
Figura 3. Apariencia del modelo de OLT <i>SmartOLT 350</i> .....	11
Figura 4. Esquema frontal del <i>SmartOLT 350</i> .....	12
Figura 5. Apariencia del modelo de ONU/ONT <i>WaveAccess 3021</i> .....	13
Figura 6. Vista frontal de las ONUs/ONTs en la red de acceso GPON del laboratorio. ..	13
Figura 7. Bobinas de fibra óptica en el montaje GPON realizado en el laboratorio. ....	14
Figura 8. Divisores ópticos o <i>splitters</i> . ....	15
Figura 9. Caja de empalmes (parte superior) y bobina de distribución (parte inferior). ..	15
Figura 10. Esquema detallado de la red GPON. ....	16
Figura 11. Configuración de la caja de empalmes.....	17
Figura 12. Longitudes de onda utilizadas en GPON. ....	18
Figura 13. OTDR monomodo que se utilizará para evaluar a nivel físico la red GPON .	20
Figura 14. Prueba 1 con el OTDR en el sentido de bajada ( <i>Downstream</i> ). ....	21
Figura 15. Prueba 2 con el OTDR en el sentido de bajada desconectando las bobinas de fibra óptica. ....	23
Figura 16. Fuente de luz (izquierda) y medidor de potencia óptica (derecha). ....	24
Figura 17. Prueba con el OTDR en el sentido de subida de la red GPON en el tramo de distribución. ....	29
Figura 18. Configuración de red del TGMS. ....	34
Figura 19. Arranque de la máquina virtual donde está instalado el TGMS. ....	35
Figura 20. Acceso al TGMS. ....	35
Figura 21. Pantalla principal del TGMS. Menú <i>Devices</i> . ....	36
Figura 22. Panel de gestión de los diferentes OLTs. ....	36
Figura 23. Panel de gestión por puertos.....	37
Figura 24. Panel de gestión de las ONUs conectadas a la red GPON. ....	38
Figura 25. Monitor de estado de una ONU.....	40

Figura 26. Menú de configuración por puertos de una ONU.....	40
Figura 27. Menú de configuración de ONU.....	41
Figura 28. Apartado <i>Profiles</i> dentro del menú <i>Profiles</i> . ....	42
Figura 29. Apartado <i>Services</i> dentro del menú <i>Profiles</i> .....	42
Figura 30. Apartado <i>Bandwidth</i> dentro del menú <i>Profiles</i> .....	43
Figura 31. Apartado <i>VLAN Maps</i> dentro del menú <i>Profiles</i> . ....	43
Figura 32. Menú <i>Suscribers</i> . ....	44
Figura 33. Menú <i>Alarms</i> .....	44
Figura 34. Menú <i>Log</i> . ....	45
Figura 35. Menú <i>Configuration</i> . ....	45
Figura 36. Ejemplo de mapeado VLAN.....	49
Figura 37. Ejemplo de configuración de un mapa de ancho de banda. ....	51
Figura 38. Ejemplo de creación de un servicio de datos Ethernet. ....	52
Figura 39. Ejemplo de creación de un perfil de usuario con un servicio de datos (Internet).....	53
Figura 40. Sincronización entre el TGMS y el OLT cuando se actualiza la configuración. .....	53
Figura 41. Conexión inalámbrica a las ONUs vía Wifi. ....	55
Figura 42. Interfaz de usuario Web de los routers de las ONUs. ....	55
Figura 43. Interfaces WAN creadas con sus parámetros de configuración. ....	58
Figura 44. Información del estado de las interfaces WAN. ....	58
Figura 45. Dos interfaces WAN con <i>Status Connected</i> . ....	59
Figura 46. Selección de la interfaz WAN utilizada. ....	59
Figura 47. Información general del router. ....	60
Figura 48. Test de velocidad para comprobar las tasas de datos.....	61
Figura 49. Configuración de los canales multicast. ....	62
Figura 50. Creación de un paquete de canales multicast.....	63
Figura 51. Creación de los servicios Ethernet y multicast. ....	63

Figura 52. Creación de perfiles de usuario con servicio de Internet y vídeo.....	64
Figura 53. Configuración del servicio multicast en el menú de configuración por puertos de la ONU. ....	65
Figura 54. Asignación de packs de canales a la ONU. ....	65
Figura 55. Configuración del método de emisión multicast a través del <i>VLC Media Player</i> .....	66
Figura 56. Configuración de la dirección IP y el puerto de emisión en el <i>VLC Media Player</i> . ....	67
Figura 57. Prueba de emisión multicast. ....	68
Figura 58. Activación del modo de gestión CLI para la configuración de la red GPON. ....	72
Figura 59. Acceso al CLI desde el terminal de Windows. ....	73
Figura 60. Estructura y jerarquía de menús del CLI. ....	73
Figura 61. Movimiento por la estructura de menús del CLI. ....	76
Figura 62. Información general sobre las ONUs conectadas. ....	76
Figura 63. Relación entre las entidades del protocolo MIB para configurar el servicio de Internet. ....	80
Figura 64. Ejemplo de notificación de la actualización del estado de una ONU. ....	88
Figura 65. Relación entre las entidades del protocolo MIB para configurar el servicio de Internet y vídeo. ....	92
Figura 66. Diagrama de flujo del script <i>configuracion_red_GPON.py</i> . ....	105
Figura 67. Ejemplo de configuración de los servicios configurados en una ONU.....	107
Figura 68. Fichero de configuración XML. ....	109
Figura 69. Activación del modo de gestión TGMS. ....	111
Figura 70. Visualización del modo de gestión de la red GPON activo .....	112

## **ÍNDICE DE TABLAS**

Tabla 1. Potencias recibidas por el OLT y las ONUs observadas en el TGMS.....	26
Tabla 2. Etiquetado VLAN en los elementos de la red GPON. ....	47

## **ÍNDICE DE ECUACIONES**

Ecuación 1. Cálculo de la atenuación en el <i>splitter</i> .....	14
Ecuación 2. Cálculo de la atenuación teórica en el sentido <i>Downstream</i> de la red GPON. .....	19
Ecuación 3. Cálculo aproximado de la atenuación real en el sentido descendente de la red GPON. ....	25
Ecuación 4. Cálculo de la atenuación teórica en el sentido <i>Upstream</i> de la red GPON. .	27
Ecuación 5. Cálculo de la distancia máxima de la red GPON.....	23
Ecuación 6. Cálculo aproximado de la atenuación real en el sentido ascendente de la red GPON. ....	30



# 1

## Introducción

### 1.1 Motivación

En este Trabajo Fin de Grado se ha llevado a cabo la configuración de servicios en una red de acceso GPON (*Gigabit-capable Passive Optical Network*). Esta red óptica pasiva con capacidad de gigabit se encuentra situada en el laboratorio de Comunicaciones Ópticas (2L007) de la Escuela Técnica Superior de Ingenieros de Telecomunicación [1] de la Universidad de Valladolid.

La red GPON, cuyos dispositivos fueron comprados a la empresa TELNET Redes Inteligentes [2], es una red óptica pasiva basada en el estándar Gigabit. Este tipo de redes de acceso son las más desplegadas a nivel mundial hoy en día debido en parte a su gran simplicidad, ya no existen componentes activos en toda la planta externa, esto es, entre la oficina central del operador y las dependencias del abonado (entre 20 y 25 kilómetros de distancia). La red, basada en una topología lógica en árbol con un elemento central y varios equipos de usuario, sigue las reglas del protocolo GPON, que son el conjunto de recomendaciones G.984.x del ITU-T (*International Telecommunication Union*) [3]. Este protocolo permite obtener tasas de transmisión muy elevadas en ambos canales de comunicación. En el sentido de bajada (*Downstream*), puede llegar hasta 2,488 Gbps y en el de subida (*Upstream*), hasta 1,244 Gbps [3].

Para disponer de una red GPON, son necesarios una serie de dispositivos. Los componentes principales de esta red, también diseñados por la empresa TELNET Redes Inteligentes [2], son el elemento central que se situaría en las dependencias del operador, denominado OLT (*Optical Line Termination*) y los equipos de usuario, denominados ONUs/ONTs (*Optical Network Unit/Terminal*). Con estos elementos y utilizando un

divisor óptico que permita dividir la señal y llegar a diferentes usuarios, se puede conseguir una red de alta velocidad con topología lógica en árbol y un alcance físico de unos 20-25 kilómetros [3].

Una vez clara la topología de la red y montada en el laboratorio, el punto básico de este estudio es la configuración de servicios en la red GPON, es decir, ponerla en funcionamiento a nivel de proveedor de servicios. En este sentido, se configurarán servicios de datos (Internet) y servicios de vídeo. La red también permite configurar servicios de telefonía pero al no disponer de servidor de telefonía, el estudio se centrará en los servicios de Internet y vídeo. Para ello, se utilizarán dos plataformas de gestión desarrolladas por la empresa TELNET Redes Inteligentes [2]. La primera se denomina TGMS (*TELNET GPON Management System*) [4] y es un sistema web sencillo y fácil de manejar. Permite crear diferentes servicios y agruparlos en perfiles de usuario personalizados. Abstrae al programador de la complejidad del protocolo GPON. Por otro lado, el segundo modo de gestión, denominado CLI (*Command Line Interface*) [5], es una interfaz mucho más compleja basada en comandos. Estos comandos están íntimamente relacionados con los definidos en el protocolo GPON [3].

Finalmente, tras configurar los servicios, se diseñaron y programaron una serie de programas con los lenguajes de programación Python [6] y XML [7] (*eXtensible Markup Language*) para automatizar el control y configuración de la red. El uso de estos dos lenguajes de programación permite hacer más sencillos algunos aspectos relacionados con la gestión de la red y la configuración de servicios y perfiles de abonado.

## **1.2 Objetivos**

### **1.2.1 Objetivo General**

El objetivo principal de este Trabajo de Fin de Grado es la configuración de la red de acceso GPON de manera que los usuarios de la red puedan disfrutar de varios tipos de servicios englobados en perfiles de abonado a los cuales dichos usuarios estarán adscritos. Este objetivo general se puede desglosar en otros más específicos que se detallan a continuación.

## **1.2.2 Objetivos Específicos**

Con la realización de este estudio se han cubierto los siguientes objetivos específicos:

1. Análisis a nivel físico de una maqueta real red de acceso GPON, en concreto, su topología, su estructura y los componentes principales que la forman.
2. Utilización de elementos que permitan caracterizar redes a nivel físico, tales como OTDR (*Optical Time Domain Reflectometer*), fuente de luz y medidor de potencia óptica.
3. Configuración de servicios de Internet y vídeo y agrupación en perfiles de usuario utilizando el modo de gestión de la red TGMS.
4. Análisis del protocolo GPON para la configuración de servicios de Internet y vídeo utilizando el modo de gestión de la red CLI.
5. Automatización del proceso de gestión y configuración de la red utilizando los lenguajes de programación Python y XML (eXtensible Markup Language).

## **1.3 Fases y Métodos**

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

### **1.3.1 Fase de Análisis**

Esta fase tiene la finalidad de familiarizarse con la red de acceso GPON. Para ello, se analiza la topología y los componentes de la red. Este estudio se dividirá en dos puntos fundamentales:

- Análisis del montaje real de la red: estudio de la estructura y topología de la red y de las conexiones que tienen los componentes entre sí.
- Análisis de potencias en los canales de comunicación: estudio del rango de

potencias que hace que la red esté perfectamente operativa. Se realizará el análisis tanto en el canal de bajada (*Downstream*) como en el de subida (*Upstream*). Para ello se utilizarán dispositivos especializados, en concreto, el OTDR (*Optical Time Domain Reflectometer*), una fuente de luz y un medidor de potencia óptico.

### **1.3.2 Fase de Configuración**

Esta fase tiene como objetivo la configuración de servicios de Internet y vídeo en la red. Una vez se ha adquirido el suficiente manejo de la red a través del análisis físico anterior, se ha de poner en marcha (es decir, configurar) de forma que los usuarios de la red puedan disponer de estos servicios. Esta fase está dividida en dos partes, atendiendo al modo de gestión de la red utilizado:

- ✓ Configuración de servicios utilizando el modo de gestión TGMS [4]: se utilizará el sistema de gestión web desarrollado por TELNET Redes Inteligentes [2] para configurar los diferentes parámetros que forman los servicios y posteriormente, agrupar los mismos en perfiles de usuario que serán asignados a los clientes de la red.
  
- ✓ Configuración de servicios utilizando el modo de gestión CLI [5]: se analizará en profundidad el protocolo GPON [3] para la configuración de servicios. Esta configuración se llevará a cabo a través de comandos relacionados con lo expuesto en dicho protocolo.

### **1.3.3 Fase de Programación**

En esta fase final del proyecto se llevará a cabo una tarea de programación. Se utilizará el lenguaje de programación Python [6] como base, junto con XML [7], para realizar programas que faciliten la gestión y configuración de la red de acceso. Partiendo de lo desarrollado en la fase de configuración, se diseñará un programa general que automatice la configuración de servicios así como otros subprogramas que faciliten la gestión de la red.

### **1.3.4 Fase de Realización de los Informes**

En esta fase, se procedió a realizar los informes del proyecto:

- Informe de Prácticas de Empresa, ya que este TFG tenía asociadas una prácticas de empresa dentro del grupo de investigación.
- Memoria del Trabajo de Fin de Grado.

## **1.4 Estructura de la Memoria del TFG**

El Capítulo 2 presenta el análisis realizado a nivel físico de la red GPON. Se describe el montaje general de la red con su topología y principales elementos y se realiza un estudio sobre los niveles de potencia en los dos canales de comunicación.

En el Capítulo 3, se estudia la configuración de servicios a través del modo de gestión TGMS [4]. Se describe de forma breve la interfaz del TGMS con sus diferentes menús y las posibilidades que ofrecen y posteriormente, se enumeran los pasos necesarios para proporcionar a los usuarios servicios de Internet y vídeo. Finalmente, se analiza la configuración que debe tener los routers integrados en las ONUs para el correcto funcionamiento de los servicios.

En el Capítulo 4, se estudia la configuración de servicios a través del modo de gestión CLI [5]. Se describe de forma breve la estructura de menús que conforman el CLI y posteriormente, se enumeran los comandos necesarios para proporcionar los servicios. Para el correcto uso de estos comandos, será necesario realizar un análisis del protocolo GPON [3]. Asimismo, se irán analizando las principales diferencias con el otro modo de gestión de la red GPON.

El Capítulo 5 recoge los programas realizados en lenguaje de programación Python [6], junto con XML [7], para facilitar la configuración de la red. Se explicará el programa general de gestión que automatiza la configuración de servicios y otros scripts que permiten una gestión más sencilla de la red. Estos programas pueden resultar de gran utilidad para investigaciones futuras.

En el Capítulo 6, se recogen las conclusiones derivadas de todo el trabajo realizado en este Trabajo de Fin de Grado, así como las líneas futuras que se abren a partir de él para próximas investigaciones.

Finalmente, en el Capítulo 7, se incluyen todas las referencias bibliográficas citadas a lo largo de esta memoria.

# 2

## Análisis de la red GPON a nivel físico

### 2.1 Introducción

En este capítulo, se realiza un análisis a nivel físico de la implementación de la red de acceso GPON. La red está situada en el laboratorio de Comunicaciones Ópticas (2L007) de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Valladolid [1]. Se estudiarán en profundidad los principales elementos que componen la red, es decir, el dispositivo central de la red denominado OLT (*Optical Line Termination*) y los dispositivos situados en la zona del abonado que se denominan ONT/ONU (*Optical Network Terminal/Unit*). También se analizarán otros elementos de la red como el divisor óptico (*splitter*), los cables de fibra óptica y la caja de empalmes, que permite regular las distancias entre los diferentes componentes de la red.

Asimismo, se realizará un análisis sobre un parámetro fundamental para el funcionamiento de la red de acceso como es la potencia óptica recibida en ambos extremos de la red. En este caso, habrá que distinguir dos situaciones dependiendo del sentido de la comunicación: el sentido de bajada (*Downstream*) en el que el protocolo GPON [3] permite obtener tasas de hasta 2,488 Gbps y el sentido de subida (*Upstream*) en el que esta tasa puede llegar hasta 1,244 Gbps. La longitud de onda de transmisión no es la misma en los dos sentidos de la comunicación. En el sentido de bajada, se transmite en 1490 nm (nanómetros) mientras que en el sentido de subida, la longitud de onda utilizada es 1310 nm. Esto implica que la atenuación en la fibra no es la misma en ambos casos con lo que los niveles de potencia no serán simétricos y la atenuación en cada uno de los canales de comunicación no será la misma.

El análisis de potencias servirá para determinar el rango de distancias hasta el cual la red está operativa. Para realizar este estudio, se utilizará un OTDR (*Optical Time Domain Reflectometer*), que permitirá comprobar de forma fácil y visual los eventos y las distancias a las que se producen a lo largo de los kilómetros de fibra óptica. También se utilizará una fuente de luz y un medidor de potencia óptica recibida en ambos canales de la red para comprobar que los resultados teóricos se ajustan al montaje real.

## 2.2 Montaje real de la red GPON en el laboratorio

Una red de acceso es el conjunto de elementos que permiten a los usuarios finales conectarse con los proveedores de servicios, a través de un medio de transmisión.

La red de acceso bajo estudio se trata de una red óptica pasiva con capacidad de Gigabit, conocida como GPON (*Gigabit-capable Passive Optical Network*). Una red óptica pasiva PON (*Passive Optical Network*) es una red óptica punto-multipunto en la que no existen elementos activos entre las instalaciones del operador (OLT - *Optical Line Termination*) y el equipo terminal de usuario (ONT/ONU - *Optical Network Terminal/Unit*). Por lo tanto, toda la planta externa, formada por fibra óptica y divisores ópticos será totalmente pasiva [8]. Para ser una red GPON, debe seguir las reglas GPON, que son el conjunto de recomendaciones G.984.x del ITU-T donde se analizan las técnicas para compartir un medio común (que es la fibra óptica) por varios usuarios, encapsular la información y gestionar los elementos de red [3].

Se generan dos flujos de transmisión diferentes según el sentido de la comunicación:

- Flujo descendente (*Downstream*): desde el OLT a las ONUs/ONTs. Es un flujo punto-multipunto con una tasa definida en el estándar GPON de 2,488 Gbps. La transmisión se produce en 1490 nm, cerca de la 3ª ventana, donde la atenuación (en torno a 0,2 – 0,3 dB/km) es muy baja [3].
- Flujo ascendente (*Upstream*): desde las ONUs al OLT. Es un flujo multipunto-punto con una tasa definida en el estándar GPON de 1,244 Gbps. El acceso al medio común (la fibra óptica) requiere la implementación de un mecanismo que evite colisiones entre las diferentes

ONUs, en concreto el estándar contempla TDMA (*Time Division Multiple Access*), de forma que todos los elementos de la red GPON están fuertemente sincronizados. La transmisión se produce en 1310 nm, cerca de la 2ª ventana, donde la atenuación (en torno a 0,4 dB/km) es algo mayor que en el caso anterior [3].

A continuación, se describirá el montaje de la red de forma general y se analizarán en profundidad los principales elementos que la forman, que son el OLT y las ONUs/ONTs. El aspecto general que presenta la red de acceso GPON es el mostrado en la Figura 1.

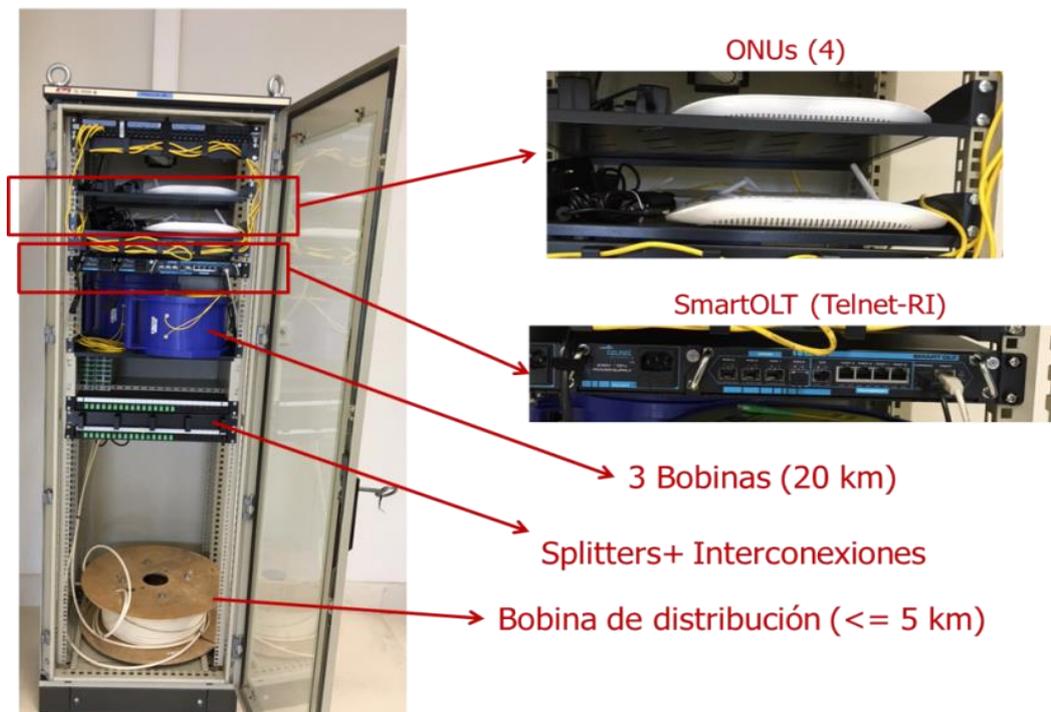


Figura 1: Vista general y por partes de la red de acceso GPON

Se observan los elementos que componen la red, todos ellos fabricados por la empresa TELNET Redes Inteligentes de manera que está garantizada su interoperabilidad [2].

En la parte izquierda de la Figura 1, se ve de forma completa la red de acceso. Las dimensiones del armario en el que está contenida la red son de unos 2 metros de alto y unos 50 centímetros de ancho. En la parte derecha, están los principales elementos de los que se compone la red.

En la parte superior, se ven las 4 ONUs/ONTs, que son los equipos de usuario. Debajo se sitúa el OLT, que es el elemento central de la red y se situaría en la oficina central del operador o proveedor de servicio. El OLT se conecta a las ONUs/ONTs mediante las 3 bobinas azules que se ven en la parte central (20 kilómetros de fibra en total), que unen el OLT con el divisor óptico (*splitter*). Estas tres bobinas contienen 5, 10 y 5 kilómetros de fibra óptica enrollados en su interior. El *splitter* o divisor óptico es de razón 1:8 (1 entrada y 8 salidas) y permite dividir la señal óptica hasta en 8 flujos diferentes. En este montaje se usará el divisor 1:8, pero en el armario hay un segundo *splitter* 2:8, que se podría conectar con el anterior para realizar montajes de dos niveles de división. Para conectar el *splitter* a las ONUs/ONTs, se utiliza la bobina de distribución situada en la parte de abajo, cuya longitud (hasta 5 kilómetros) se puede regular con la caja de empalmes negra situada justo encima. Este tramo de fibra simula que cada usuario se encuentre a una distancia distinta del OLT, tal y como existiría en una red real.

El esquema de la red, con topología en árbol, es el que se muestra en la Figura 2. Sigue la estructura típica de una red GPON [9].

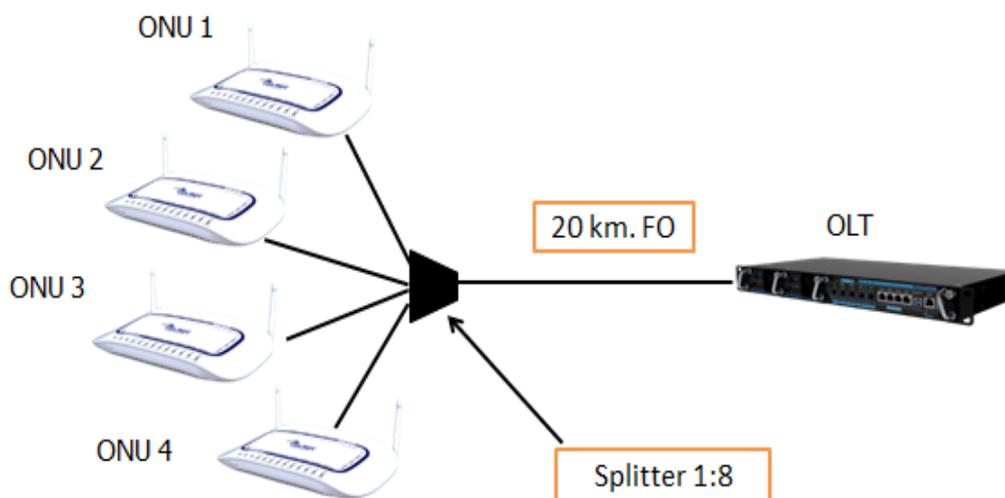


Figura 2: Esquema de la red de acceso GPON. Topología en árbol.

A continuación, se analizarán los elementos que componen la red. En primer lugar, se estudiará el OLT, elemento de gestión central de la red. El OLT es el elemento

responsable de la configuración y el control de las ONUs/ONTs. Cada ONU/ONT transmite bajo las reglas determinadas por el OLT. La forma de gestionar el OLT a la red de acceso completa puede ser mediante una interfaz web (TGMS – *TELNET GPON Management System* [4]) o mediante comandos relacionados con el estándar GPON (CLI – *Command Line Interface* [5]). Estos modos de gestión permiten controlar el número de usuarios, dar de alta o de baja nuevos abonados, proporcionar y gestionar de forma avanzada servicios de datos, vídeo y telefonía y controlar el ancho de banda proporcionado a cada usuario [10]. Estos modos de gestión se estudiarán en profundidad en los Capítulos 3 y 4, respectivamente.

El modelo de OLT utilizado en esta red es el *SmartOLT 350* [11], que implementa en su interior las funcionalidades descritas en el estándar GPON. Como se ve en la Figura 3 y de forma más detallada en la Figura 4, el *SmartOLT 350* cuenta con 4 puertos GPON denominados PON 0, PON 1, PON 2 y PON 3. Cada puerto PON está conectado con un puerto de transporte (PON 0 con PORT 0, y así sucesivamente). Cada uno de estos puertos PON soporta tasas de 2,488 Gbps en bajada y 1,244 Gbps en subida y permite la conexión de hasta 64 ONUs/ONTs, con lo que el *SmartOLT* puede tener conectados hasta 256 equipos de usuario (ONUs) [10]. Además, nuestro OLT tiene incorporado un puerto de transporte 10 Gbps, pero para su activación se requiere comprar el módulo que le permita transmitir a esa velocidad. Finalmente, el OLT viene equipado con un puerto de gestión *FastEthernet*, a través del cual será posible la gestión del OLT a través de sus dos modos (TGMS y CLI) [10].



Figura 3: Apariencia del modelo de OLT *SmartOLT 350*

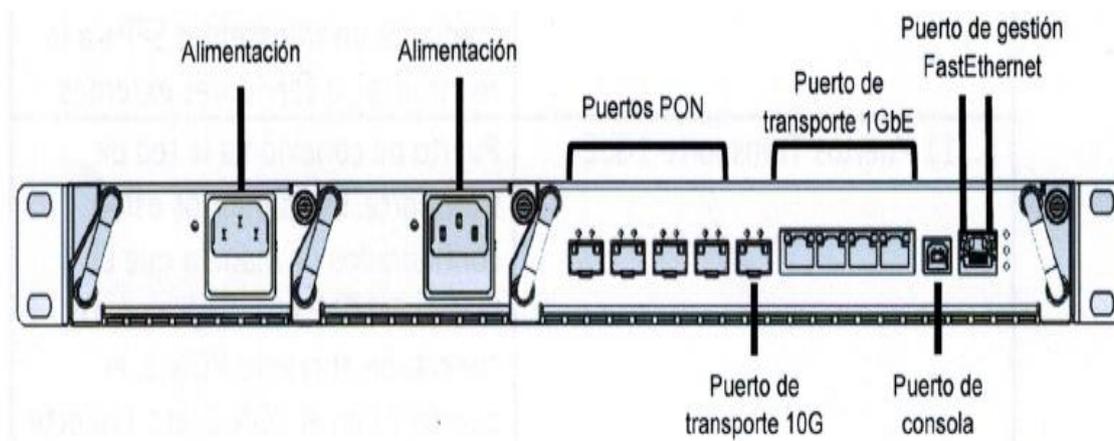


Figura 4: Esquema frontal del *SmartOLT 350*

En segundo lugar, se analizarán las características del equipo de usuario, denominado ONU o ONT (*Optical Network Unit/Terminal*). El modelo de las ONUs es el *WaveAccess 3021* [12] como se muestra en la Figura 5. Los equipos de esta serie, son terminales diseñados para el uso en entornos de hogar y de negocio. Tienen capacidades inalámbricas que permiten la conexión de varios terminales de usuario a la red de fibra óptica, basada en la tecnología GPON, que proporciona grandes tasas como se ha mencionado anteriormente [13].

Estos equipos permiten disponer de servicios de alta calidad como la conexión a Internet, el servicio telefónico de Voz sobre IP y el servicio de TV por Internet sin necesidad de ningún equipo adicional, es decir, tienen capacidades de *router* internas. En este caso, al no disponer de servidor de telefonía, la red solamente proporcionará servicio de datos y de vídeo a las ONUs/ONTs. En los Apartados 3.3 y 3.4, se analizarán en mayor profundidad las posibilidades que ofrecen las ONUs/ONTs en relación con estos servicios [12][13].

El laboratorio dispone de 4 ONUs *WaveAccess 3021* [12] de nivel 3 (con router integrado) como se ve en la Figura 6 y de clase óptica B+ (descrita en la recomendación ITU-T G.984.2, Amendment 1 [14]). Este último dato será de gran importancia en el apartado de análisis de potencias ya que determina el rango de potencias de transmisión y recepción en ambos canales de la red GPON [8].



Figura 5: Apariencia del modelo de ONU/ONT WaveAccess 3021



Figura 6: Vista frontal de las ONUs/ONTs en la red de acceso GPON del laboratorio

Por último, se analizarán el resto de elementos que forman la red. Las tres bobinas azules que se observan en la Figura 1 son bobinas que contienen 20 kilómetros de fibra óptica monomodo enrollados en su interior. Hay dos bobinas de 5 kilómetros y otra de 10 kilómetros. Estas bobinas están unidas mediante conectores de forma que se puede cambiar el orden de las bobinas o incluso se puede prescindir de alguna de las tres. En el siguiente apartado, se realizarán pruebas con diferentes longitudes. La función de estas bobinas es conectar el OLT con el divisor óptico.



Figura 7: Bobinas de fibra óptica en el montaje GPON realizado en el laboratorio

El divisor óptico o *splitter* es un elemento óptico pasivo que permite dividir la señal óptica en diferentes flujos de menor potencia que el flujo entrante. Es un elemento pasivo ya que no requiere de componentes electrónicos o señales eléctricas para la división de la señal [15]. En este caso, divide la señal procedente del OLT en diferentes flujos que comunican con las ONUs/ONTs. En este montaje concreto, solo se utiliza un nivel de *splitting* (aunque podrían utilizarse dos pues se dispone de dos divisores). El *splitter* utilizado cuenta con una entrada y ocho salidas. Por tanto, las pérdidas serán las que se muestran en la Ecuación 1.

$$Loss (dB) = 10 \times \log(n^{\circ} \text{ salidas}) = 10 \times \log(8) = 9,03 \text{ dB}$$

Ecuación 1: Cálculo de la atenuación en el *splitter*

Tal y como se mencionó anteriormente, aparte del *splitter* con una entrada y ocho salidas, el laboratorio dispone de otro divisor óptico con dos entradas y ocho salidas. En la imagen de la Figura 8, se observa el *splitter* de una sola entrada utilizado (izquierda) y el *splitter* de dos salidas (derecha).



Figura 8: Divisores ópticos o *splitters*

Finalmente, los elementos restantes son la bobina de distribución y la caja de empalmes (Figura 9). Como ya se ha explicado, del *splitter* salen diferentes flujos que van hacia las ONUs/ONTs. La bobina de distribución contiene 5 kilómetros de fibra óptica monomodo de tipo G652-D [16], que conectan el *splitter* con las ONUs/ONTs. La distancia entre estos dos elementos se puede regular con la caja de empalmes. La fibra monomodo G652-D también se utiliza en las tres bobinas azules que contienen los 20 kilómetros de fibra.



Figura 9: Caja de empalmes (parte superior) y bobina de distribución (parte inferior)

La caja de empalmes cuenta con 16 puertos de entrada y 16 puertos de salida. La señal procedente del *splitter* se conecta a un puerto de entrada y sale por un puerto de salida. La distancia entre el *splitter* y la ONU/ONT en cuestión dependerá de la combinación de puerto de entrada y puerto de salida (siendo el máximo los 5 kilómetros de fibra óptica que hay en el interior de la bobina de distribución).

En la Figura 10 y la Figura 11, se muestra el esquema general de la red, donde se detallan las distancias y la distribución de distancias de la caja de empalmes. Por ejemplo, si se conecta la salida del *splitter* al puerto de entrada 7 y el puerto de salida 7 se conecta a una ONU, la distancia entre el *splitter* y la ONU sería de 300 metros. Se puede también concatenar distancias: si se conecta la salida del *splitter* al puerto de entrada 7, el puerto de salida 7 al puerto de entrada 13 y el puerto de salida 13 a la ONU, la distancia entre el *splitter* y la ONU sería de  $300 + 500 = 800$  metros. La concatenación de los 15 puertos de entrada y de salida sería la distancia máxima entre *splitter* y ONU (los 5 kilómetros de la fibra de distribución).

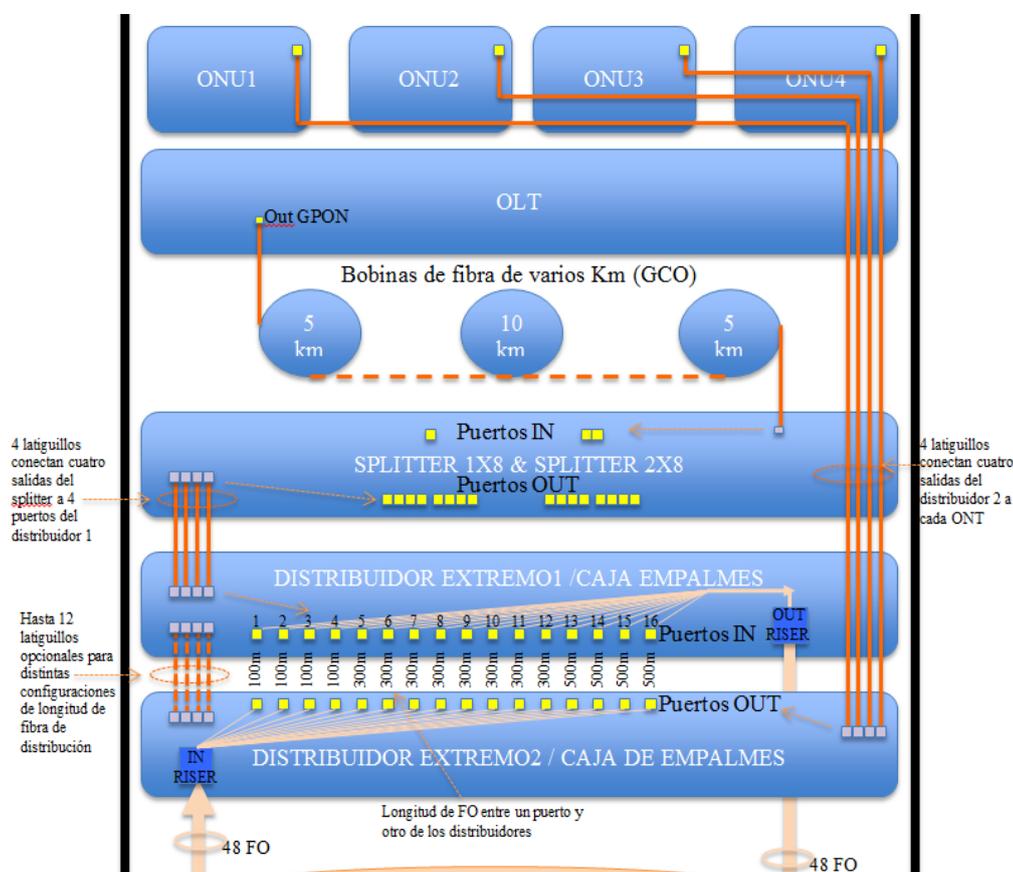


Figura 10: Esquema detallado de la red GPON

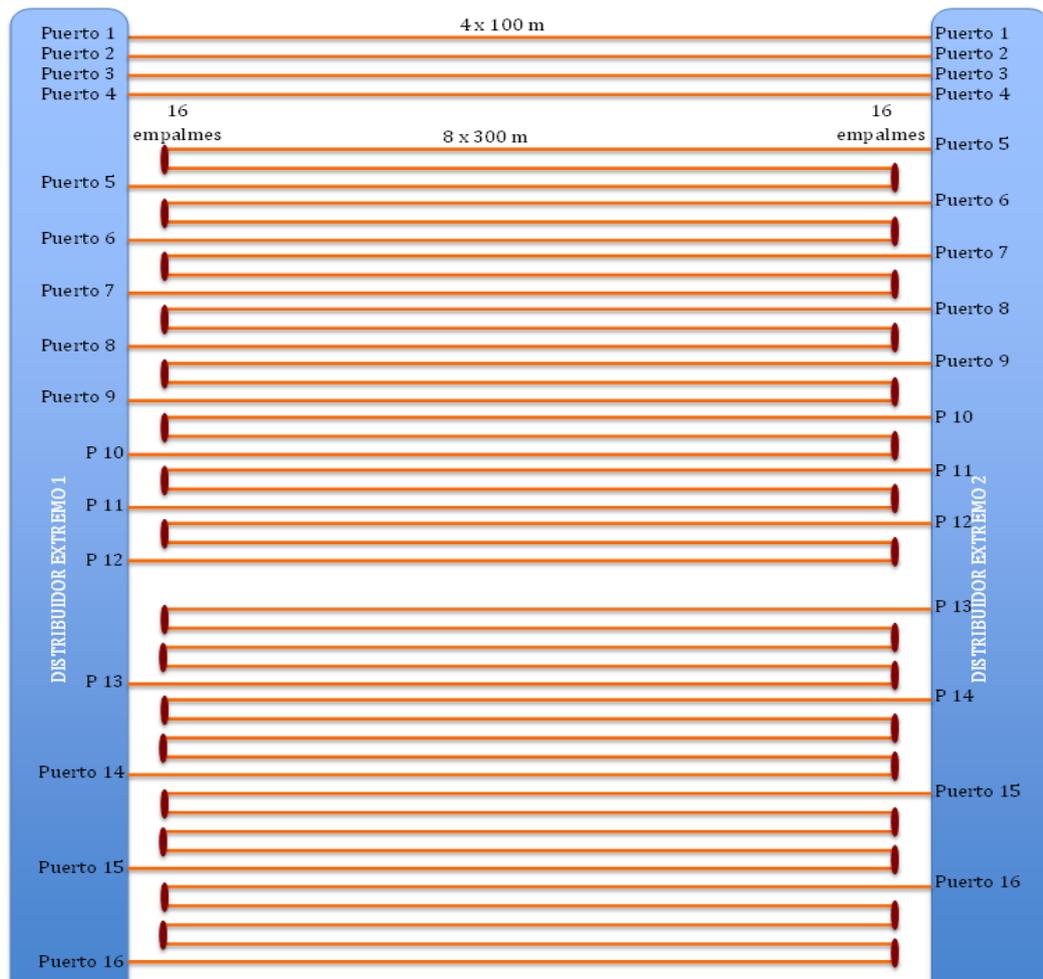


Figura 11: Configuración de la caja de empalmes

### 2.3 Análisis de potencias y pruebas con OTDR en sentido Downstream

Una vez estudiada la estructura a nivel físico, es hora de tratar un aspecto muy importante en toda red de comunicaciones como es la potencia óptica recibida en ambos extremos de la red GPON. La gran ventaja de la fibra óptica respecto a otros medios de transmisión es su baja atenuación pero para el correcto funcionamiento de la red es necesario que el nivel de potencia se mantenga entre unos límites de forma que no se produzca saturación ni el nivel sea tan bajo que no se detecte la señal.

Como se ha mencionado anteriormente, la comunicación se produce en dos sentidos de forma que hay dos canales de comunicación, es decir, dos flujos de información caracterizados por parámetros diferentes. En este apartado, el análisis se centrará en el sentido de bajada, denominado *Downstream*.

El flujo de bajada *Downstream* es el que va desde el OLT hacia las ONUs. Es por tanto una comunicación punto-multipunto. La transmisión se produce en la longitud de onda de 1490 nm, en la tercera venta de transmisión donde la atenuación es muy baja (en torno a 0,2 – 0,3 dB) [3][8] (Figura 12).

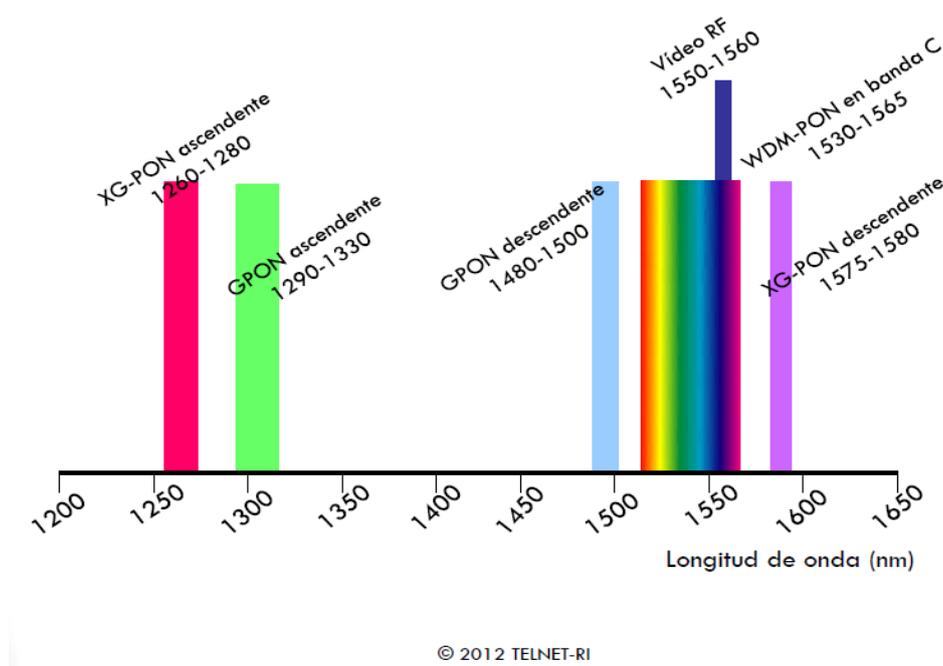


Figura 12: Longitudes de onda utilizadas en GPON

En primer lugar, se realizará un cálculo teórico de la atenuación que se produce en la red en el sentido de bajada. El cálculo es aproximado y para realizarlo, se tomarán una serie de premisas:

- Se consideran las 3 bobinas de fibra óptica y por tanto, el dato de la distancia es 20 kilómetros.
- Se desprecia la distancia entre el splitter y las ONUs ya que es una distancia muy corta (puede llegar a 5 kilómetros pero lo habitual es que esta distancia no supere el kilómetro en el caso que nos atañe). No se consideran las pérdidas por empalmes ya que los que hay en la caja de empalmes se realizaron con la fusionadora automática y el nivel de pérdidas es prácticamente cero. Asimismo, la fórmula empleada prevé que pueda haber dos niveles de *splitting*. Sin embargo, en el montaje llevado a

cabo en el laboratorio para las pruebas realizadas, solo hemos considerado un único nivel.

- Los datos de atenuación de la fibra y la de los conectores son los empleados por la empresa TELNET Redes Inteligentes para realizar sus cálculos [8]. El dato de atenuación tomado para 1490 nm será de 0,3 dB/km y las pérdidas introducidas por conectores serán de < 0,5 dB/km (se tomará 0,5 para que considerar el peor caso) ya que las ONUs son de clase óptica B+ [12].

La ecuación para calcular la atenuación es la que se muestra en la Ecuación 2.

$$\begin{aligned} \text{Atenuación teórica Downstream} &= (\text{Atenuación Splitter 1} + \text{Atenuación Splitter 2}) + \\ &+ (\text{Atenuación fibra/km} \times \text{Distancia}) + (\text{Atenuación empalme} \times n^{\circ} \text{ empalmes}) + \\ &+ (\text{Atenuación conectores} \times n^{\circ} \text{ conectores}) = (9\text{dB} + 0) + (0,3 \text{ dB/km} \times 20 \text{ km}) + 0 + \\ &+ (0,5 \text{ dB/conector} \times 12 \text{ conectores}) = 21\text{dB} \end{aligned}$$

Ecuación 2: Cálculo de la atenuación teórica en el sentido *Downstream* de la red GPON

La atenuación del splitter ya fue calculada en la Ecuación 1. El número de conectores es debido a las uniones entre OLT, bobinas de fibra, *splitter*, caja de empalmes y ONUs (podría ser mayor o menor dependiendo de las conexiones entre bobinas y en la caja de empalmes pero se toma este número como término medio). Estos 21 dB de atenuación son un cálculo aproximado ya que se han tomado valores máximos de seguridad, pero en la realidad relativamente menores, de forma que existe un margen en el cálculo final.

El rango de atenuación de la clase óptica B+ es de 13 a 28 dB [8] con lo que el dato teórico de atenuación (21 dB) es aceptable. Si se prescinde de algunos de los 20 kilómetros de fibra, la atenuación baja pero el valor sigue por encima del límite inferior de 13 dB. Si por el contrario se prescinde del *splitter*, la atenuación estará por debajo del límite y por tanto, se produce saturación en los elementos de la red. Esto se observa en el TGMS [4]. En caso de haber saturación, el campo *Status* del panel de gestión de ONUs (Figura 24) muestra el valor ‘OMCI Configuration Error’, lo que indica que el nivel de potencia no se encuentra entre los límites permitidos. Por esto, se hace necesario la presencia de al menos un nivel de *splitting* (además de por la necesidad de dividir la señal).

En el siguiente apartado, se calculará la atenuación en el sentido de subida (*upstream*), que será ligeramente superior por la mayor atenuación de la fibra óptica en la longitud de onda de 1310 nm (segunda ventana). Al ser mayor que la atenuación en el sentido de bajada, se utilizará ese dato más restrictivo para calcular la máxima distancia que puede haber entre el OLT y las ONUs, es decir, se calcularán el número de kilómetros de fibra que, con un nivel de *splitting*, hacen que siga llegando suficiente potencia a los elementos de la red [8].

A continuación, se explicará cómo utilizar el OTDR (*Optical Time Domain Reflectometer*) para visualizar de forma clara los diferentes eventos que suceden en la fibra (pérdidas por inserción en conectores, *splitter*, etc.) en el sentido descendente, es decir, desde el OLT hasta las ONUs/ONTs.

El OTDR (Figura 13) es un instrumento optoelectrónico que representa de forma visual diferente información sobre un enlace o red de fibra óptica, como puede ser las pérdidas de la fibra óptica, la longitud de la red o enlace, la posición de los diferentes eventos existentes en la misma, el estado de conectores, empalmes, fusiones u otros dispositivos. Este elemento permite en caso de haber un problema, localizarlo de forma sencilla y ver de qué puede tratarse y cómo solucionarlo, por ejemplo, un conector sucio o en mal estado [17].



Figura 13: OTDR monomodo que se utilizará para evaluar a nivel físico la red GPON

Los parámetros básicos de configuración se enumerarán a continuación. Es muy importante configurarlos de forma adecuada para apreciar bien los diferentes eventos. En concreto, las principales características configurables son:

- Longitud de onda: en el OTDR disponible en el laboratorio (monomodo), hay dos: 1310 nm, que será usada para el sentido de subida y 1550 nm, que será empleada para las mediciones en el sentido de bajada, al ser la longitud de onda disponible más cercana a los 1490 nm que establece el estándar.
- Ancho del pulso: es el parámetro más importante pues determina la resolución de la medición. Cuanto más estrecho sea (del orden de ns), mayor resolución y los eventos juntos se verán mejor habrá pero el rango dinámico será más pequeño y la traza podría ser ruidosa. Un pulso largo (del orden de  $\mu$ s) da un mayor margen dinámico y permite detectar eventos más lejanos pero a costa de perder resolución [17].

Para hacer pruebas en el sentido *Downstream*, se debe desconectar el OLT y en su lugar conectar el OTDR. La primera prueba se realizará con las 3 bobinas azules (20 kilómetros de fibra óptica conectados), y los resultados de la traza completa pueden verse en la Figura 14.

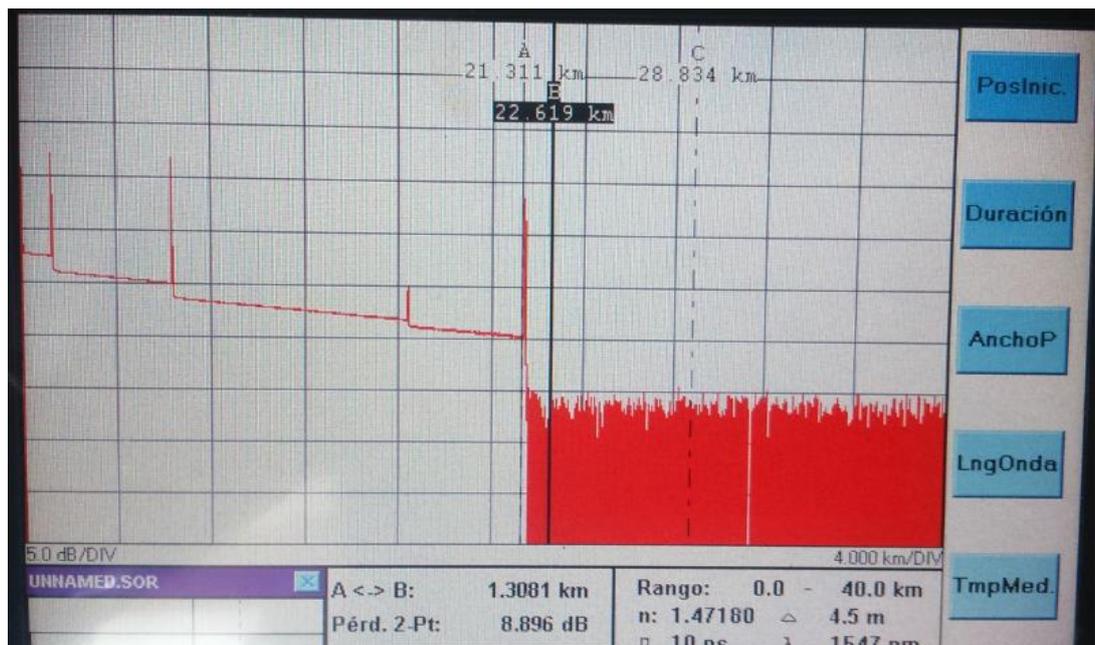


Figura 14: Prueba 1 con el OTDR en el sentido de bajada (*Downstream*)

La traza de la Figura 14 permite ver los 20 kilómetros de fibra entre el OLT y el splitter. El marcador A marca 21,3 km aproximadamente, a lo que hay que restar 1,3 km que son de fibra de lanzamiento. Esta fibra de lanzamiento es necesaria ya que la potencia al principio de la fibra es muy alta de tal forma que se pueden producir reflexiones que pueden saturar el receptor, creando *deadzones* (regiones donde las reflexiones ocultan datos de medidas) bastante grandes [17].

El primer evento que se visualiza (pegado al margen izquierdo de la Figura 14) corresponde al conector que une la fibra de lanzamiento con el OTDR. El segundo pico se debe a la unión entre la fibra de lanzamiento y los 20 kilómetros de fibra. Posteriormente, se ven tres eventos que corresponden a las conexiones entre las tres bobinas de fibra óptica y en el último además, se suma la alta caída debida al splitter. Los picos de estas tres conexiones son muy diferentes; esto se puede deber a que algún conector esté sucio. Entre los eventos, se observa que la atenuación debida a la fibra óptica es muy baja (el nivel de señal baja muy poco entre los diferentes eventos). Una vez se produce la caída debida al *splitter*, la traza empieza a ser ruidosa con lo que no se consiguen distinguir los picos debidos a las ONUs.

Para poder observar las ONUs, se desconectarán algunos kilómetros de fibra de forma que la distancia entre el OLT y las ONUs sea menor y se puedan ver mejor los eventos que están juntos (las ONUs), tal y como se muestra en la Figura 15. Se utilizará un pulso de 300 ns (estrecho) para poder distinguir los eventos que estén próximos [17]. Como ya se explicó anteriormente, desconectar algunos kilómetros de fibra no supone ningún problema ya que la atenuación se sigue manteniendo dentro de los límites y no se produce saturación.

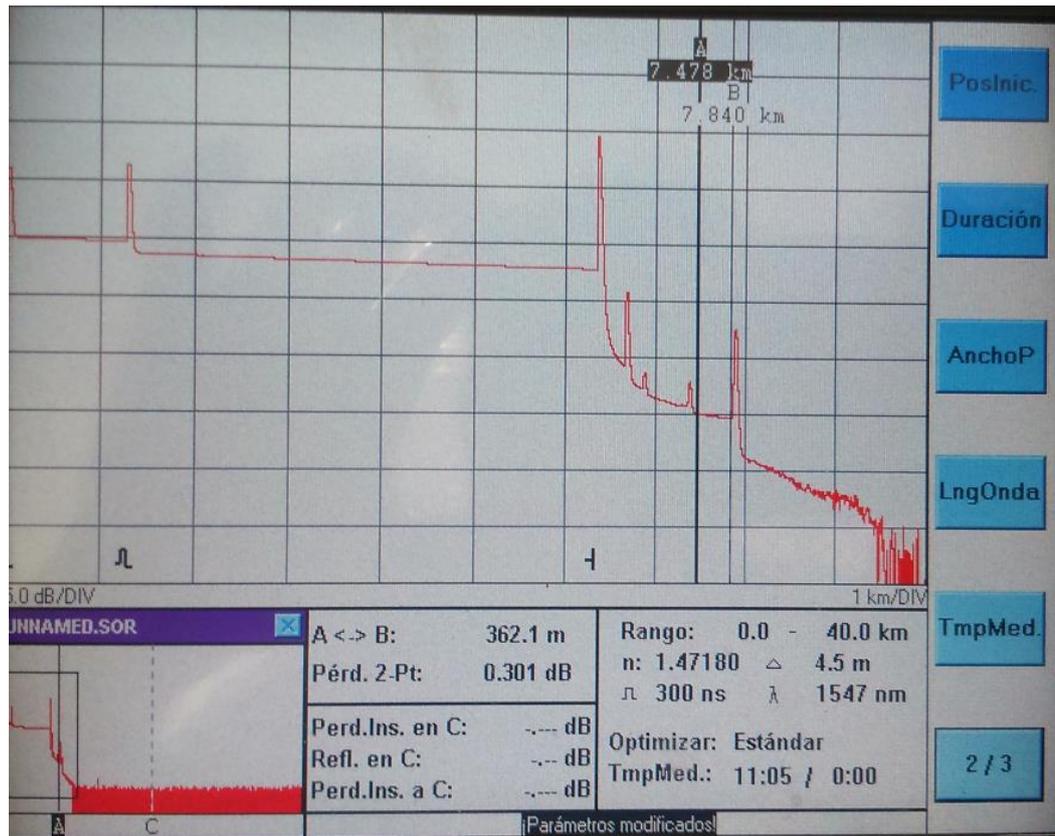


Figura 15: Prueba 2 con el OTDR en el sentido de bajada desconectando las bobinas de fibra óptica

En esta segunda prueba, se han desconectado 15 kilómetros de fibra de modo que solo hay conectada una bobina de 5 kilómetros. El primer pico corresponde al conector que está al final de la fibra de testeo (1,3 kilómetros). El siguiente evento es la caída debida al *splitter*. A partir de ahí, se observan cuatro eventos que corresponden a cada una de las cuatro ONUs. La diferencia entre los picos de estos eventos puede deberse al estado de los conectores. Cada ONU está situada a una distancia diferente del *splitter* (esto se consigue, como se ha visto antes, con la caja de empalmes; ver Figura 10 y Figura 11) de ahí que cada evento aparezca a una distancia diferente. En caso de que se hubiera configurado la misma distancia para varias ONUs, los eventos se superpondrían y no se podrían distinguir.

Finalmente, para comprobar que los datos calculados de atenuación son fiables, se utilizarán la fuente de luz y el medidor de potencia óptica, tal y como se muestra en la Figura 16.



Figura 16: Fuente de luz (izquierda) y medidor de potencia óptica (derecha)

La fuente de luz (*PON LIGHT SOURCE*), del fabricante fibercom [18], puede transmitir en tres longitudes de onda: 1310, 1490 y 1550 nm. El medidor de potencia óptica (*OPTICAL POWER METER*), del fabricante Haktronics [19], puede realizar medidas en 820, 850, 1310 y 1550 nm. Para realizar medidas en el sentido de bajada (*Downstream*), se transmitirá en 1490 nm con la fuente de luz y se medirá la potencia óptica en 1550 nm que es la longitud de onda más próxima y en la que la atenuación es bastante similar a 1490 nm.

En primer lugar, se configurará la fuente de luz transmitiendo en 1490 nm y el medidor de potencia óptica midiendo en 1550 nm. La potencia que mida el medidor habrá que restarla luego en el cálculo de la atenuación. En este caso, la potencia medida es de -2,29 dBm. Este valor se tomará como referencia.

Para realizar la medición de la atenuación en el sentido de bajada (con los 20 kilómetros de fibra óptica conectados entre OLT y *splitter*), se ha de desconectar el OLT

y en su lugar, conectar la fuente de luz. Asimismo, hay que desconectar una ONU y conectar en ese punto el medidor de potencia óptica. El medidor refleja una potencia de -20,65 dBm. Por tanto, la atenuación de extremo a extremo en el sentido de bajada es la que aparece en la Ecuación 3.

$$\begin{aligned} \textit{Atenuación Downstream} &= \textit{Valor referencia} - \textit{Valor medido} = \\ &= -2,29 \textit{ dBm} - (-20,65 \textit{ dBm}) = 18,36 \textit{ dB} \end{aligned}$$

Ecuación 3: Cálculo aproximado de la atenuación real en el sentido descendente de la red GPON

La atenuación teórica era de 21 dB pero ya se comentó que los datos teóricos de atenuación en la fibra y pérdidas por conectores eran más altos que los reales para tener un margen de seguridad. De esta forma, los resultados teóricos y prácticos son bastante similares y ambos entran dentro de los límites de atenuación de la clase óptica B+ (13-28 dB) [8]. Como ya se explicó anteriormente, se pueden desconectar algunos kilómetros de fibra pero no se puede prescindir del splitter, ya que además de ser necesario para dividir la señal, la atenuación sería muy baja de forma que se produciría saturación y el TGMS [4] mostraría el valor ‘OMCI Configuration Error’ en el campo *Status* del panel de gestión de ONUs (Figura 24) indicando que la potencia recibida no se encuentra entre los límites permitidos.

Con ayuda del medidor de potencia óptica, también se puede averiguar la potencia a la que transmiten el OLT y las ONUs. Teóricamente, el OLT transmite entre 1,5 dBm y 5 dBm, según los datos proporcionados por el fabricante en sus respectivos manuales [10]. Por su parte, las ONUs transmiten entre -0,5 y 5 dBm [13]. Conectando el medidor de potencia óptico a la salida de ambos, se obtiene que el OLT transmite en torno 1,75 dBm mientras que las ONUs en un valor próximo a 2 dBm, lo que coincide con lo expuesto teóricamente.

El TGMS (*TELNET GPON Management System*) [4], que se estudiará en mayor profundidad en el Capítulo 3, permite conocer los valores de potencia recibidos por el OLT y por las ONUs, con lo que podemos comparar con todos los valores indicados anteriormente. En la Tabla 1, se recogen estos valores que permitirán comprobar que los cálculos realizados en este apartado se ajustan a la realidad. Estos valores están tomados con los 20 kilómetros de fibra óptica conectados entre el OLT y el *splitter*, que es la configuración con la que se han realizado todos los cálculos anteriores. Las distancias

entre el *splitter* y las ONUs son diferentes pero resultan casi despreciables a efectos de cálculo pues son muy pocos metros de diferencia [8].

ONU	Distancia entre splitter y ONU	Potencia recibida por el OLT	Potencia recibida por la ONU
1	100 metros	-17,59 dBm	-16,36 dBm
2	500 metros	-17,93 dBm	-15,88 dBm
3	300 metros	-17,10 dBm	-16,17 dBm
4	100 metros	-17,54 dBm	-16,86 dBm

Tabla 1: Potencias recibidas por el OLT y las ONUs observadas en el TGMS

De los datos de la Tabla 1, se analizarán los correspondientes al sentido de bajada. El sentido *Downstream* corresponde a la potencia recibida por las ONUs cuando transmite el OLT. El OLT transmite a un valor en torno a 1,75 dBm y la atenuación *Downstream* medida era de 18,36 dB. Por tanto, el valor de potencia recibida por la ONU debe estar en torno a  $1,75 \text{ dBm} - 18,36 \text{ dB} = -16,61 \text{ dBm}$ . Efectivamente, los valores de potencia recibidos por las ONUs y registrados en el TGMS [4] son muy similares a esta cifra, tal y como se observa en los datos recogidos de la Tabla 1. Además estos valores se encuentran dentro del rango permitido, ya que la sensibilidad de las ONUs de clase óptica B+ es de -28 dBm y la potencia de saturación es -8 dBm [13].

## 2.4 Análisis de potencias y pruebas con OTDR en sentido Upstream

En este apartado, se realizará el mismo análisis que en el apartado anterior pero para el otro flujo de comunicación, el sentido de subida (*Upstream*).

El flujo de subida (*Upstream*) es el que va desde las ONUs hacia el OLT. Es por tanto un flujo multipunto-punto con una tasa definida en el estándar GPON de 1,244 Gb/s. El acceso al medio común (la fibra óptica) requiere la implementación de un mecanismo que evite colisiones entre las diferentes ONUs, en concreto TDMA (*Time*

*Division Multiple Access*), de forma que todos los elementos de la red GPON deben estar altamente sincronizados. La transmisión se produce en torno 1310 nm como se observa en la Figura 12, cerca de la 2ª ventana, donde la atenuación (en torno a 0,4 dB/km) es algo mayor que para el sentido de bajada [3].

Al igual que para el *Downstream*, en primer lugar se realizará un cálculo de la atenuación producida en el sentido de subida de extremo a extremo (Ecuación 4). Las premisas para realizar el cálculo son las mismas, es decir, se desprecian las pérdidas por empalmes y la distancia entre *splitter* y ONUs, se consideran 20 kilómetros de fibra y los datos teóricos son los utilizados por la empresa TELNET Redes Inteligentes [2].

$$\begin{aligned} \text{Atenuación teórica Upstream} &= (\text{Atenuación Splitter 1} + \text{Atenuación Splitter 2}) + \\ &+ (\text{Atenuación fibra/km} \times \text{Distancia}) + (\text{Atenuación empalme} \times n^{\circ} \text{ empalmes}) + \\ &+ (\text{Atenuación conectores} \times n^{\circ} \text{ conectores}) = (9\text{dB} + 0) + (0,4 \text{ dB/km} \times 20 \text{ km}) + 0 + \\ &+ (0,5 \text{ dB/conector} \times 12 \text{ conectores}) = 23\text{dB} \end{aligned}$$

Ecuación 4: Cálculo de la atenuación teórica en el sentido *Upstream* de la red GPON

La atenuación del *splitter* ya fue calculada en la Ecuación 1. El número de conectores es debido a las uniones entre OLT, bobinas de fibra, *splitter*, caja de empalmes y ONUs (podría ser mayor o menor dependiendo de las conexiones entre bobinas y en la caja de empalmes pero se toma este número como término medio). Como se mencionó en cálculos anteriores, estos valores teóricos son mayores que los reales para garantizar un margen de seguridad.

Estos 23 dB de atenuación están dentro del rango de la clase óptica B+ a la que pertenecen las ONUs. Como se indicó en el cálculo de la atenuación el sentido de bajada, se pueden desconectar algunos kilómetros de fibra ya que la atenuación seguiría dentro de los límites pero el *splitter* es necesario para llegar al valor mínimo de atenuación y no aparezcan problemas de saturación.

Al ser la atenuación mayor en el sentido de subida que en el de bajada (la atenuación de la fibra óptica es mayor en 1310 nm que en 1490 nm), se utilizará el dato de la atenuación *Upstream* para calcular la máxima distancia que puede abarcar la red, esto es, la máxima distancia entre el OLT y las ONUs. Según el protocolo GPON, la distancia máxima de una red GPON está en torno a los 20-25 kilómetros [3]. La ecuación que rige el cálculo de la distancia máxima de la red GPON se muestra en la Ecuación 5.

$$\begin{aligned}
 & \text{Distancia máxima red GPON} \\
 & = \frac{((\text{Pot mínima} - \text{Sensibilidad} - \text{Guarda}) - (\text{Atenuación Splitters, Empalmes, Conectores}))}{\text{Atenuación fibra}} \\
 & = \frac{\left( (-0,5 \text{ dBm} - (-28 \text{ dBm}) - 2 \text{ dB}) - \left( 9 \text{ dB} + 0 + 0,5 \frac{\text{dB}}{\text{conector}} * 12 \text{ conectores} \right) \right)}{0,4 \frac{\text{dB}}{\text{km}}} \\
 & = 26,25 \text{ km}
 \end{aligned}$$

Ecuación 5: Cálculo de la distancia máxima de la red GPON

Los datos tomados son los empleados por TELNET Redes Inteligentes para realizar sus cálculos y vienen recogidos en los correspondientes manuales [8][10]. De este modo, se considera un margen de guarda de 2 dB, una potencia mínima de transmisión de las ONUs (en el sentido *Upstream* transmiten las ONUs y recibe el OLT) de -0,5 dBm y una sensibilidad de -28 dBm [13].

Con estas condiciones, la distancia máxima entre OLT y ONU es 26,25 kilómetros. La red GPON bajo estudio cumple este requisito ya que la distancia máxima que podría haber entre OLT y ONUs es de 25 kilómetros (20 kilómetros entre OLT y *splitter* + 5 kilómetros de la bobina de distribución) y además cumple con la distancia máxima que considera el protocolo GPON [3].

Al igual que en el apartado anterior, se utilizará el OTDR para ver los diferentes eventos que hay a lo largo de la red GPON en este canal de la red. En este caso, al estar en el sentido de subida, la longitud de onda que se debe seleccionar en el OTDR es 1310 nm [17], por lo que el OTDR tenemos que configurarlo en esta longitud de onda.

Ahora, a diferencia del sentido de bajada, para realizar la medida se debe desconectar una ONU y conectar el OTDR en su lugar. De esta forma, primero se observará el tramo de fibra de distribución que une con el *splitter* y posteriormente, se verán los 20 kilómetros de fibra óptica que unen *splitter* y OLT. Puesto que el tramo de 20 kilómetros ya se ha analizado en el apartado anterior, la prueba realizada ahora mostrará solamente la parte correspondiente a la bobina de distribución, tal y como se muestra en la Figura 17. Para visualizar esta parte con alta resolución, se ha utilizado un tipo de pulso estrecho de 10 ns [17].

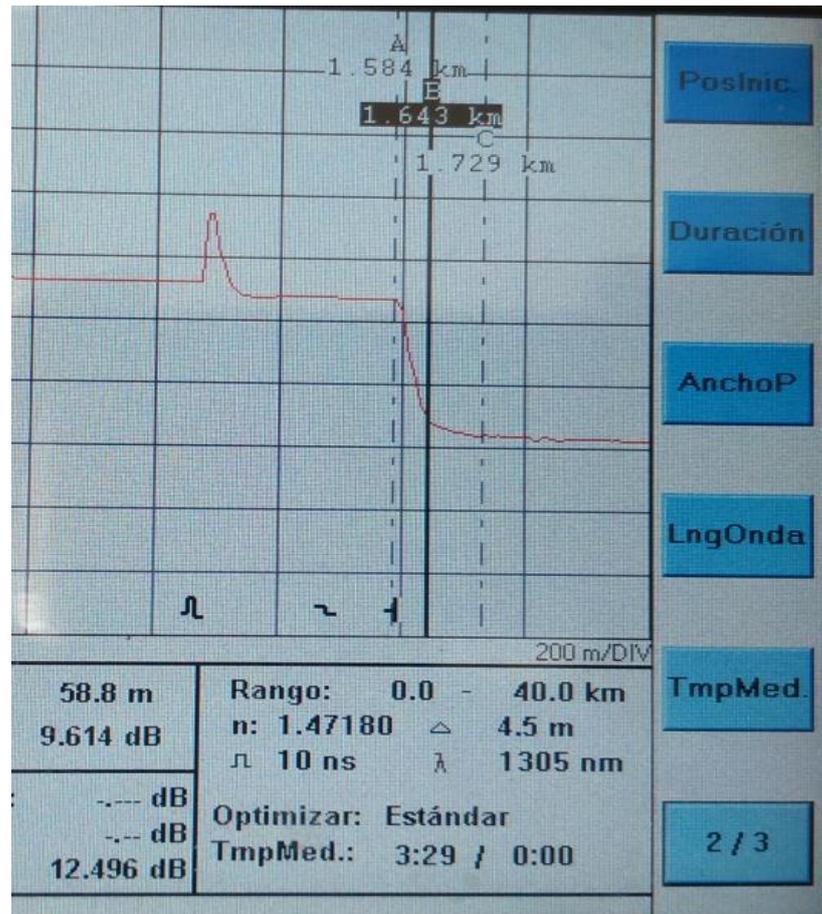


Figura 17: Prueba con el OTDR en el sentido de subida de la red GPON en el tramo de distribución

En esta imagen de la Figura 17, se observa que, una vez superado el pico en torno a 1,3 km debido al conector que une con la fibra de testeo, hay unos 300 metros hasta la caída debida al *splitter*. Esto significa que la distancia entre ONU y *splitter* configurada en la caja de empalmes es de 300 metros y se habrá empleado algún puerto del 5 al 12 (ver Figura 10 y Figura 11).

Una vez hechos los cálculos teóricos y realizadas las pruebas con el OTDR, es momento de comprobar los resultados obtenidos. Para ello, se volverán a utilizar la fuente de luz y el medidor de potencia óptica.

En primer lugar, se conectará la fuente de luz transmitiendo en 1310 nm y el medidor de potencia óptica midiendo en 1310 nm. En este caso, a diferencia del canal *Downstream*, la medida será más exacta al poder disponer de los dos aparatos operativos en la longitud de onda adecuada. La potencia reflejada en el medidor será la referencia y

habrá que restarla posteriormente en el cálculo de la atenuación. El valor obtenido es de -3,26 dBm.

Para realizar la medida, se desconectará una de las ONUs y en su lugar, se conectará la fuente de luz. El medidor de potencia óptico se conectará en lugar del OLT de forma que se pueda medir la atenuación extremo a extremo. El valor reflejado en el medidor de potencia óptico es de -22,72 dBm. Por tanto, la atenuación en el sentido de subida es la que se muestra en la Ecuación 6.

$$\begin{aligned} \textit{Atenuación Upstream} &= \textit{Valor referencia} - \textit{Valor medido} = \\ &= -3,26 \textit{ dBm} - (-22,72 \textit{ dBm}) = 19,46 \textit{ dB} \end{aligned}$$

Ecuación 6: Cálculo aproximado de la atenuación real en el sentido ascendente de la red GPON

El valor teórico de atenuación era de 23 dB, pero los datos teóricos utilizados por la empresa TELNET Redes Inteligentes son más altos que los reales para tener un margen de seguridad. Por tanto, los resultados teóricos se adecúan a los resultados prácticos y ambos valores están dentro del rango permitido en la clase óptica B+ (13-28 dB) [8].

Finalmente, se realizará la comprobación de los resultados obtenidos. En el sentido de subida, la comunicación se produce desde las ONUs hacia el OLT con lo que es necesario saber el valor de potencia de transmisión de las ONUs. Este valor se obtuvo, con ayuda del medidor de potencia óptico, en el apartado anterior y era aproximadamente 2 dBm (valor aceptable pues debe estar entre -0,5 y 5 dBm según el manual de la ONU [13]). Por tanto, la potencia recibida por el OLT es  $2 \textit{ dBm} - 19,46 \textit{ dB} = -17,46 \textit{ dBm}$ . Se observa que este valor calculado con los resultados obtenidos es muy similar al reflejado por el TGMS [4] (ver Tabla 1) y se encuentra en el rango permitido de potencias que admite el OLT para la clase óptica B+ (potencia de saturación 8 dBm y sensibilidad -28 dBm) [10].

## 2.5 Conclusiones

La red de acceso GPON situada en el laboratorio de Comunicaciones Ópticas se trata de una red óptica pasiva (ya que no tiene elementos activos entre la parte del operador y los equipos de usuario) que sigue las reglas definidas en el estándar

GPON. Los elementos principales de esta red son el OLT (*Optical Line Termination*) y las ONUs/ONTs (*Optical Network Unit/Terminal*). El OLT se conecta a través de 20 kilómetros de fibra óptica a un divisor óptico (*splitter*) que divide la señal hacia las diferentes ONUs (en esta red son 4) con ayuda de una bobina de distribución de 5 kilómetros y una caja de empalmes que regula la distancia.

Existen dos flujos de datos dependiendo del sentido de la comunicación. El flujo de bajada (*Downstream*) es el que va desde el OLT hacia las ONUs y utiliza la longitud de onda de 1490 nm. El sentido de subida (*Upstream*) va de las ONUs hacia el OLT y emplea la longitud de 1310 nm. Es muy importante cumplir los requisitos de potencia de la red y sus elementos. Por tanto, es necesario que la atenuación de extremo a extremo esté dentro de unos valores determinados y para que no se produzca saturación (los equipos reciben demasiada potencia), se ha de incluir al menos un nivel de *splitting*. Por el otro lado, para que los equipos reciban un nivel de potencia mayor que su sensibilidad, la red no debe superar una atenuación máxima. Este hecho determina la longitud máxima entre los extremos de la red GPON.

Para realizar este análisis de potencias y atenuaciones, se utilizó el OTDR, en el que se pueden ver los diferentes eventos que suceden a lo largo del enlace de fibra óptica de la red GPON y el conjunto fuente de luz–medidor de potencia óptica. Estos aparatos han permitido obtener resultados prácticos de los valores de potencia de transmisión de los componentes y de la atenuación en los dos canales de la red. Estos datos prácticos son muy similares a los resultados teóricos por lo que el análisis realizado resulta coherente.



# 3

## **Configuración de servicios en la red GPON utilizando el modo de gestión TGMS**

### **3.1 Introducción**

En este capítulo, se describe uno de los dos modos de gestión de la red GPON, el denominado TGMS (*TELNET GPON Management System*). Se trata de una plataforma web que permite la gestión de varios OLT así como de las ONUs conectadas. Cuenta con una interfaz sencilla y muy visual que permite al usuario configurar la red sin necesidad de conocer en profundidad el protocolo GPON [3].

En primer lugar, se realizará una pequeña descripción de los menús que forman el TGMS [4]. Posteriormente, se describirán los pasos necesarios para poner en marcha los servicios de la red, es decir, proporcionar a los abonados servicio de datos y vídeo. De una forma sencilla, el TGMS permite configurar los diferentes servicios (datos, voz, televisión...) que se van a ofrecer a los abonados y agruparlos en perfiles de usuario. Asimismo, el TGMS tiene el control sobre el estado de los equipos de la red de forma que el sistema lanza un aviso en caso de que se produzca algún fallo o anomalía, tanto a nivel físico como de servicio.

### **3.2 Interfaz TGMS: Telnet GPON Management System**

El TGMS (*TELNET GPON Management System*), o sistema de gestión GPON de TELNET [4], es una plataforma web que permite el control de todos los elementos conectados a la red GPON. Permite controlar varios OLTs siempre que se encuentren en la subred de gestión, aunque en la red del laboratorio 2L007 solamente se trabajará con

un OLT (al que se conectarán 4 ONUs). Gracias a su interfaz gráfica, sencilla y visual, permite una abstracción del protocolo GPON de manera que la configuración de servicios se realiza de forma muy intuitiva y flexible.

Antes de analizar con más detalle la interfaz gráfica del TGMS, es importante conocer los detalles que permiten su funcionamiento. El TGMS que proporciona TELNET Redes Inteligentes está instalado en una máquina virtual compatible con *VirtualBox* [20]. Para el correcto funcionamiento del TGMS, se deben configurar las interfaces de red en *VirtualBox* teniendo en cuenta que el TGMS debe ser capaz de conectarse a los OLTs de la red, y además debe ser accesible de manera local en el ordenador que ejerce de host de la máquina virtual [4].

Por tanto, para tener acceso al OLT desde el TGMS, se debe conectar el puerto de gestión *FastEthernet* del OLT (ver Figura 4) a la misma subred donde se encuentre el ordenador que ejerce de host de la máquina virtual de manera que deben tener direcciones IP (*Internet Protocol*) del mismo rango. En la Figura 18, se muestra la configuración de red necesaria para el funcionamiento del TGMS, con las direcciones IP correspondientes. Se trata de un esquema genérico que aparece en el manual [4]. El esquema de esta red GPON debería incluir también el *splitter*.

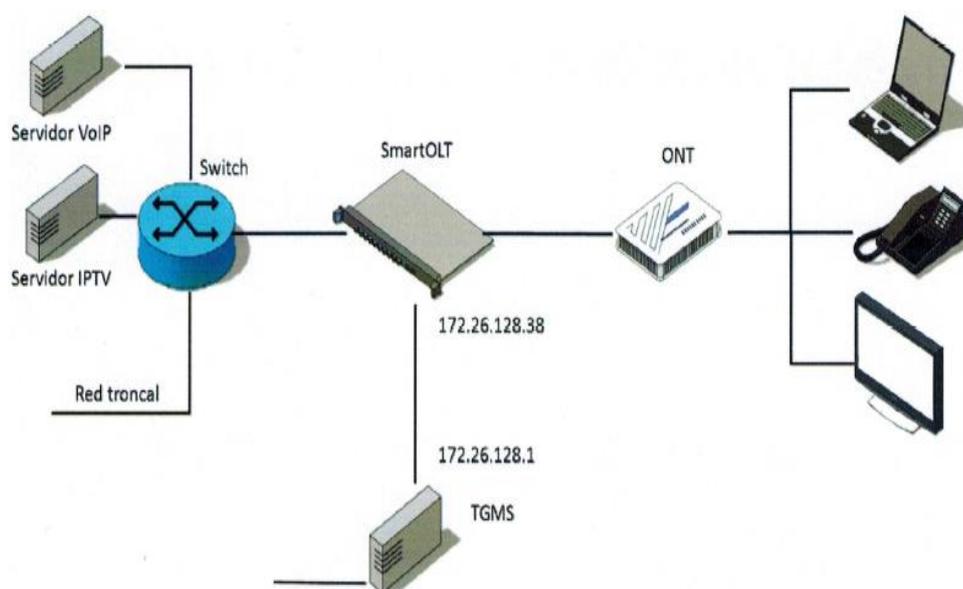


Figura 18: Configuración de red del TGMS

El OLT tiene como dirección IP la 172.26.128.38 (puede cambiarse) de manera que, en el ordenador en el que está la máquina virtual con el TGMS instalado, se ha configurado la dirección 172.26.128.1 en la interfaz que conecta con el OLT. De esta forma, ambos elementos forman parte de la misma subred, que se denominará subred de gestión. El siguiente paso es arrancar la máquina virtual desde *Oracle VM VirtualBox* [20]. Para ello, se debe seleccionar la máquina virtual de nombre TGMS y pulsar el botón Iniciar mostrado en la Figura 19. El proceso de arranque lleva unos 3-4 minutos.

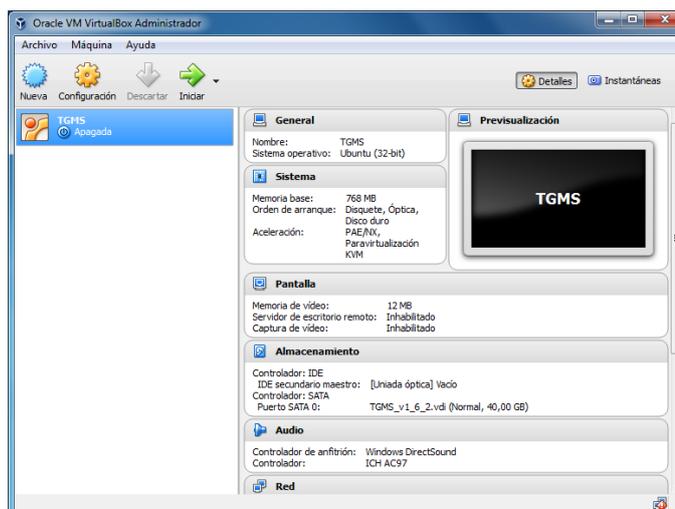


Figura 19: Arranque de la máquina virtual donde está instalado el TGMS

Una vez arrancada y operativa la máquina virtual, para acceder al TGMS basta con abrir el navegador (Firefox o Chrome) y acceder a una de estas dos IPs: <http://172.26.128.1> o <http://192.168.56.99> [4]. Al introducir estas direcciones, aparecerá el entorno web del TGMS (Figura 20), que pedirá las credenciales de acceso. El *Login* es *root* y el *Password* es *management*.

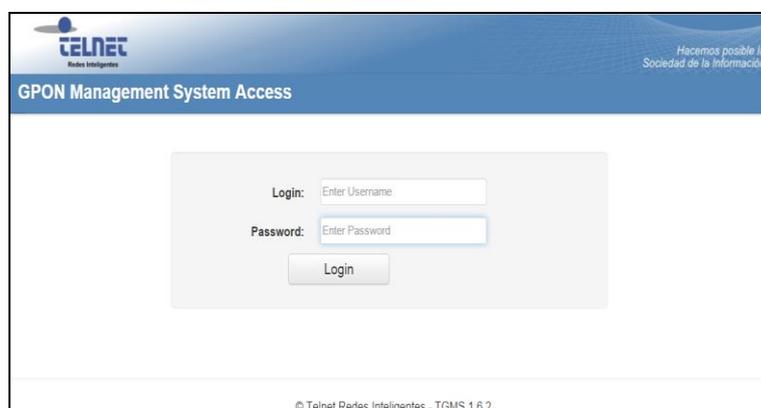


Figura 20: Acceso al TGMS

Introducidas las claves de acceso, aparece el menú principal del TGMS. A continuación, se hará una breve descripción de los menús que componen el TGMS, haciendo especial énfasis en aquellas características que sean relevantes para la creación de servicios. El TGMS está formado por seis menús: *Devices*, *Profiles*, *Subscribers*, *Alarms*, *Log* y *Configuration*. Se accede a cada menú pulsando en la pestaña correspondiente con el nombre del menú, que aparece en la parte superior de la interfaz del TGMS (ver Figura 21). Para conocer todos los detalles de cada menú, se debe acudir al manual del TGMS [4].

### 3.2.1 Menú *Devices*

En la pantalla principal del sistema (Figura 21), aparece el menú *Devices*, que por defecto muestra el panel de gestión de los OLTs.

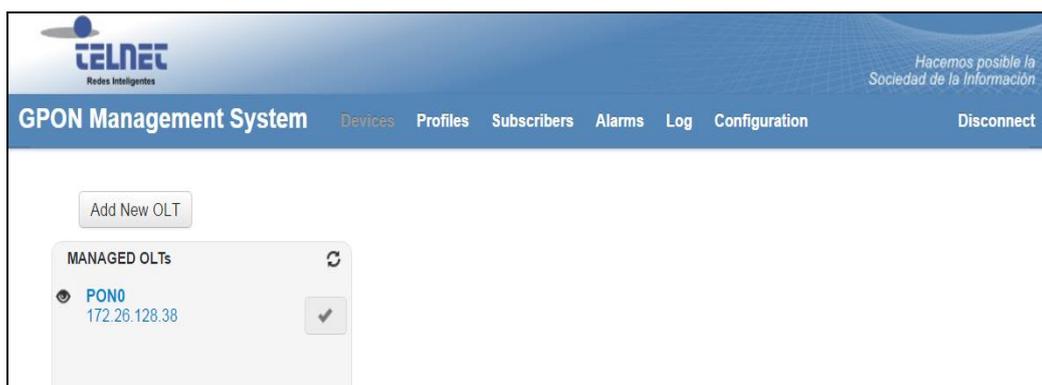


Figura 21: Pantalla principal del TGMS. Menú *Devices*

El panel de gestión de OLTs muestra la información más relevante de los OLTs gestionados por la red (en este caso solo uno) como se observa en la Figura 22.



Figura 22: Panel de gestión de los diferentes OLTs

Los posibles estados de conectividad son: hay conectividad, no hay conectividad y la información no está sincronizada. En este último caso, es necesario pulsar sobre el icono  para sincronizar la información entre TGMS y OLT. Las notificaciones posibles que pueden aparecer en el recuadro de dicho panel son: alarmas activas para una o más ONUs, existencia de ONUs deshabilitadas y errores de aprovisionamiento (o de configuración en las ONUs) [4]. En esta imagen, según la configuración de red existente, se observa que no aparece ninguna notificación.

Pulsando sobre este panel de gestión de OLTs (recuadro azul de la parte izquierda de la Figura 23), aparece el panel de gestión por puertos (parte central y derecha de la Figura 23). En la red bajo estudio, todas las ONUs están conectadas al puerto PON0. Este menú permite ver el estado de las ONUs conectadas a la red GPON. En este caso, las cuatro ONUs están registradas y activas, tal y como se observa en la parte de la derecha de la Figura 23.



Figura 23: Panel de gestión por puertos

Al pulsar en el botón *Details* del panel de gestión por puertos (parte inferior derecha en la Figura 23), se accede al panel de gestión de ONUs (Figura 24).

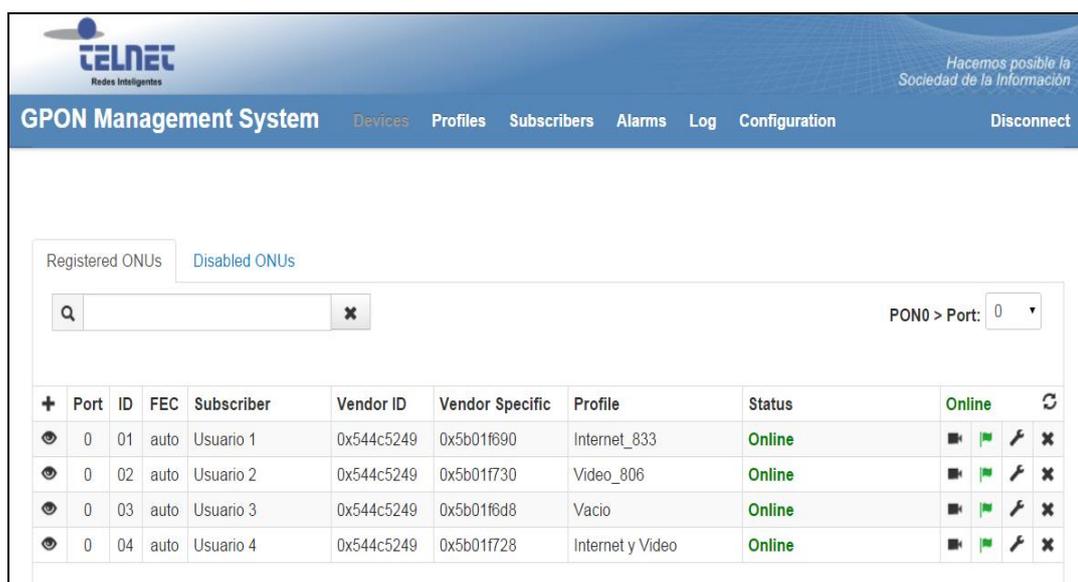


Figura 24: Panel de gestión de las ONUs conectadas a la red GPON

En la pestaña *Registered ONUs*, aparecen todas las ONUs habilitadas para conectarse al PON0 -> Port 0, que es el puerto en el que están conectadas las 4 ONUs de la red. Como se explicó en el Capítulo 2, cada puerto PON está conectado con un puerto de transporte (PON 0 con Port 0, y así sucesivamente). En la parte superior derecha, se puede cambiar el puerto seleccionado (0, 1, 2, 3) o mostrar todos (All). En el icono + , se puede añadir una nueva ONU y en el icono en forma de ojo 👁️ , consultar y/o modificar la configuración de una ONU ya registrada. Para eliminar una ONU, basta con pulsar en el icono de eliminar ✖️ situado a la derecha del todo. Para actualizar el la información de las ONUs, hay que pulsar el icono de refrescar 🔄 .

La información que se muestra sobre cada ONU es la siguiente [4]:

- **Port:** puerto en el que está registrada dicha ONU.
- **ID:** identificador de cada ONU (introducido por el usuario que manjea el TGMS y modificable). En este caso, las ONUs están numeradas con el número que aparece en la etiqueta que tienen puesta.
- **FEC:** indica si se utiliza corrección de errores FEC (*Forward Error Correction*).
- **Subscriber:** suscriptor al que está vinculado cada ONU.

- Vendor ID / Vendor Specific: son los identificadores de equipo. El campo *Vendor ID* es el mismo para las 4 ONUs. El campo *Vendor Specific* corresponde con últimos 8 valores hexadecimales de la dirección MAC de cada ONU. Esta dirección MAC aparece en la parte inferior de los equipos.
- Profile: nombre del perfil de servicios asignado.
- Status: estado de la ONU. Los estados más comunes que puede tener una ONU son los que se van a enumerar a continuación. Sin embargo, el resto de posibles estados vienen detallados en la página 26 del manual [4].
  - ✓ *Online*: la ONU está activa y los servicios han sido configurados correctamente.
  - ✓ *Never connected*: la ONU nunca ha sido conectada a la red y no se dispone de información sobre ella.
  - ✓ *Not connected*: la ONU no está conectada en ese momento, pero lo ha podido estar.
  - ✓ *Detected*: la ONU ha sido detectada y el sistema está comprobando su configuración. Suele producirse cuando la ONU acaba de ser conectada o cuando se ha realizado un cambio en la configuración.
  - ✓ *Disabled*: la ONU está desactivada. En este caso, aparecerá en la pestaña *Disabled ONUs*.
  - ✓ *OMCI Configuration Error*: error de comunicación con la OLT. Se suele producir cuando el nivel de potencia recibido no está entre los límites permitidos.
- Online: en este campo del interfaz, aparecen 3 iconos: acceso al monitor de estado , alarmas activas  y acceso al menú de configuración por puertos de la ONU . El monitor de estado de la ONU (Figura 25) aparece al hacer clic en el icono .

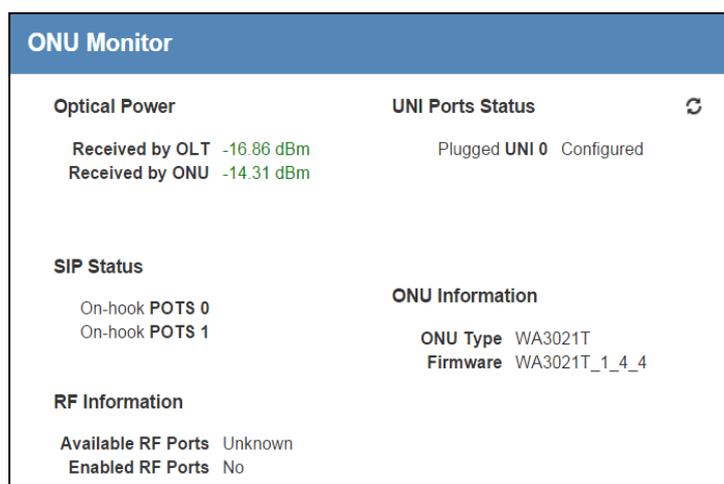


Figura 25: Monitor de estado de una ONU

La principal información que proporciona este menú es la potencia óptica recibida por el OLT (comunicación en el sentido de subida) y por la ONU (comunicación en el sentido de bajada). Esta información se analizó y se presentó en la Tabla 1 del Capítulo 2 para realizar una comprobación de los resultados teóricos y prácticos.

En el caso de haber alarmas activas, el icono  aparecerá en rojo. Este icono da acceso al menú de información de alarmas activas. Finalmente, el icono  da acceso al menú de configuración por puertos de la ONU (Figura 26). Este menú permite ver los servicios asignados a dicho ONT y en caso de vídeo y telefonía, configurarlos.

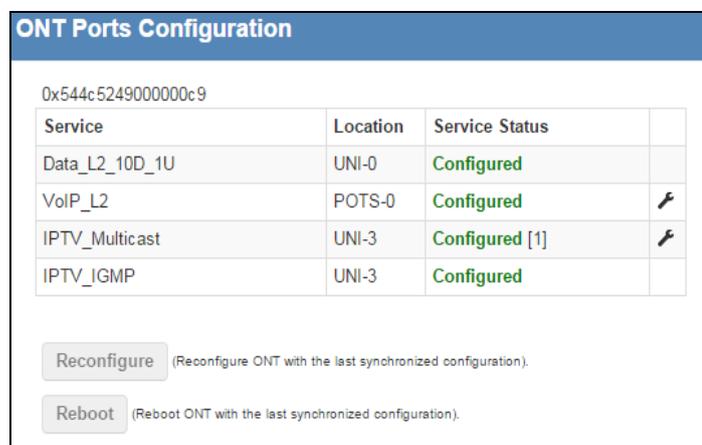


Figura 26: Menú de configuración por puertos de una ONU

Por otro lado, en el interfaz principal de la Figura 24, tanto si se desea añadir una nueva ONU ( + ), como si se quiere modificar una registrada ( 📡 ), aparece el menú de configuración de ONU, tal y como se muestra en la Figura 27.

The screenshot shows a window titled "ONU Configuration" with a close button in the top right corner. The window contains two columns of configuration fields:

- Left Column:**
  - ONU ID:** Text input field containing "01".
  - Vendor ID:** Text input field containing "0x544c5249".
  - Specific ID:** Text input field containing "0x5b01f690".
  - Signal Degraded Threshold:** Section header with a dropdown menu set to "Bit Error Rate (BER) - 10" and a value of "-5".
  - Signal Fail Threshold:** Section header with a dropdown menu set to "Bit Error Rate (BER) - 10" and a value of "-3".
- Right Column:**
  - FEC:** Dropdown menu set to "default".
  - RF PORT (If present):** Dropdown menu set to "default".
  - Profile:** Dropdown menu set to "Voz".
  - Subscriber:** Text input field containing "12345678H".
  - Lower Optical Threshold:** Section header with a dropdown menu set to "Default", a minus sign, a dash, and a unit of "dBm".
  - Upper Optical Threshold:** Section header with a dropdown menu set to "Default", a minus sign, a dash, and a unit of "dBm".

At the bottom right of the window, there are two buttons: "Close" and "Save".

Figura 27: Menú de configuración de ONU

En esta pantalla, se permite introducir (o modificar) el identificador de ONU y se selecciona el perfil y suscriptor asociados a la misma. En el caso de que la ONU se encuentre conectada a la PON y se vaya a habilitar, el campo *Specific ID* vendrá ya completo. Si se quiere configurar una ONU antes de conectarla a la red, se deberá introducir manualmente. Asimismo, también se pueden configurar los valores límites de la BER (*Bit Error Rate*) y potencia óptica, de forma que se genere una alarma si se sobrepasan estos límites. En caso de no configurar estos valores, se toman unos valores por defecto [4].

### 3.2.2 Menú Profiles

Volviendo a la pantalla principal del TGMS (Figura 21), se pasará a describir el menú *Profiles* (Figura 28). Este menú permite configurar la oferta de perfiles de abonado que se da a los clientes. Cada perfil de usuario (*Profile*) se compone de uno o varios servicios (voz, datos, vídeo, etc.). A su vez, cada uno de estos servicios tiene asignado un ancho de banda (*Bandwith Map*) y un mapeado de VLAN (*VLAN Map*). La forma de configurar anchos de banda, mapeados VLAN, servicios y perfiles se detallará en los Apartados 3.3 y 3.4.

+	Profile	FEC	RF	Service	Service Type	PPTP ID	Bandwidth Map	VLAN Map	6
👁	Internet y Video	No	No	Internet_833	Eth	UNI-0	BW_Internet	VLAN_Servidor_DHCP	✘
IPTV_Multicast_806				MC	UNI-0	BW_IPTV_Multicast	VLAN_IP_Estatica		
IPTV_IGMP_806				Eth	UNI-0	BW_IPTV_IGMP	VLAN_IP_Estatica		
👁	Internet_806	No	No	Internet_806	Eth	UNI-0	BW_Internet	VLAN_IP_Estatica	✘
👁	Internet_833	No	No	Internet_833	Eth	UNI-0	BW_Internet	VLAN_Servidor_DHCP	✘
👁	Vacio	No	No						✘
👁	Video_806	No	No	IPTV_Multicast_806	MC	UNI-0	BW_IPTV_Multicast	VLAN_IP_Estatica	✘
IPTV_IGMP_806				Eth	UNI-0	BW_IPTV_IGMP	VLAN_IP_Estatica		
👁	Video_833	No	No	IPTV_IGMP_833	Eth	UNI-0	BW_IPTV_IGMP	VLAN_Servidor_DHCP	✘
IPTV_Multicast_833				MC	UNI-0	BW_IPTV_Multicast	VLAN_Servidor_DHCP		

Figura 28: Apartado *Profiles* dentro del menú *Profiles*

Los servicios aparecen recogidos en el apartado *Services* (Figura 29). Cada servicio tiene un ancho de banda (apartado *Bandwidth Maps* - Figura 30) y un mapeado VLAN asociado (apartado *VLAN Maps* – Figura 31). En los apartados siguientes, se explicará cómo configurar estos parámetros para proporcionar servicio de datos y vídeo.

+	Service	Service Type	Bandwidth Map	VLAN Map	6
👁	Internet_806	Ethernet	BW_Internet	VLAN_IP_Estatica	✘
👁	Internet_833	Ethernet	BW_Internet	VLAN_Servidor_DHCP	✘
👁	IPTV_IGMP_806	Ethernet	BW_IPTV_IGMP	VLAN_IP_Estatica	✘
👁	IPTV_IGMP_833	Ethernet	BW_IPTV_IGMP	VLAN_Servidor_DHCP	✘
👁	IPTV_Multicast_806	Multicast	BW_IPTV_Multicast	VLAN_IP_Estatica	✘
👁	IPTV_Multicast_833	Multicast	BW_IPTV_Multicast	VLAN_Servidor_DHCP	✘

Figura 29: Apartado *Services* dentro del menú *Profiles*

Information				Downstream (Kbs)		Upstream (Kbs)		3
+	Bandwidth map	Flow Type	Status Reporting	BW	Allowed Excess	BW	Best Effort	
🔍	BW_IPTV_Multicast	Data	NSR	99968	49984	99968	49984	✘
🔍	BW_IPTV_IGMP	Data	NSR	99968	49984	99968	49984	✘
🔍	BW_Internet	Data	NSR	200000	99968	200000	99968	✘

Figura 30: Apartado *Bandwidth Maps* dentro del menú *Profiles*

+	VLAN	User - Priority	User - Tag	C - Priority	C - Tag	S - Priority	S - Tag	2
🔍	VLAN_IP_Estatica	Any	806	Copy	806	Untagged	Untagged	✘
🔍	VLAN_Servidor_DHCP	Any	833	Copy	833	Untagged	Untagged	✘

Figura 31: Apartado *VLAN Maps* dentro del menú *Profiles*

El apartado *VoIP Servers* no se utilizará en este estudio, ya que no se dispone de servidor de telefonía. Los apartados *Multicast Packs* y *Multicast Channels* se explicarán en el Apartado 3.4, y corresponderán al servicio de vídeo.

### 3.2.3 Menú *Subscribers*

El menú *Subscribers* (Figura 32) permite gestionar los usuarios de la GPON. Cada suscriptor tiene un identificador único y una descripción. Cada ONU debe tener suscriptor asociado como se ve en el menú de configuración de ONU (Figura 27).

+	Subscriber	Description	4
👁	Usuario 1	Usuario 1	✖
👁	Usuario 2	Usuario 2	✖
👁	Usuario 3	Usuario 3	✖
👁	Usuario 4	Usuario 4	✖

Figura 32: Menú *Suscribers*

### 3.2.4 Menú Alarms

El menú *Alarms* (Figura 33) permite consultar las alarmas: tanto el historial (pestaña *Alarm History*) como las que se encuentran activas (pestaña *Active Alarms*). Además, en el apartado *Event Log* se registran todas las alarmas y todos los cambios de estado que se producen en las ONUs.

OLT		ONT		Activation	Deactivation	Active Alarm					
ID	Dev	Port	ID	Serial Number	Day	Time	Day	Time	Alarm	Freq.	Code
SmartOLT_1 00:09:58:dd:00:00	-	-	?	0x0000000000000000	2015-01-20	12:50:37	2015-01-20	12:51:37	Test Alarm	>1/min	●
SmartOLT_1 00:09:58:dd:00:00	-	-	?	0x0000000000000000	2015-01-20	12:28:00	2015-01-20	12:29:00	Test Alarm	>1/min	●
SmartOLT_1 00:09:58:dd:00:00	0	3	0	0x544c524900000599	2015-01-20	10:36:48	2015-01-20	10:37:48	Loss of GEM Channel Delineation	>1/min	●

Figura 33: Menú *Alarms*

### 3.2.5 Menú Log

En el menú *Log* (Figura 34) se visualiza el registro de todas las acciones realizadas en el TGMS. Se clasifican en 4 categorías:

- Provisioning: todo lo relacionado con la creación, modificación y/o eliminación de un perfil (anchos de banda, VLANs, servicios, etc.).
- OLT Configuration: cambios en el *firmware* y en el número de serie de la OLT.

- OLT Sync: sincronización entre OLT y TGMS.
- Configuration: relativo a la gestión de usuarios.

Time	Event	Type
2015-01-20 13:04:35	CONFIGURATION SENT TO OLT 'SmartOLT_1' located in 10.10.169.172    'root' [master]	OLT Sync
2015-01-20 13:04:30	MODIFIED Multicast Provision for SmartOLT_1 > Dev: 0 > chn: 3 > UNI-0 > IPTV_Multicast    'root' [master]	Provisioning
2015-01-20 13:04:18	MODIFIED ONT '0' (0) in OLT:SmartOLT_1 > dev:0 > ch:3    'root' [master]	Provisioning
2015-01-20 13:00:37	CONFIGURATION SENT TO OLT 'SmartOLT_1' located in 10.10.169.172    'root' [master]	OLT Sync
2015-01-20 13:00:29	MODIFIED ONT '0' (0) in OLT:SmartOLT_1 > dev:0 > ch:3    'root' [master]	Provisioning
2015-01-20 12:59:42	CONFIGURATION SENT TO OLT 'SmartOLT_1' located in 10.10.169.172    'root' [master]	OLT Sync

Figura 34: Menú *Log*

### 3.2.6 Menú *Configuration*

Finalmente, el menú *Configuration* (Figura 35) es el menú de configuración del TGMS (no confundir con la configuración de la GPON). Sirve para crear diferentes usuarios del TGMS (en este caso, solo hay una cuenta del usuario llamada *root*), gestionar la seguridad del TGMS y acceder al menú de configuración avanzada *Server Configuration*, que cuenta con una interfaz diferente. En este menú *Server Configuration* es donde se actualizan las versiones del TGMS y el *firmware* de OLTs y ONUs. Asimismo, también permite configurar la hora, la fecha, el huso horario, etc. Los detalles sobre este menú de configuración avanzada se encuentran en las páginas 34 a 37 del manual [4].

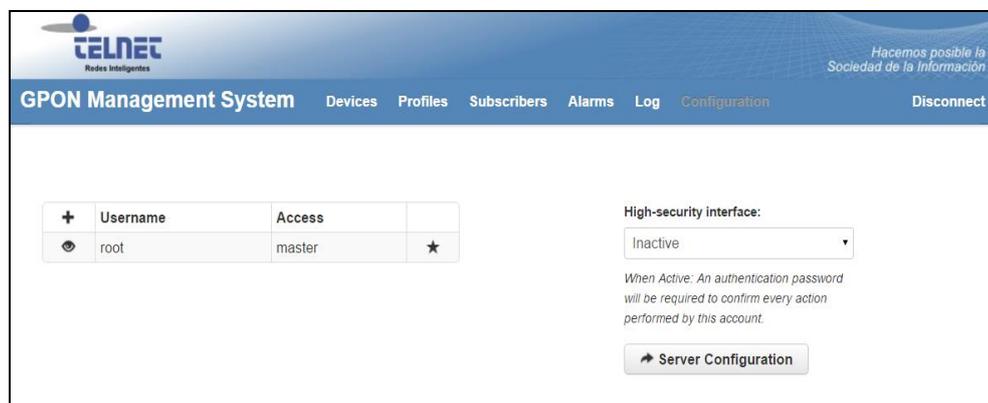


Figura 35: Menú *Configuration*

### 3.3 Configuración del servicio de Internet

La principal función de toda red de acceso es proporcionar servicio a los clientes. En este apartado, se enumerarán los pasos que hay que seguir para proporcionar servicio de datos (Internet) a los usuarios de la red GPON bajo estudio utilizando el TGMS. En primer lugar, se explicará la configuración que se debe asignar a los elementos de la red GPON (OLT y ONUs) a través del TGMS. Posteriormente, se indicarán los pasos a seguir para configurar los routers de cada ONU y finalmente, se mostrarán algunos resultados interesantes sobre pruebas realizadas con el servicio de Internet.

#### 3.3.1 Configuración del servicio de Internet en la interfaz del TGMS

Para configurar servicios de datos (Internet) a través del TGMS, los pasos a seguir son los siguientes [4]:

1. Creación de un mapeado VLAN: se llevará a cabo en el icono  del apartado *VLAN Maps* dentro del menú *Profiles* (Figura 31). El protocolo GPON [3] funciona con etiquetado VLAN, descrito en el protocolo IEE 802.1Q [21].

Una VLAN (*Virtual Local Area Network*) o red de área local virtual es un grupo de dispositivos que forman una red lógica sin necesidad de compartir el mismo medio físico. Las VLANs permiten reducir las limitaciones que introducen las arquitecturas físicas y crear redes flexibles en las que es sencillo añadir o eliminar usuarios [22].

La nomenclatura utilizada en el TGMS [4] para las VLAN, basada en el protocolo IEE 802.1Q [21], es la siguiente (ver Figura 31):

- *S-Tag*: etiqueta de servicio. Se le debe asignar un valor cuando se emplee doble etiquetado *QinQ* [21]. Solamente se usará cuando lleguen servicios de varios operadores por lo que en esta red, se utilizará solamente etiquetado simple.

- *C-Tag*: etiqueta de cliente. Es la correspondiente al etiquetado sencillo en la red de transporte y la red GPON y se debe especificar siempre.
- *User-Tag*: etiqueta de usuario. Corresponde al etiquetado sencillo a la salida de la ONU y sólo se deberá usar cuando los equipos conectados estén configurados para usar VLAN.

En la Tabla 2, se muestra un resumen del uso de estas etiquetas y las modificaciones que sufren al pasar por el OLT y por las ONUs en los dos sentidos de la comunicación.

	Al pasar por el OLT	Al pasar por la ONU
Sentido de bajada ( <i>Downstream</i> )	Se elimina la etiqueta <i>S-Tag</i> correspondiente al doble etiquetado (si la hay) y se mantiene la etiqueta simple <i>C-Tag</i> . En caso de llegar al OLT tráfico no etiquetado o con etiqueta incorrecta, se descarta.	El tráfico puede seguir etiquetado (en cuyo caso la etiqueta simple pasará a llamarse <i>User-Tag</i> ) o pasar a ser tráfico desetiquetado en caso de que vaya a un equipo final que provea un solo tipo de servicio.
Sentido de subida ( <i>Upstream</i> )	Se mantiene la etiqueta simple <i>C-Tag</i> y se puede añadir el doble etiquetado en caso de que sea necesario en la red de transporte.	Se puede marcar el tráfico desetiquetado o añadir la etiqueta simple <i>C-Tag</i> . En caso de que las ONUs reciban tráfico con etiquetado diferente al configurado, se descartará.

Tabla 2: Etiquetado VLAN en los elementos de la red GPON

La etiqueta VLAN se compone de dos campos: identificador (*Tag*) y prioridad (*Priority*). Los identificadores van desde el 1 hasta el 4094 mientras que las prioridades van de 0 a 7 [21]. Sin embargo, no todos hacen que la red esté operativa. Se deben utilizar identificadores VLAN que permitan que las ONUs reciban direcciones IP. Para conocer estos identificadores válidos para la red GPON situada en el laboratorio 2L007, se ha consultado al técnico superior en Redes y Comunicaciones de la Escuela Técnica Superior de Ingenieros de Telecomunicación de Valladolid [1].

Los dos posibles identificadores VLAN son el 806, que es el utilizado por los ordenadores del laboratorio 2L007, y el 833, que conecta con un servidor DHCP (*Dynamic Host Configuration Protocol*) [23] que asigna de forma dinámica las direcciones IP a las ONUs. Para el primer caso, las direcciones IP se deberán introducir manualmente en el menú del router de la ONU (se explicará en el Apartado 3.3.2).

Estos son los campos que hay que completar en el TGMS [4] para crear el mapeado VLAN:

- ✓ *VLAN\_ID*: nombre utilizado para el mapeado VLAN. Se utilizará *VLAN\_IP\_Estática* para el identificador 806 y *VLAN\_Servidor\_DHCP* para el 833.
- ✓ *User-Priority / User-Tag*: se marcará *Any* como prioridad para mantener el valor y como *User-Tag*, se utilizará el 806 para *VLAN\_IP\_Estática* y el 833 para *VLAN\_Servidor\_DHCP*.
- ✓ *C-Priority / C-Tag*: se marca *Copy* como prioridad de forma que se aceptará cualquier prioridad que venga del exterior de la red y *el C-Tag* será el mismo que el *User-Tag*.
- ✓ *S-Priority / S-Tag*: no se utiliza doble etiquetado de forma que estos campos serán marcados como *Untagged*.

La Figura 36 muestra un ejemplo de configuración de mapeado VLAN. Una vez completados todos los campos, se deberá pulsar  para crear el mapeado.

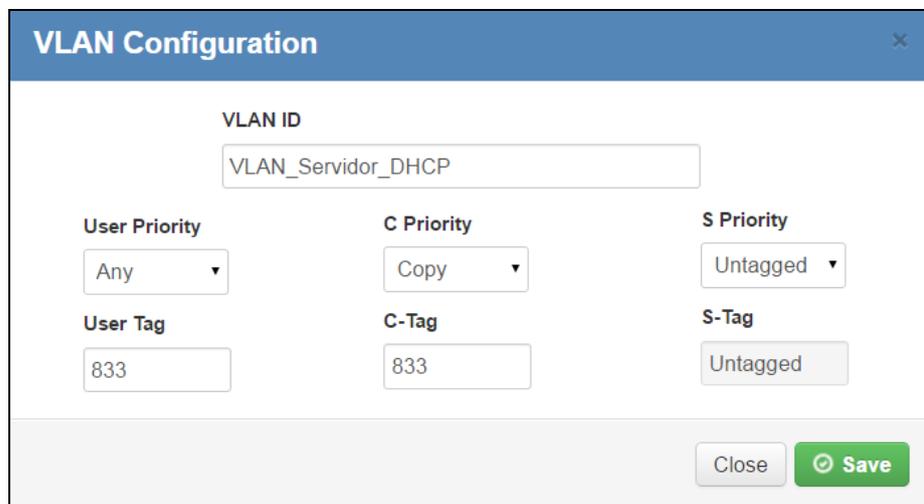


Figura 36: Ejemplo de mapeado VLAN

2. Creación de un mapa de ancho de banda: se llevará a cabo en el icono  del apartado *Bandwidth Maps* dentro del menú *Profiles* (Figura 32). La comunicación en la red GPON se produce en dos sentidos: bajada (*Downstream*), con una tasa de 2,488 Gbps y subida (*Upstream*), con una tasa de 1,244 Gbps [3]. Cada flujo debe tener un ancho de banda garantizado y un ancho de banda en exceso en ambos sentidos.

Para no exceder estas tasas de datos, el TGMS controla los anchos de banda asignados. En primer lugar, el ancho de banda garantizado, en cualquier sentido de la comunicación, será redondeado a múltiplos de 64 Kbps [3] y no debe exceder la tasa máxima (el TGMS lanza un mensaje de error si se excede). Los anchos de banda en exceso pueden superar esta tasa y el algoritmo DBA (*Dynamic Bandwidth Allocation*) empleado será el que reparta el ancho de banda en exceso [24]. Los detalles de este algoritmo de asignación dinámica de ancho de banda no son accesibles desde el TGMS.

En segundo lugar, realiza una comprobación de ancho de banda a la hora de asignar perfiles a las ONUs. Si al añadir un nuevo perfil de usuario a

una ONU se supera la tasa máxima en alguno de los dos sentidos de la comunicación, el TGMS mostrará un mensaje de error y no permitirá esta asignación.

Los campos que se deben completar para crear un mapa de ancho de banda son los siguientes [4]:

- ✓ *Bandwidth map*: nombre del mapa de ancho de banda.
- ✓ *Flow type*: tanto para servicios de datos como de vídeo, el tipo de flujo serán datos por lo que se seleccionará *Data*. *VoIP* se utilizará para el servicio de telefonía.
- ✓ *Status Reporting*: tipo de algoritmo DBA empleado: NSR (*Non-Status Reporting*) o SR (*Status Reporting*) *Type 0* (DBRu) o *Type 1* (DBRu). Se utilizará en la mayoría de los casos NSR. En ambos casos, es el OLT el que realiza la asignación dinámica de ancho de banda pero en el NSR, las ONUs no envían reportes del estado de sus colas [24] de forma periódica al OLT.
- ✓ *Downstream (Kbs) BW / Allowed Excess*: ancho de banda garantizado (BW) y en exceso (*Allowed Excess*) para el sentido de bajada. En ambos casos, el valor introducido se redondeará a 64 Kbps y se ha de tener en cuenta que la tasa máxima en el sentido de bajada son 2,488 Gbps [3].
- ✓ *Upstream (Kbs) BW / Best Effort*: anchos de banda garantizado (BW) y *Best Effort* (equivalente en este caso al ancho de banda en exceso en el *Downstream*). En ambos casos, el valor introducido se redondeará a 64 Kbps y se ha de tener en cuenta que la tasa máxima en el sentido de subida son 1,244 Gbps [3].

La Figura 37 muestra un ejemplo de configuración de un mapa de ancho de banda. Una vez completados todos los campos, se deberá pulsar  para crear el mapa de ancho de banda.

Figura 37: Ejemplo de configuración de un mapa de ancho de banda

3. Creación de un servicio Ethernet: se llevará a cabo en el icono **+** del apartado *Services* del menú *Profiles* (Figura 29). El servicio se define a través de cuatro campos [4]:

- ✓ *Service ID*: nombre del servicio (dado por nosotros).
- ✓ *Service Type*: el servicio puede ser de tipo Ethernet, *Multicast* (vídeo) o SIP (telefonía). Para crear un servicio de datos (Internet), se debe seleccionar el tipo Ethernet.
- ✓ *Bandwidth Map*: se seleccionará uno de los mapas de ancho creados anteriormente.
- ✓ *VLAN Map*: se seleccionará uno de los mapeados VLAN creados anteriormente.

La Figura 38 muestra un ejemplo de configuración de un servicio Ethernet de datos. Una vez completados todos los campos, se deberá pulsar **Save** para crear el servicio.

**Service Configuration**

Service ID:

Service Type:

Bandwidth Map Listing:

VLAN Map:

Information		DS (Kbps)		US (Kbps)	
Type	Report	BW	Excess	BW	BE
Data	NSR	149952	49984	149952	99968

User		C		S	
Prio	Tag	Prio	Tag	Prio	Tag
Any	833	Copy	833	Untagged	Untagged

Close Save

Figura 38: Ejemplo de creación de un servicio de datos Ethernet

4. Creación y agrupación del servicio (o servicios) en un perfil de usuario: se llevará a cabo en el icono **+** del apartado *Profiles* del menú *Profiles* (Figura 28). El perfil tendrá un nombre (ONU Profile ID) y se debe seleccionar si se desea activar la corrección de errores FEC y si se desea activar la salida de radiofrecuencia (RF) de las ONUs [4]. Estos campos se mantendrán desactivados.

Se añadirán tantos servicios al perfil como se quiera. En este caso de ejemplo, se va a proporcionar al usuario un servicio de Internet, de manera que se seleccionará en el campo *Service* el servicio o servicios creados anteriormente (no tiene mucho sentido añadir más de 2 servicios puesto que solamente hay servicio en 2 identificadores VLAN). Una vez seleccionado, se pulsará el icono **+** para añadir el servicio al perfil y al pulsar **Save** se creará servicio. La Figura 39 muestra un ejemplo de creación de un perfil de usuario.

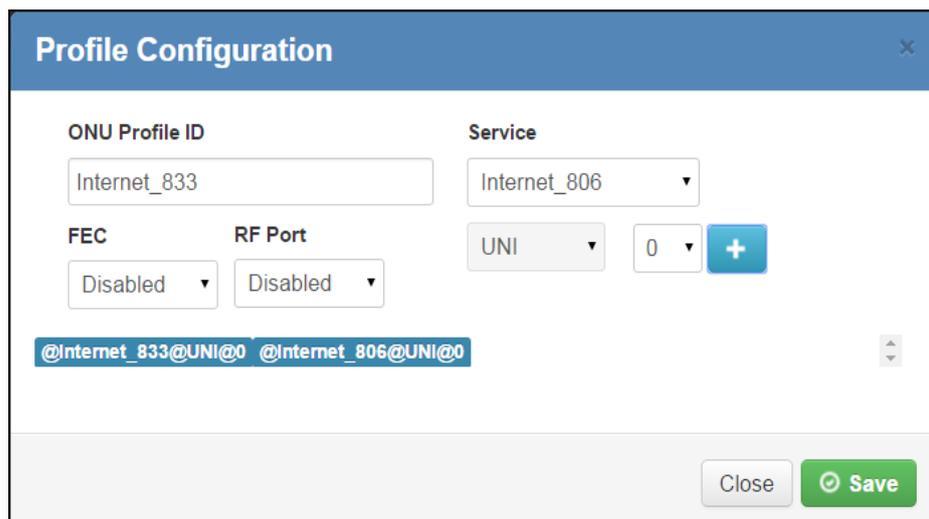


Figura 39: Ejemplo de creación de un perfil de usuario con un servicio de datos (Internet)

5. Asignación del perfil de usuario a las ONUs: finalmente, se asignará el perfil correspondiente a la ONU (u ONUs) desde el menú de configuración de ONU (Figura 27). Al desplegar el campo *Profile*, se mostrarán todos los perfiles disponibles. Tras seleccionar el perfil deseado, se hará clic en el botón . Posteriormente, en la pantalla principal del TGMS, será necesario sincronizar la información entre TGMS y OLT. Pulsando sobre el icono  se produce la sincronización (el proceso tarda unos segundos) como se ve en la Figura 40. Este proceso es muy importante pues de no producirse, los cambios del TGMS no se volcarán a la red.

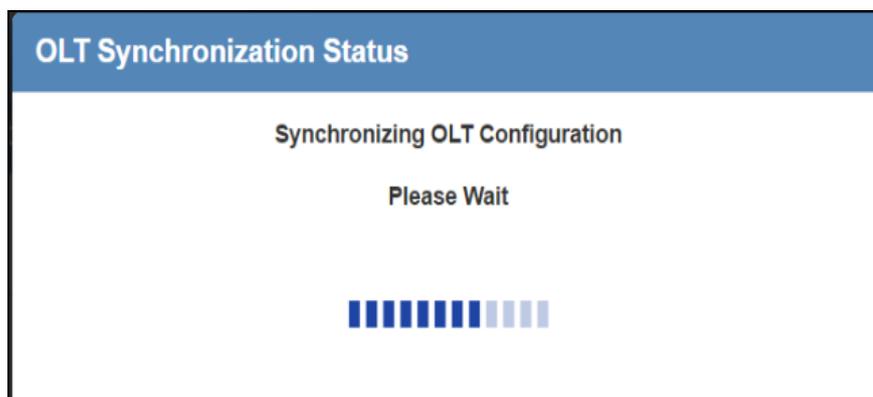


Figura 40: Sincronización entre el TGMS y el OLT cuando se actualiza la configuración

### **3.3.2 Configuración del servicio de Internet en los routers de las ONUs**

Una vez configurado el TGMS, es necesario configurar el router integrado en las ONUs (son de nivel 3, modelo *WaveAccess 3021*) [12] para disponer de servicio de Internet. La red dispone de 4 ONUs que tendrán un identificador de ONU de 1 a 4 según la etiqueta que tienen colocada encima (el identificador se puede cambiar en el panel de gestión de ONUs (Figura 24):

- *ONU 01*: tiene como dirección MAC 78:3d:5b:01:f6:90 y está conectada al ordenador 7 del laboratorio vía cable Ethernet.
- *ONU 02*: tiene como dirección MAC 78:3d:5b:01:f7:30 y está conectada al ordenador 10 del laboratorio vía cable Ethernet.
- *ONU 03*: tiene como dirección MAC 78:3d:5b:01:f6:d8 y está conectada al ordenador 14 del laboratorio vía cable Ethernet.
- *ONU 04*: tiene como dirección MAC 78:3d:5b:01:f7:28 y está conectada al ordenador 18 del laboratorio vía cable Ethernet.

En primer lugar, es necesario conectarse a una ONU. La conexión a la ONU puede realizarse vía cable Ethernet desde los ordenadores del laboratorio anteriormente indicados o vía Wifi [12]. En el icono de conexiones inalámbricas, aparecerán las 4 ONUs identificadas como *TELNET\_XXXX*. Estos últimos caracteres serán los cuatro últimos caracteres de la dirección MAC como se observa en la Figura 41. Para conectarse a una de estas ONUs, basta con seleccionar el nombre de la red inalámbrica de la lista desplegable e introducir la clave de autenticación. Existen varios métodos de autenticación, tales como *Open* (sin autenticación), *Shared*, 802.1x, WPA, WPA-PSK, WPA2, WPA2-PSK, Mixed WPA2/WPA, Mixed WPA2/WPA-PSK. Por defecto, en nuestro caso, se utilizará WPA2-PSK [13] y la clave de autenticación se encuentra en la parte inferior de las ONUs.



Figura 41: Conexión inalámbrica a las ONUs vía Wifi

Una vez se realice la conexión a la ONU, es necesario acceder a la interfaz de usuario Web. Para ello, basta con ir a la dirección IP 192.168.1.1 en el navegador e introducir las claves de acceso, es decir, el nombre de usuario (1234) y la contraseña (1234) [13]. Al acceder a esta interfaz, aparece el menú *Device Info* que muestra la información general. La Figura 42 muestra la pantalla principal de la interfaz.

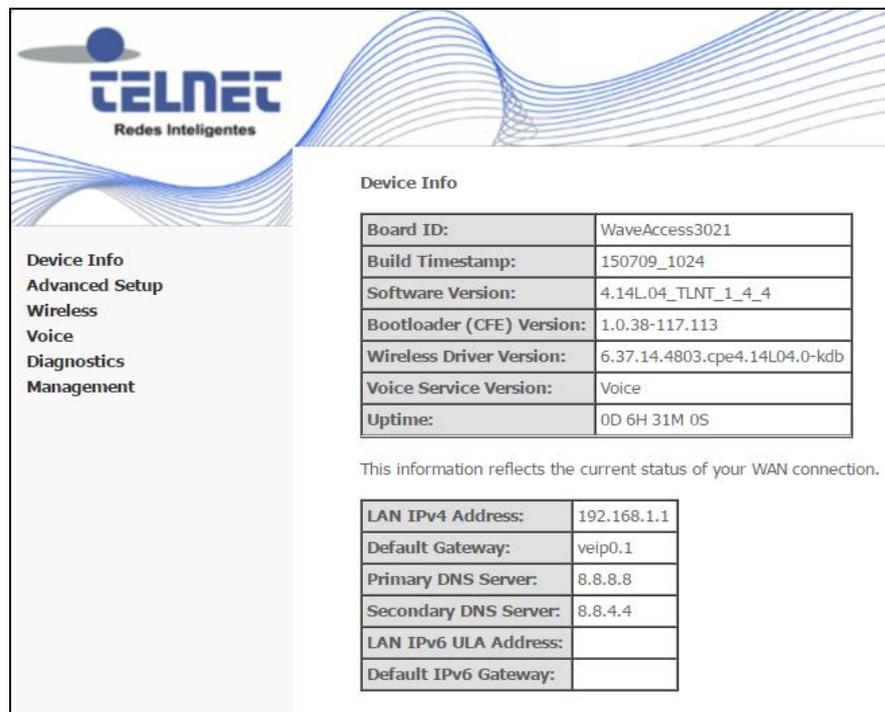


Figura 42: Interfaz de usuario Web de los routers de las ONUs

En el manual de usuario avanzado de la ONU [13], viene información sobre los diferentes menús que aparecen en la parte izquierda de la Figura 42. En este estudio, se analizará en profundidad el apartado *WAN Service* dentro del menú *Advanced Setup*. En este menú, se crearán las interfaces WAN (*Wide Area Network*) necesarias para disponer de servicio de Internet. Los pasos para crear una interfaz WAN son [13]:

1. En el apartado *WAN Service* dentro del menú *Advanced Setup*, se pulsa el botón *Add* para añadir una interfaz WAN.
2. Se selecciona la *WAN Service Interface Configuration*. Solo hay una posible y es la denominada *veip0/veip0*.
3. Se selecciona el tipo de servicio: PPPoE (*Point to Point over Ethernet*), *IP over Ethernet* o *Bridging*. Se utilizará la configuración *IP over Ethernet* ya que para utilizar PPPoE se requiere de un servidor especial.

En esta misma pantalla, se debe marcar la prioridad (entre 0 y 7) y el identificador VLAN (806 u 833, como se explicó anteriormente). También se debe seleccionar el VLAN TPID (*Tag Protocol Identifier*), que en este caso será 0x8100 al utilizar etiquetado VLAN 802.1Q [21] y especificar que solo se utilizarán direcciones IPv4.

4. En la pantalla *WAN IP Settings*, se debe seleccionar la forma de asignar la dirección IP a la ONU.

En caso de que se haya utilizado el identificador VLAN 833, se marcará la opción *Obtain an IP address automatically*, de modo que el servidor DHCP asigne de forma automática la configuración IP a la interfaz WAN [23]. Por otro lado, si se ha utilizado el identificador 806, se ha de seleccionar la opción de "*Use the following Static IP Address*" y rellenar los siguientes campos:

- *WAN IP Address*: dirección IP de la interfaz WAN para la conexión de datos de esa ONU. Se utilizarán direcciones IP de la forma 157.88.128.XXX, que son las utilizadas por el resto de

ordenadores del laboratorio 2L007. Para evitar utilizar la IP de otro equipo, los tres últimos dígitos estarán entre 240 y 248.

- *WAN Subnet Mask*: Máscara de la subred, que será la 255.255.255.0.
  - *WAN gateway IP Address*: dirección IP del gateway en la red de transporte con el que se comunican las ONUs, que será la 157.88.128.250. Para averiguar esta dirección, se ha ejecutado el comando *ipconfig* en un equipo del laboratorio.
5. En la siguiente pantalla, se debe activar la opción *Enable NAT*. NAT (*Network Address Translation*) permite traducir entre direcciones IP privadas y públicas [25]. La ONU tendrá una dirección IP pública (la introducida anteriormente o la proporcionada por el servidor DHCP) y una privada (192.168.1.1). Los dispositivos que se conecten a la ONU, ya sea mediante cable Ethernet o vía Wifi, recibirán una dirección IP privada del tipo 192.168.1.X, donde X irá desde 2 en adelante. Esta dirección la reciben gracias al servidor DHCP [23] que se encuentra dentro del router integrado de la ONU [12].
  6. Posteriormente, se seleccionará la interfaz de salida que será la que se acaba de crear. El nombre de la misma será *veip0.X*, donde X va de 1 en adelante.
  7. En este punto, se seleccionan los servidores DNS (*Domain Name System*). Se utilizará la opción *Use the following Static DNS IP address* tomando como servidor DNS primario (*Primary DNS server*) 8.8.8.8 y servidor DNS secundario (*Secondary DNS server*) 8.8.4.4. Estos servidores DNS son puestos por Google a disposición de los usuarios.
  8. Finalmente, aparecerá un resumen con la configuración seleccionada. Al seleccionar la opción *Apply/Save* se crea la interfaz WAN.

Desde el apartado *WAN Service* dentro del menú *Advanced Setup* también se pueden ver las interfaces WAN creadas con sus principales parámetros de configuración. Para esta red, lo adecuado es crear dos interfaces WAN: una para

el identificador VLAN 806 y otra para el identificador 833 como se ve en la Figura 43.

**Wide Area Network (WAN) Service Setup**

Choose Add, Remove or Edit to configure a WAN service over a selected interface.

Interface	Description	Type	Vlan8021p	VlanMuxId	VlanTpid	Igmp Proxy	Igmp Source	NAT	Firewall	IPv6	Mld Proxy	Mld Source	Remove	Edit
veip0.1	ipoe_veip0.833	IPoE	7	833	0x8100	Enabled	Enabled	Enabled	Disabled	Disabled	Disabled	Disabled	<input type="checkbox"/>	Edit
veip0.2	ipoe_veip0.806	IPoE	7	806	0x8100	Enabled	Enabled	Enabled	Disabled	Disabled	Disabled	Disabled	<input type="checkbox"/>	Edit

Figura 43: Interfaces WAN creadas con sus parámetros de configuración

El último paso para poder disponer de Internet en los dispositivos conectados a la ONU es reconfigurar la ONU con la última información sincronizada entre TGMS y OLT (lo que se hizo en el Apartado 3.3.1). Para ello, se debe acudir al menú de configuración por puertos de la ONU y pulsar el botón *Reconfigure (Reconfigure ONU with the last synchronized configuration)* [4]. De esta forma, se volcará a la ONU la información (ya sincronizada) del TGMS y OLT.

Una vez realizado esto, la interfaz o interfaces WAN cuyos identificadores VLAN tengan servicios de Internet asociados en el TGMS pasarán al estado de *Connected* en el apartado WAN del menú *Device Info* [13] y el cliente podrá disponer del servicio de Internet, con los parámetros de configuración recogidos en el TGMS.

En el ejemplo de la Figura 44, se observa como la interfaz WAN es la única con *Status Connected*. Esto significa que el servicio de Internet configurado en el TGMS tenía como identificador VLAN el 833. La dirección IP asignada por el servidor DHCP es la 10.0.103.115, tal y como se observa en dicha figura.

**WAN Info**

Interface	Description	Type	VlanMuxId	IPv6	Igmp Pxy	Igmp Enbl	MLD Pxy	MLD Src Enbl	NAT	Firewall	Status	IPv4 Address	IPv6 Address
veip0.1	ipoe_veip0.833	IPoE	833	Disabled	Enabled	Enabled	Disabled	Disabled	Enabled	Disabled	Connected	10.0.103.115	
veip0.2	ipoe_veip0.806	IPoE	806	Disabled	Enabled	Enabled	Disabled	Disabled	Enabled	Disabled	Unconfigured	157.88.128.242	

Figura 44: Información del estado de las interfaces WAN

Si las dos interfaces estuvieran conectadas, significaría que el perfil de usuario del TGMS asociado a la ONU tiene dos servicios de Internet, uno para cada identificador VLAN (Figura 45).

WAN Info

Interface	Description	Type	VlanMuxId	IPv6	Igmp Pxy	Igmp Enbl	MLD Pxy	MLD Src Enbl	NAT	Firewall	Status	IPv4 Address	IPv6 Address
veip0.1	ipoe_veip0.833	IPoE	833	Disabled	Enabled	Enabled	Disabled	Disabled	Enabled	Disabled	Connected	10.0.103.115	
veip0.2	ipoe_veip0.806	IPoE	806	Disabled	Enabled	Enabled	Disabled	Disabled	Enabled	Disabled	Connected	157.88.128.242	

Figura 45: Dos interfaces WAN con *Status Connected*

En este caso, habría que seleccionar cuál de las dos interfaces WAN va a ser la utilizada. Esto se hace en el subpartado *Default Gateway* dentro del apartado *Routing* del menú *Advanced Setup*. En la Figura 46, se observa cómo se selecciona la interfaz *veip0.2*, asociada al identificador VLAN 806.

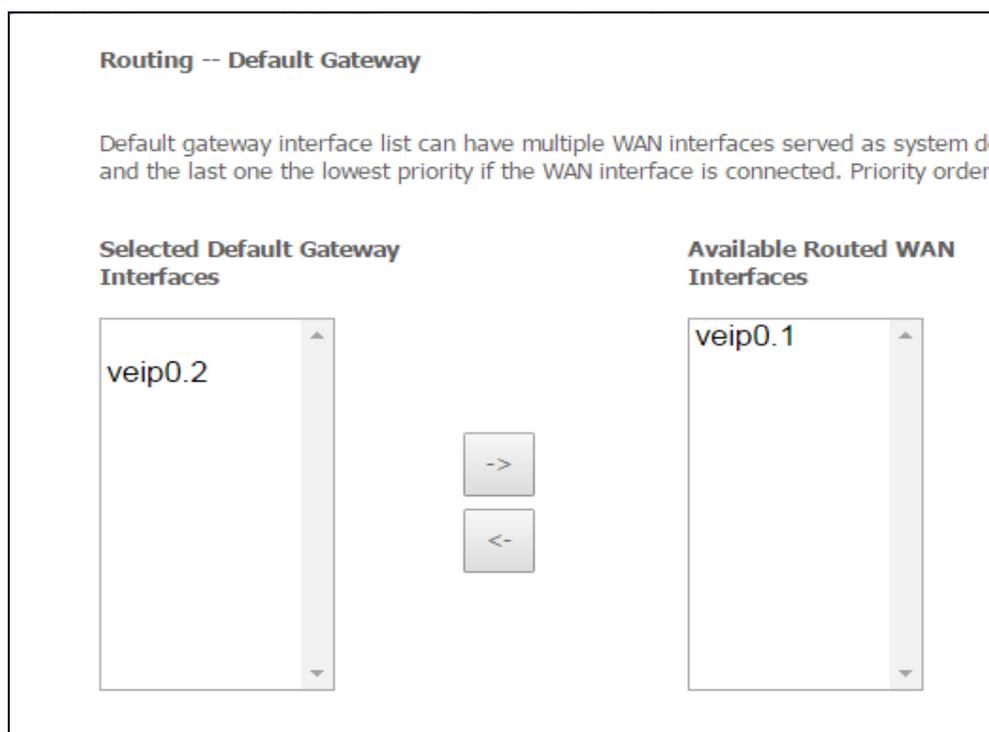
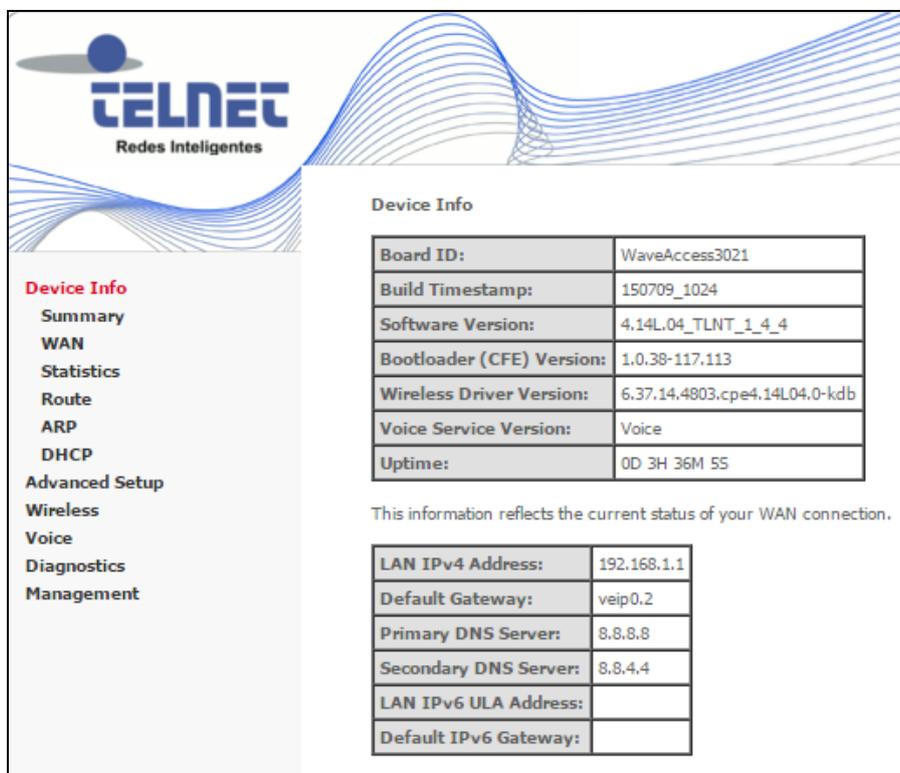


Figura 46: Selección de la interfaz WAN utilizada

Para comprobar cuál es la interfaz WAN que está siendo utilizada como salida por defecto, basta con acudir a la pantalla principal del interfaz web de usuario del router [13]. En esta pantalla principal, se recoge la información principal del router. El campo *Default Gateway* (ver Figura 47) indica el nombre de la interfaz que está siendo utilizada

como salida por defecto. Como se mostraba en la Figura 46, la interfaz activa es la *veip0.2*.



The screenshot shows the Telnet router configuration interface. The top left features the Telnet logo with the tagline 'Redes Inteligentes'. A navigation menu on the left lists various configuration sections: Device Info (highlighted), Summary, WAN, Statistics, Route, ARP, DHCP, Advanced Setup, Wireless, Voice, Diagnostics, and Management. The main content area is titled 'Device Info' and contains two tables. The first table lists hardware and software details, and the second table, titled 'This information reflects the current status of your WAN connection.', lists network parameters.

Device Info	
Board ID:	WaveAccess3021
Build Timestamp:	150709_1024
Software Version:	4.14L.04_TLNT_1_4_4
Bootloader (CFE) Version:	1.0.38-117.113
Wireless Driver Version:	6.37.14.4803.cpe4.14L04.0-kdb
Voice Service Version:	Voice
Uptime:	0D 3H 36M 55

This information reflects the current status of your WAN connection.

LAN IPv4 Address:	192.168.1.1
Default Gateway:	veip0.2
Primary DNS Server:	8.8.8.8
Secondary DNS Server:	8.8.4.4
LAN IPv6 ULA Address:	
Default IPv6 Gateway:	

Figura 47: Información general del router

### 3.3.3 Pruebas y resultados de la configuración de un servicio de Internet

Una vez configurados el TGMS y el router de la ONU y sincronizada la información entre ambos y el OLT, el cliente puede disponer de servicio de Internet.

El protocolo GPON [3] permite obtener tasas de hasta 2,488 Gbps en el sentido de bajada y 1,244 Gbps en el sentido de subida. Para poder comprobar que los anchos de banda configurados en el TGMS [4] son los que efectivamente disfruta el cliente en su conexión basta con utilizar cualquier test de velocidad disponible en Internet.

Sin embargo, al superar ciertas tasas de datos, puede que la tarjeta de red del equipo no esté preparada para soportar dicha tasa. Se han realizado pruebas en las que se ha ido aumentando el ancho de banda de forma paulatina para ver hasta qué tasa pueden responder los ordenadores. En los ordenadores fijos del laboratorio 2L007 conectados a 60

las ONUs vía cable Ethernet, modelo *Optiplex740* de la empresa AMD [20], lo máximo que se puede obtener son unos 300 Mbps simétricos. En un ordenador portátil más moderno, modelo *Pavilion dm4* de la empresa *Hewlett-Packard (hp)* [21], conectado a la ONU vía cable Ethernet, se ha conseguido obtener una tasa entre los 500 y 600 Mbps como se observa en la Figura 48.



Figura 48: Test de velocidad para comprobar las tasas de datos

### 3.4 Configuración del servicio de Internet y vídeo

En este apartado, se explicarán los pasos que hay que seguir para la creación de un servicio de vídeo. Como se verá más adelante, tener servicio de vídeo implica tener también configurado un servicio de Internet. En primer lugar, se estudiará la configuración que ha de tener el TGMS y posteriormente, se analizará la del router de las ONUs, como en el apartado anterior. Finalmente, se realizarán unas pruebas para mostrar los resultados obtenidos de la configuración del servicio de vídeo.

#### 3.4.1 Configuración del servicio de Internet y vídeo en la interfaz del TGMS

Para proporcionar servicio de vídeo, es necesario crear dos servicios: uno de tipo multicast, ya que los canales se configurarán como direcciones multicast y otro de tipo Ethernet, para así poder gestionar el tráfico IGMP [28] (*Internet Group Management Protocol*) asociado [4]. Por tanto, el servicio de vídeo es en realidad servicio de Internet y vídeo.

A continuación, se describen los pasos necesarios para crear los dos servicios en el TGMS. El servicio Ethernet se creará de la misma forma que se ha creado en el Apartado 3.3.1, siguiendo exactamente los mismos pasos. En cuanto al servicio multicast, los pasos a seguir son los siguientes [4]:

1. Configuración de los canales multicast: se realiza en el icono **+** dentro del apartado *Multicast Channels* del menú *Profiles*, donde se definen los canales multicast. Estos canales se deben definir como una dirección IP de tipo multicast (rango 224.0.0.2–239.255.255.255) [4]. Cada canal se puede caracterizar por una dirección IP única o por un rango de direcciones. Es muy importante recalcar que solo se podrá recibir tráfico multicast en los canales que hayan sido configurados. En la Figura 49, se muestra un ejemplo de configuración de canales multicast.

The screenshot shows the 'GPON Management System' interface with the 'Multicast Channels' tab selected. The table below lists the configured channels with their start and end IP addresses and a status indicator.

+	Channel	Start	End	
+	Canal 7	224.224.224.224	224.224.224.224	✘
+	Canal 4	225.0.0.2	225.0.0.2	✘
+	Canal 6	227.2.2.2	227.2.2.2	✘
+	Canal 5	233.33.33.33	233.33.33.33	✘
+	Canal 1	239.0.0.2	239.0.0.2	✘
+	Canal 2	239.0.0.3	239.0.0.3	✘
+	Canal 3	239.0.0.4	239.0.0.4	✘

Figura 49: Configuración de los canales multicast

2. Creación de un pack de canales multicast: se realiza en el icono **+** dentro del apartado *Multicast Packs* del menú *Profiles*. Los canales multicast creados se deben asociar en paquetes que serán posteriormente asignados a los clientes. Se puede definir paquetes diferentes, con los canales multicast que cada cliente quiera. En la Figura 50, se muestra un ejemplo de un paquete de canales, que agrupa todos los canales multicast creados en el paso anterior.

+	Pack	Channel	IP Start	IP End	1
⊙	Video	Canal 1	239.0.0.2	239.0.0.2	✘
		Canal 2	239.0.0.3	239.0.0.3	
		Canal 3	239.0.0.4	239.0.0.4	
		Canal 4	225.0.0.2	225.0.0.2	
		Canal 5	233.33.33.33	233.33.33.33	
		Canal 6	227.2.2.2	227.2.2.2	
		Canal 7	224.224.224.224	224.224.224.224	

Figura 50: Creación de un paquete de canales multicast

3. Creación de un mapa de ancho de banda para el servicio multicast: es necesario crear un mapa de ancho de banda para el servicio multicast diferente al del servicio Ethernet. Sin embargo, el ancho de banda de la conexión a Internet será el definido para el servicio Ethernet. La forma de crear el ancho de banda se muestra en la Figura 37.
4. Creación del servicio multicast: se realiza de la misma forma que en el Apartado 3.3.1 (ver Figura 38). El tipo de servicio será ahora multicast. Además, el mapa de ancho de banda seleccionado será el creado en el paso anterior y el mapeado VLAN será el mismo que el utilizado para el servicio Ethernet asociado, como se observa en la Figura 51.

⊙	IPTV_IGMP_806	Ethernet	BW_IPTV_IGMP	VLAN_IP_Estatica	✘
⊙	IPTV_IGMP_833	Ethernet	BW_IPTV_IGMP	VLAN_Servidor_DHCP	✘
⊙	IPTV_Multicast_806	Multicast	BW_IPTV_Multicast	VLAN_IP_Estatica	✘
⊙	IPTV_Multicast_833	Multicast	BW_IPTV_Multicast	VLAN_Servidor_DHCP	✘

Figura 51: Creación de los servicios Ethernet y multicast

5. Creación y agrupación de los servicios Ethernet y multicast en un perfil de usuario: se lleva a cabo de la misma manera que en el Apartado 3.3.1 (ver Figura 40). En este caso, hay que agrupar en el mismo perfil de usuario el servicio Ethernet y el servicio multicast, que forman entre ambos el servicio de Internet y Vídeo. Se puede añadir más de un servicio de Internet y Vídeo a un perfil (Figura 52), aunque en esta red, solamente

puede haber dos servicios de Internet y Video (asociados a los dos identificadores VLAN operativos para la red, el 806 y el 833).

●	Video_806	No	No	IPTV_Multicast_806	MC	UNI-0	BW_IPTV_Multicast	VLAN_IP_Estatica	✘
				IPTV_IGMP_806	Eth	UNI-0	BW_IPTV_IGMP	VLAN_IP_Estatica	
●	Video_833	No	No	IPTV_IGMP_833	Eth	UNI-0	BW_IPTV_IGMP	VLAN_Servidor_DHCP < >	✘
				IPTV_Multicast_833	MC	UNI-0	BW_IPTV_Multicast	VLAN_Servidor_DHCP < >	

Figura 52: Creación de perfiles de usuario con servicio de Internet y vídeo

6. Asignación del perfil de usuario a las ONUs: finalmente, se asignará el perfil correspondiente a la ONU (u ONUs) desde el menú de configuración de ONU (Figura 27). Al desplegar el campo *Profile*, se mostrarán todos los perfiles disponibles. Tras seleccionar el perfil deseado, se hará clic en el botón . Posteriormente, en la pantalla principal del TGMS, será necesario sincronizar la información entre TGMS y OLT. Pulsando sobre el icono  se produce la sincronización (el proceso tarda unos segundos) como se observa en la Figura 40. Este proceso es muy importante pues de no producirse, los cambios del TGMS no se volcarán a la red.
7. Asignación de los packs de canales: finalmente, para completar la configuración, es necesario ir al menú de configuración por puertos de la ONU (Figura 26). En este menú, aparecen todos los servicios asociados a la ONU. En el servicio multicast, el campo *Service Status* no estará configurado (*NOT Configured*) como se ve en la Figura 53. Para configurarlo, será necesario pulsar el icono  situado justo al lado. Una vez pulsado, se podrán añadir con el botón  los packs de canales que se deseen (ver Figura 54). Finalmente, habrá que pulsar  para guardar la configuración y desde la pantalla principal del TGMS, sincronizar la información entre el TGMS y el OLT en el icono .

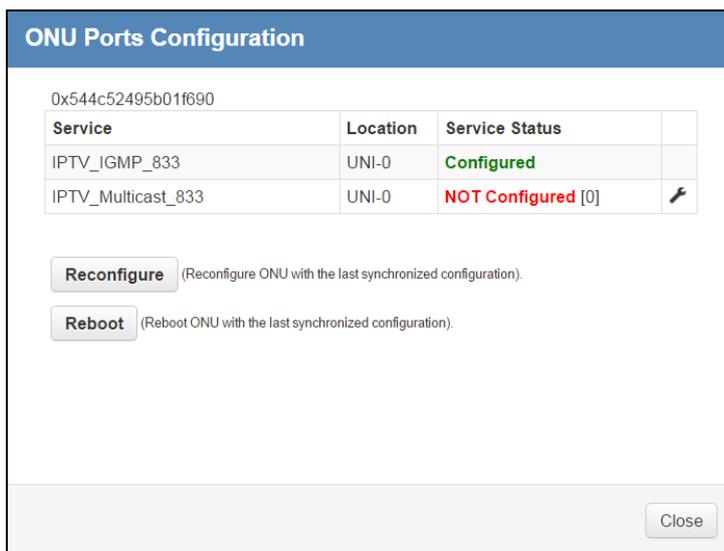


Figura 53: Configuración del servicio multicast en el menú de configuración por puertos de la ONU

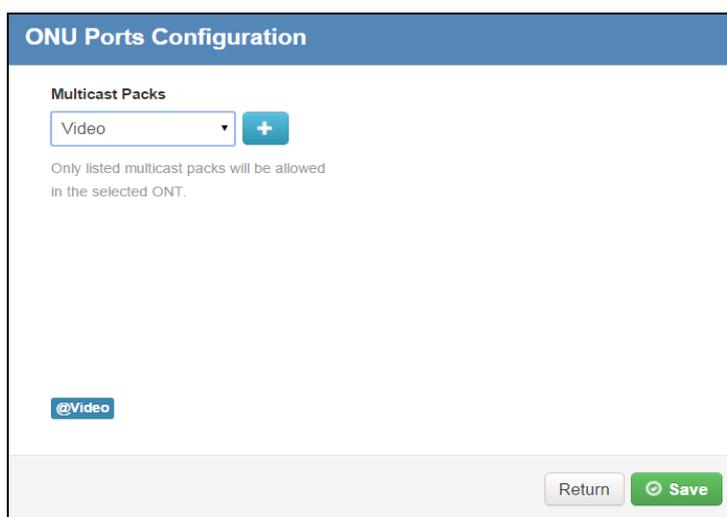


Figura 54: Asignación de packs de canales a la ONU

### 3.4.2 Configuración del servicio de Internet y vídeo en los routers de las ONUs

Al igual que para el servicio de Internet, es necesario configurar el router de la ONU [12] para poner en marcha el servicio de Internet y Vídeo.

Los pasos a seguir son exactamente los mismos que en el Apartado 3.3.2 con una salvedad. En el paso 5, además de activar la opción *Enable NAT* [25], hay que habilitar la opción *Enable IGMP Multicast Proxy* para que el tráfico de control IGMP [28] se curse sin problemas. En el caso del servicio de Internet esto es opcional.

### 3.4.3 Pruebas y resultados de la configuración de un servicio de Internet y vídeo

Con el servicio de Internet ya se realizaron pruebas en el Apartado 3.3.3. Para realizar pruebas con el servicio de vídeo, se utilizará el programa *VLC Media Player* [29].

Como servidor de vídeo, se utilizará uno de los ordenadores del laboratorio 2L007 (exceptuando los ordenadores 7, 10, 14 y 18, que están conectados a las ONUs). Desde este ordenador, se lanzará la emisión multicast siguiendo estos pasos [30]:

1. Abrir una instancia de *VLC Media Player*. Desde la pestaña Medios, seleccionar *Emitir*.
2. Desde la pantalla *Abrir Medio*, en la pestaña *Archivo*, seleccionar el vídeo a emitir pulsando el botón *Añadir*. Una vez seleccionado el vídeo, pulsar *Emitir*.
3. En la pantalla *Fuente*, simplemente pulsar *Siguiente*.
4. En la pantalla *Configuración de destino*, en el campo *Nuevo destino*, se escoge la configuración del método de emisión. Se debe seleccionar *UDP (legacy)* y pulsar *Añadir* (ver Figura 55).

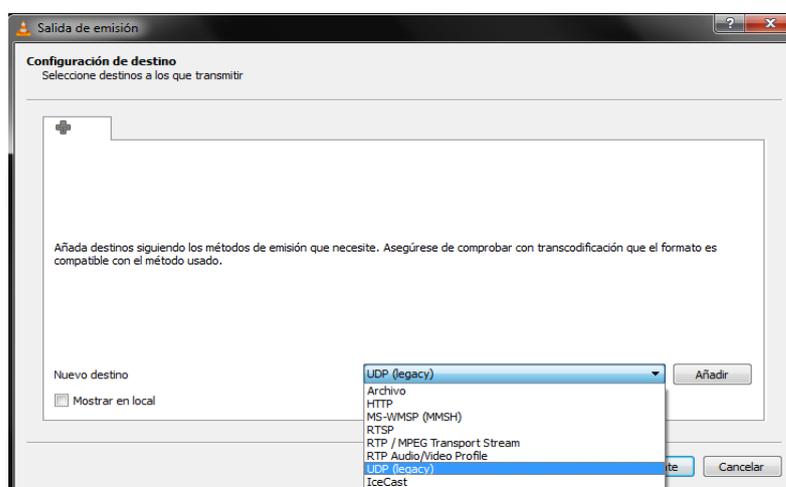


Figura 55: Configuración del método de emisión multicast a través del *VLC Media Player*

5. Configurar la dirección IP y el puerto de emisión. La dirección IP de emisión tiene que estar en el rango multicast definido en el TGMS [4] (224.0.0.2 – 239.255.255.255) y ser una de las que se han configurado en el TGMS. En este caso, se escogerá la IP 239.0.0.2 y el puerto 1234 (ver Figura 56).

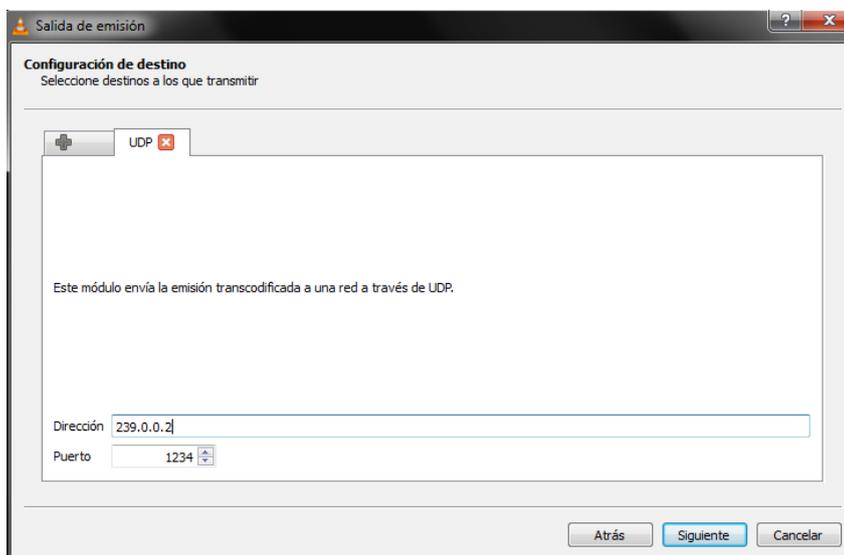


Figura 56: Configuración de la dirección IP y el puerto de emisión en el *VLC Media Player*

6. En la pantalla *Opciones de Transcodificación*, es recomendable habilitar la opción *Habilitar transcodificar*.
7. Finalmente, en la pantalla de *Configuración de preferencias*, hay que seleccionar la opción *Emitir todas las emisiones elementales* y pulsar *Emitir*. De esta forma, ya estaría configurada la emisión.

A continuación se deberá configurar el cliente. Desde uno de los ordenadores conectados a las ONUs (bien sea por cable Ethernet o por Wifi), se lanzará otra instancia de *VLC Media Player* [29] para conectarse a la emisión. Para ello, desde la pestaña Medios, se debe seleccionar *Abrir ubicación de red* e introducir la URL de la siguiente forma: *udp://@direcciónIP:puerto*. A la vista de la Figura 56, la URL será *udp://@239.0.0.2:1234*. En cuestión de segundos, se comenzará a visualizar la emisión, tal y como se observa en la Figura 57.

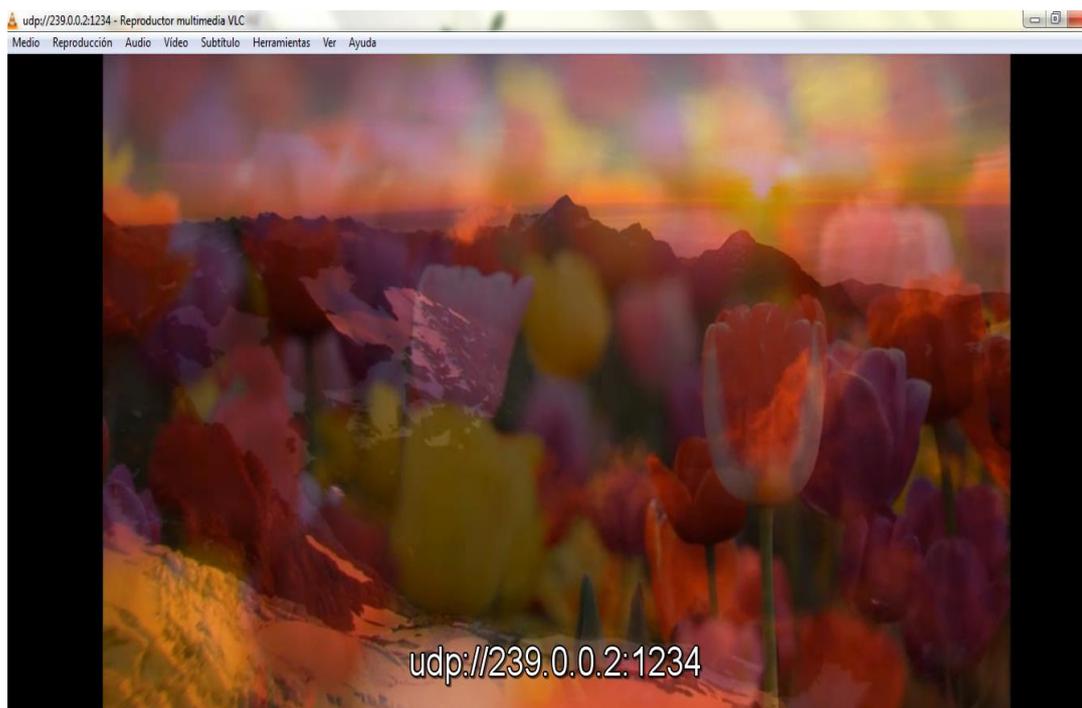


Figura 57: Prueba de emision multicast

Se pueden visualizar diferentes emisiones al mismo tiempo ya que se pueden lanzar emisiones en cada uno de los canales configurados en el pack de canales multicast del TGMS.

### 3.5 Conclusiones

El TGMS (*TELNET GPON Management System*) es un sistema web de gestión para la red GPON muy intuitivo y visual. Permite tener un control de los OLTs y las ONUs conectadas a la red. La configuración de servicios se hace de forma muy esquemática, sin necesidad de conocer el protocolo GPON, lo que facilita mucho el trabajo al usuario del programa. A grandes rasgos, la forma de proporcionar servicios a las ONUs es la siguiente: creación de los mapeados VLAN, creación de los mapas de ancho de banda, creación de los servicios y finalmente agrupación en perfiles de usuario. Estos perfiles pueden ser asignados a varios clientes, es decir, a varias ONUs.

Al igual que es necesario configurar el TGMS, también hay que configurar el router integrado en las ONUs, que son de nivel 3. Se deben configurar las interfaces WAN para que los equipos reciban correctamente direcciones IP. El etiquetado VLAN

determina la forma de asignación de esta IP. En el caso de disponer de un servidor DHCP, la asignación de la dirección IP será automática. De no ser así, habrá que introducirla manualmente. Los dispositivos se pueden conectar a las ONUs vía cable Ethernet o vía Wifi.

En el Capítulo 4, se verá la otra forma de gestión, el CLI (*Command Line Interface*), mucho más centrada en el protocolo GPON y con diferencias significativas respecto al TGMS.



# 4

## Configuración de servicios en la red GPON utilizando el modo de gestión CLI

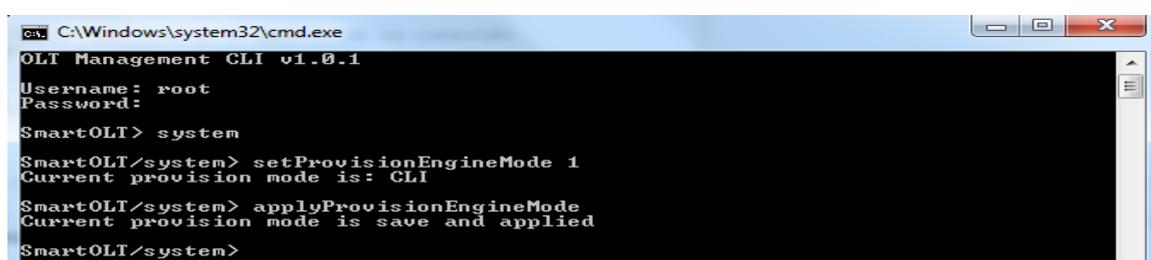
### 4.1 Introducción

En este capítulo, se describe el otro modo de gestión de la red de acceso GPON, denominado CLI (*Command Line Interface*) [5]. Se trata de una interfaz basada en línea de comandos cuyo principal objetivo es conocer en mayor profundidad el protocolo GPON y entender su funcionamiento. Los comandos están relacionados con las especificaciones del protocolo GPON [3] y se configuran a través de la interfaz de gestión y control de ONUs (OMCI - *ONU management and control interface*) definida en el estándar ITU-T G988 [31]. De esta forma, configurar servicios a través del CLI es más complejo que a través del TGMS [4].

La principal diferencia con el TGMS es que el CLI no tiene persistencia al tratarse de un CLI académico. Cuando el modo de gestión TGMS está activado y se apaga la red, al volver a arrancarla, se carga de forma automática la última configuración existente. Esto en el CLI no sucede así y constituye la principal desventaja de este modo de gestión. Para intentar paliar esta falta de persistencia y agilizar la configuración de la red, se utiliza un programa con los lenguajes de programación Python [6] y XML [7]. Esto se explicará en el Capítulo 5.

Por defecto, la red GPON tiene como modo de gestión el TGMS [10]. El cambio de modo de gestión, al igual que el acceso al CLI, se realiza en el mismo ordenador desde que el que se accede al TGMS. Para cambiar de modo de gestión, es necesario realizar una conexión Telnet desde el símbolo del sistema de Windows (cmd) a la

dirección IP del OLT 172.26.128.38 y al puerto 4555. El comando a utilizar es `C:\>telnet 172.26.128.38 4555`. Una vez introducido el comando, aparecerá la pantalla de la Figura 58, en la que se requerirá un *username* (*root*) y un *password* (*root*). De esta forma se accede al menú *System*. La forma de seleccionar un modo de gestión y otro es con el comando `setProvisionEngineMode`. Este comando seguido de un 0 activa el TGMS y seguido de un 1 activa el CLI. Finalmente se debe ejecutar el comando `applyProvisionEngineMode` para aplicar el cambio. En la Figura 58 vienen recogidos estos pasos.



```
C:\Windows\system32\cmd.exe
OLT Management CLI v1.0.1
Username: root
Password:
SmartOLT> system
SmartOLT/system> setProvisionEngineMode 1
Current provision mode is: CLI
SmartOLT/system> applyProvisionEngineMode
Current provision mode is save and applied
SmartOLT/system>
```

Figura 58: Activación del modo de gestión CLI para la configuración de la red GPON

A lo largo del capítulo, se hará, en primer lugar, una breve descripción de la interfaz del CLI con su estructura de menús [5] para posteriormente pasar a la configuración de servicios y perfiles de abonaado. A la hora de configurar los servicios, ya sean de Internet o de Internet y vídeo, es necesario configurar tanto la interfaz del CLI como los routers integrados en las ONUs.

## 4.2 Descripción del interfaz CLI

El CLI (*Command Line Interface*), o interfaz de línea de comandos [5], es un sistema de gestión de la red de acceso GPON que permite el control de la red de acceso GPON a través de una serie de comandos que están directamente relacionados con el protocolo GPON [3] y la interfaz de gestión y control de ONUs (*OMCI - ONU management and control interface*) [31]. A diferencia con el TGMS, la interfaz del CLI es mucho menos visual e intuitiva pero permite entender el funcionamiento de GPON a un nivel más bajo.

La forma de acceder al CLI es a través de una conexión Telnet desde el símbolo del sistema de Windows (`cmd`), comúnmente conocido como terminal de Windows, a la

dirección IP del OLT 172.26.128.38 y al puerto 4551. El comando a utilizar es `C:\>telnet 172.26.128.38 4551`. Una vez introducido el comando, aparecerá la pantalla de la Figura 59. En primer lugar, se pedirá un primer *password*, que es *TLNT25*, que da acceso al menú inicial (*Initial Mode*). Posteriormente, se debe acceder al menú con privilegios (*Privileged Mode*) para lo cual se debe introducir el comando *enable*. Tras esto, el CLI volverá a pedir un nuevo *password*, que en esta ocasión es *TLNT145*. El proceso de acceso al CLI viene recogido en la Figura 59.

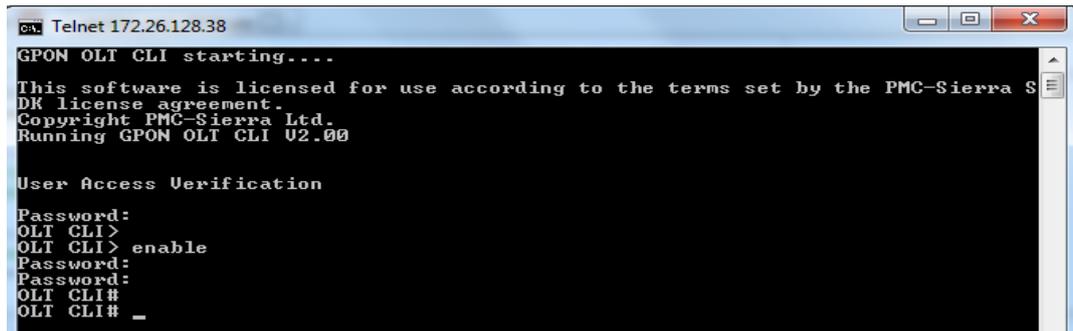


Figura 59: Acceso al CLI desde el terminal de Windows

Una vez se ha accedido al CLI, es necesario conocer los diferentes menús de los que consta el sistema. En la Figura 60, viene recogida la estructura y jerarquía existente entre todos los menús [5]. La forma de pasar de un menú a otro es a través de los comandos que unen los menús mediante flechas, como se observa en la Figura 60.

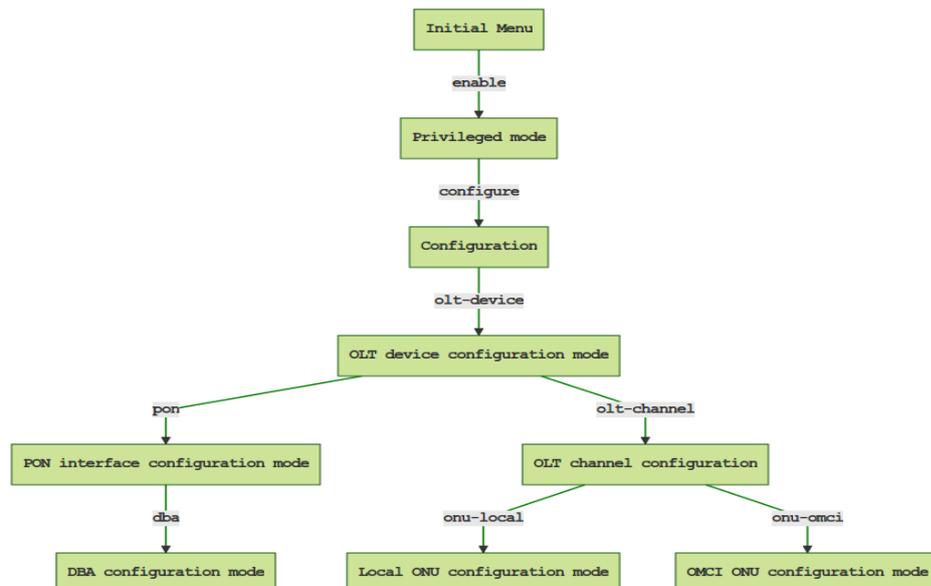


Figura 60: Estructura y jerarquía de menús del CLI

A continuación, se explicará de forma breve la estructura del CLI. La lista de todos los comandos que se pueden ejecutar desde cada menú viene recogida en el manual del CLI académico [5]. Muchos de los comandos son los mismos aunque aparecen en diferentes menús. En la configuración de los servicios, se hará especial énfasis en los comandos más relevantes.

- Menú *Initial Mode / Privileged Mode* (Menú Inicial/Menú con Privilegios). Son los menús a los que se accede tras introducir el primer y segundo *password*, respectivamente.
- Menú *Configuration* (Menú de Configuración). Se accede ejecutando el comando *configure* en el menú con privilegios.
- Menú *OLT device configuration mode* (Menú de Tarjeta de OLT). Se accede ejecutando el comando *olt-device OLT\_DEVICE\_INDEX* en el menú de configuración. En el caso de esta red, solo hay un OLT con lo que el índice de dispositivo de OLT será el 0.

Llegados a este punto, hay una bifurcación en la estructura de menús como se ve en la Figura 60. Se empezará analizando los menús de la parte izquierda.

- Menú *PON interface configuration mode* (Menú de red PON). Se accede ejecutando el comando 'pon' en el menú de tarjeta de OLT (*OLT device configuration mode*).
- Menú *DBA configuration mode* (Menú de configuración del algoritmo DBA (*Dynamic Bandwidth Allocation*)) [24]. Se accede ejecutando el comando *dba pythagoras OLT\_CHANNEL\_INDEX* en el menú de red PON. El índice de canal del OLT es el 0 ya que, como se explicó en los capítulos anteriores, las ONUs están conectadas al puerto PON 0.

Para volver al menú de tarjeta OLT (*OLT device configuration mode*) donde se produce la división en la estructura de menús, hay que ejecutar en comando *exit* en dos ocasiones. El comando *exit* permite subir un nivel en la estructura de menús. El comando

*end* hace volver directamente al menú con privilegios. A continuación se explicarán los menús de la parte de la derecha.

- Menú *OLT channel configuration* (Menú de canal de OLT). Se accede ejecutando el comando *olt-channel OLT\_CHANNEL\_INDEX* en el menú de tarjeta de OLT (*OLT device configuration mode*). El índice de canal del OLT es el 0 ya que, como se mencionó anteriormente, las ONUs están conectadas al puerto PON 0.

Desde este menú, se accede a dos menús diferentes, según la bifurcación tomada:

- Menú *Local ONU configuration mode* (Menú local de la ONU). Se accede ejecutando el comando *onu-local ONU\_INDEX* en el menú de canal de OLT (*OLT channel configuration*). El índice de la ONU no es fijo y cambia cada vez que se apaga o reinicia la red. En este caso, en el que hay 4 ONUs, los identificadores irán de 0 a 3. El CLI irá asignando el identificador de ONU más bajo disponible a medida que detecte una ONU. Este hecho constituye otra diferencia fundamental con respecto al TGMS.
- Menú *OMCI ONU configuration mode* (Menú OMCI de la ONU). Se accede ejecutando el comando *onu-omci ONU\_INDEX* en el menú de canal de OLT (*OLT channel configuration*).

En la Figura 61, se muestra todo este proceso de movimiento por los diferentes menús del CLI. Al ejecutar un comando, siempre se ve en el *prompt* (el conjunto de caracteres que se muestran en la línea de comandos para indicar que está a la espera de órdenes) el menú desde el que se está ejecutando.

Asimismo, también se observa la utilidad de los comandos *exit* y *end*. Si se ejecuta un comando que no existe, aparecerá un mensaje de error *Error, Unknown command* (como se ve en la parte inferior de la Figura 61). El CLI es sensible a mayúsculas y minúsculas. Los comandos deben ser precisos, ya que ante cualquier mínimo error, saltará un mensaje de error [5].

```
Running GPON OLT CLI U2.00

User Access Verification
Password:
OLT CLI>
OLT CLI> enable
Password:
Password:
OLT CLI#
OLT CLI# configure
OLT CLI<config>#
OLT CLI<config># olt-device 0
OLT CLI<DEV0>#
OLT CLI<DEV0># pon
OLT CLI<OLT0 PON>#
OLT CLI<OLT0 PON># dba pythagoras 0
OLT CLI<OLT0 CH0 PON-DBA<Pythagoras>>#
OLT CLI<OLT0 CH0 PON-DBA<Pythagoras>># exit
OLT CLI<OLT0 PON>#
OLT CLI<OLT0 PON># exit
OLT CLI<DEV0>#
OLT CLI<DEV0># olt-channel 0
OLT CLI<DEV0 CH0>#
OLT CLI<DEV0 CH0># onu-local 0
OLT CLI<DEV0 CH0 LOC-ONU0>#
OLT CLI<DEV0 CH0 LOC-ONU0># exit
OLT CLI<DEV0 CH0>#
OLT CLI<DEV0 CH0># onu-omci 0
OLT CLI<DEV0 CH0 ONU<OMCI>0>#
OLT CLI<DEV0 CH0 ONU<OMCI>0># end
OLT CLI#
OLT CLI# Configure
Error, Unknown command.
OLT CLI#
OLT CLI#
```

Figura 61: Movimiento por la estructura de menús del CLI

Como se ha mencionado anteriormente, un aspecto fundamental que diferencia al TGMS y al CLI es que el CLI no tiene índices fijos de ONU. Por tanto, se hace necesario trabajar con el número de serie (*serial number*) de las ONUs, formado por los campos *Vendor ID* y *Vendor Specific* (este último formado por los últimos 8 valores hexadecimales de la dirección MAC) [4][10]. Este campo aparece en la Figura 62.

Para conocer las ONUs que están conectadas a la red, su estado y los identificadores asignados, hay que introducir desde el menú de canal de OLT (*OLT channel configuration*) el comando *show serial-number allocated*. En la Figura 62, se muestra esta información.

```
OLT CLI<DEV0 CH0># show serial-number allocated
Allocated Serial number of OLT device/channel 0/0
# | serial number | onu-id | sn type | onu state
-----
1 | 54-4c-52-49-5b-01-f6-90 | 0 | Dynamic | Activate
2 | 54-4c-52-49-5b-01-f6-d8 | 1 | Dynamic | Activate
3 | 54-4c-52-49-5b-01-f7-28 | 2 | Dynamic | Activate
4 | 54-4c-52-49-5b-01-f7-30 | 3 | Dynamic | Activate
Activated ONU num: 4
```

Figura 62: Información general sobre las ONUs conectadas

El *serial number* se compone de una parte común a todas las ONUs, que es el *Vendor ID* (54-4c-52-49) y que identifica a las ONUs de un mismo fabricante, en este caso TELNET Redes Inteligentes [2]. La parte que diferencia a cada ONU es el *Vendor Specific*, formado por los últimos 8 valores hexadecimales de la dirección MAC [4][10]. Por tanto, lo que sirve para identificar de forma unívoca a una ONU en el CLI no es el identificador de ONU, sino la dirección MAC. Las direcciones MAC de las 4 ONUs aparecían recogidas en el Apartado 3.3.2.

## **4.3 Configuración del servicio de Internet**

En este apartado, se detallarán los pasos que hay que seguir para configurar el servicio de Internet. Además de configurar el CLI, hay que configurar los routers de las ONUs (al igual que se hizo con el TGMS en el Capítulo 3). Sin embargo, la forma de proceder presenta varias diferencias que se expondrán en los diferentes subapartados.

### **4.3.1 Configuración del servicio de Internet en los routers de las ONUs**

Para disponer de servicio de Internet en las ONUs, es necesario configurar el modo de gestión de la red GPON activo (TFMS, CLI) así como el router que llevan integrado las ONUs [12].

En el modo de gestión TGMS, el orden en el que se lleve a cabo esta configuración no tiene relevancia. Así pues, en el Capítulo 3, primero se describe la configuración del TGMS (Apartado 3.3.1), posteriormente la del router de las ONUs y finalmente, se volcaba la configuración del TGMS a las ONUs para que las interfaces WAN pasarán al estado *Connected* (Apartado 3.3.2). Sin embargo, si se hubieran configurado las interfaces WAN primero y posteriormente, se hubiera configurado el TGMS y volcado la configuración a las ONUs, el resultado habría sido el mismo.

Sin embargo, en el CLI, el orden en el que se produce la configuración debe ser de forma obligatoria el que se describe a continuación. En primer lugar, se ha de configurar el router de las ONUs, es decir, se han de configurar las interfaces WAN para que las ONUs reciban direcciones IP correctamente. Tras esto, hay que configurar el interfaz del CLI utilizando los comandos que se detallarán en el Apartado 4.3.2. El CLI no tiene una opción para volcar la configuración a las ONUs, es decir, la configuración

se transfiere a medida que se introducen los comandos. Por ello, es necesario que las interfaces WAN ya estén configuradas mientras se va configurando el servicio. Si primero se configura el servicio y posteriormente se configuran las interfaces WAN, la configuración del servicio de Internet no se volcará a las ONUs y no se actualizará el estado de las interfaces WAN, por lo que nunca llegarán al estado *Connected* y no recibirán direcciones IP.

La manera de configurar el router de las ONUs es exactamente la misma que en el Apartado 3.3.2. La forma de conexión al router de la ONU y los pasos a seguir para configurar las interfaces WAN están descritos en dicho apartado.

#### **4.3.2 Configuración del servicio de Internet en la interfaz del CLI**

La configuración de servicios en el TGMS abstraía de la complejidad del protocolo GPON [3]. El CLI, por el contrario, es un modo de gestión totalmente relacionado con el protocolo GPON, que permite entender su funcionamiento.

La configuración del servicio de datos (Internet) en el protocolo GPON se lleva a cabo a través de la interfaz de gestión y control de ONUs denominada OMCI (*ONU Management and Control Interface*), definida en el estándar ITU-T G988 [31]. OMCI es empleado por el OLT para controlar las ONUs, en concreto, el establecimiento y liberación de conexiones con las ONUs, la configuración y administración de servicios y/o la gestión de alarmas, seguridad, averías, rendimiento. OMCI se ha desarrollado para facilitar la interoperabilidad entre fabricantes [2].

OMCI funciona con el protocolo MIB (*Management Information Base*) [31]. Este protocolo describe el intercambio de información a través de la OMCI y está estructurado en torno a entidades. Estas entidades son representaciones abstractas de recursos y servicios en una ONU. El protocolo MIB define en torno a 300 entidades. Muchas de ellas son opcionales y no aplicables en la mayoría de ONUs. Por tanto, algunas de las entidades no se podrán definir en el CLI. También define las relaciones que deben existir entre las entidades para implementar servicios. Los mensajes OMCI viajan encapsulados en tramas GEM (*GPON Encapsulation Method*).

A continuación, se describen las entidades necesarias para la configuración del servicio de Internet así como la relación que han de tener entre ellas según lo establecido en el estándar OMCI [31]:

- *T-Cont*: contenedor de transmisión (*Transmission Container*) de datos. Esta entidad es la unidad de control básica de servicio en el canal de subida GPON. Un *T-Cont* almacena paquetes en colas de prioridad. Va asociado a un identificador denominado *Alloc-ID*, que se asocia a su vez a un identificador de ONU. Este *Alloc-ID* debe ser único para cada *T-Cont*.
- *MAC Bridge Service Profile*: esta entidad gestionada representa un puente MAC y define el tipo de perfil de servicio. Se define una instancia de esta entidad para todos los servicios de una misma ONU y se modela a través de las entidades *MAC Bridge Port Configuration Data*.
- *MAC Bridge Port Configuration Data*: estas entidades organizan los datos de estado asociados con un puente MAC. Se pueden vincular diferentes entidades de este tipo a un mismo *MAC Bridge Service Profile*.
- *GEM Port Network CTP*: estas entidades representan la terminación de un puerto GEM en una ONU. Se relaciona con la entidad *T-Cont* a través de un puerto (*Port ID*) que a su vez se vincula al *Alloc-ID* asociado al *T-Cont*. También se relaciona con la entidad *GEM Interworking Termination Point*.
- *GEM Interworking Termination Point*: estas entidades se encargan de la transformación de flujo de bytes a tramas GEM y viceversa. Se asocian a las entidades *GEM Port Network CTP* y *MAC Bridge Port Configuration Data*.
- *VLAN Tagging Filter Data* / *Extended VLAN Tagging Operation Configuration Data*: estas dos entidades organizan los datos asociados al etiquetado VLAN. Además de vincularse entre sí, se relacionan con las entidades *MAC Bridge Port Configuration Data*.

Una vez explicadas todas las entidades, se detallará la forma en la que se relacionan mediante la ejecución de los comandos del CLI. Estos comandos permitirán crear estas entidades y configurar el servicio de Internet.

La Figura 63, que parte del estándar OMCI (Figura 8.2.2.5 del estándar) [31], está adaptada al caso de esta red ya que la figura del estándar presenta algunas entidades que no tienen ningún comando CLI asociado. La Figura 63 define las relaciones que han de seguir las entidades definidas anteriormente para configurar el servicio de Internet.

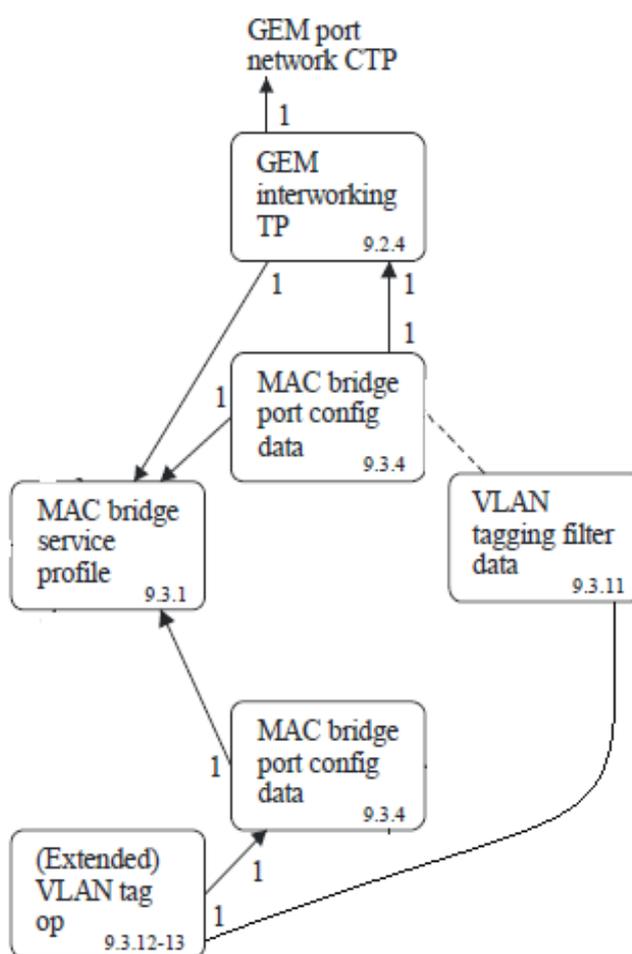


Figura 63: Relación entre las entidades del protocolo MIB para configurar el servicio de Internet

A modo de ejemplo, se llevará a cabo la configuración del servicio de Internet en la interfaz del CLI en la ONU con identificador de ONU 0. Si se quiere configurar una

ONU en concreto, se debe acudir al menú de canal de OLT (*OLT channel configuration*) y ejecutar el comando *show serial-number allocated* para saber cuál es el identificador de la ONU que corresponde con el número de serie de la ONU. Se seguirán los ejemplos proporcionados por la empresa TELNET Redes Inteligentes en su manual de scripts del CLI académico [32], adaptados a la red analizada en este estudio. Los pasos a seguir y el orden de los mismos se presentan a continuación. Este ejemplo de configuración del servicio de Internet está recogido de forma resumida en el Apéndice A.

1. En primer lugar, se ha de crear un canal OMCI en el menú local de la ONU (*Local ONU configuration mode*). El identificador de canal OMCI será el mismo que el de la ONU. Partiendo del menú con privilegios (*Privileged mode*), los comandos a ejecutar son los siguientes:

```
configure
olt-device 0
olt-channel 0
onu-local 0
omci-port 0
exit
```

2. En el menú OMCI (*OMCI ONU configuration mode*), se borrarán todas las entidades presentes para que no haya problemas con configuraciones anteriores y en el menú de canal de OLT (*OLT channel configuration*) se activará la corrección de errores hacia adelante (FEC). Este paso no es obligatorio pero sí es recomendable.

```
onu-omci 0
ont-data mib-reset
exit
fec direction uplink 0
```

3. En el menú local de la ONU (*Local ONU configuration mode*) se definirá un *Alloc-ID*, que quedará vinculado a esa ONU. Este *Alloc-ID* es único en toda la red (no puede ser utilizado por otra ONU) e irá vinculado a la entidad *T-Cont* (punto 4). Posteriormente, en el menú de canal de OLT

(*OLT channel configuration*), se asociará ese *Alloc-ID* a un puerto (*Port-ID*) que estará vinculado a la entidad *GEM Port Network CTP* de la Figura 63, que se explicará con más detalle en el punto 7. Tanto el *Alloc-ID* como el puerto son identificadores únicos en toda la red y no se pueden reutilizar (ni aunque se esté configurando otra ONU).

```
onu-local 0
alloc-id 810
exit
port 800 alloc-id 810
```

4. En el menú OMCI (*OMCI ONU configuration mode*), se procederá a la creación de las entidades que formarán el servicio. Los parámetros relevantes de configuración de estas entidades irán destacados en negrita. El resto de parámetros tomarán los valores por defecto definidos por el fabricante [2] en su manual de scripts del CLI académico [32].

En primer lugar, se asociará el *Alloc-ID* definido con el *T-Cont* con identificador 0 (este identificador sigue un orden ascendente con rango 0-255). El identificador de *T-Cont* es local, de forma que el resto de ONUs pueden usar el *T-Cont* con identificador 0.

```
onu-omci 0
t-cont set slot-id 128 t-cont-id 0 alloc-id 810
```

Al crear una entidad se generará una respuesta. En caso de que haya creado correctamente, la respuesta será *Command Success*. Si hubiera habido algún fallo, aparecería un mensaje de error y en caso de ya estar creada dicha entidad, el CLI avisaría de que esa entidad ya había sido creada.

5. Creación de la entidad *MAC Brigde Service Profile*, que define como perfil de servicio un puente MAC. Se crea una entidad de este tipo para todos los servicios que se quieran configurar en una ONU. Se vinculará

esta entidad con los *MAC Bridge Port Configuration Data* a través del campo *bridge-group-id*, que se configurará en 1 (podría tomar otro valor). El campo *slot-id* debe tener el valor 0 al tratarse de ONUs con router integrado [12]. El resto de parámetros toman los valores por defecto [32].

```
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
```

6. Creación de los *MAC Bridge Port Configuration Data*. Ambos tendrán el campo *bridge-id-ptr* con el mismo índice que campo *bridge-group-id* de la entidad *MAC Bridge Service Profile*. Se deben crear dos instancias de esta entidad ya que cada una irá relacionada con unas entidades diferentes (ver Figura 63).

El primer *MAC Bridge Port Configuration Data* debe ir asociado a la entidad *Extended VLAN Tagging Operation Config Data* (ver Figura 63). Para ello, el campo *tp-type* (tipo del punto de terminación del *MAC Bridge*) será de tipo *lan* y el campo *tp-ptr* (puntero del punto de terminación) será el mismo que el número de instancia de la entidad *Extended VLAN Tagging Operation Config Data* (en este caso, 257 que es el valor utilizado en los ejemplos [32]). El campo *port-num* debe ser diferente para cada instancia de esta entidad y siguiendo el ejemplo del manual [32], tendrá el mismo valor que el número de instancia (en este caso, 1). El resto de parámetros toman los valores por defecto.

```
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 encaps-method llc lanfcs-ind forward
```

El segundo *MAC Bridge Port Configuration Data* irá asociado a la entidad *VLAN Tagging Filter Data* a través del número de instancia (número que aparece al lado de *instance*) que será en las dos entidades el mismo (en este caso, el 2). También irá asociado a la entidad *GEM Interworking Termination Point* (ver Figura 63). Para ello, el campo *tp-type* ha de ser de

tipo *gem* y el *tp-ptr* tiene que coincidir con el número de instancia del *GEM Interworking Termination Point*. Siguiendo el ejemplo de la instancia anterior, el campo *port-num* tendrá el mismo valor que el número de instancia.

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

7. Creación de la entidad *GEM Port Network CTP*, asociada al *Port-ID* vinculado al *Alloc-ID*, ambos definidos en el paso 3. Esta entidad está asociada al *GEM Interworking Termination Point* (ver Figura 63). El número de instancia tiene que coincidir con el campo *gem-port-nwk-ct-conn-ptr* de la entidad *GEM Interworking Termination Point* (en este caso, ese número es 2). El campo *direction* debe tomar el valor *bidirectional* ya que se produce flujo de datos en los dos sentidos de la comunicación. El resto de valores se toman por defecto de los ejemplos del manual [32].

```
gem-port-network-ctp create instance 2 port-id 800 t-cont-ptr 32768 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
```

8. Creación de *GEM Interworking Termination Point*. Esta entidad se vincula al *GEM Port Network CTP* a través del campo *gem-port-nwk-ctp-conn-ptr*, que debe coincidir con el número de instancia del *GEM Port Network CTP* (en este caso, 2). También se asocia con el segundo *MAC Bridge Port Configuration Data*. El campo *interwork-option* debe ser del tipo *mac-bridge-lan* y el número de instancia debe ser el mismo que el campo *tp-ptr* del *MAC Bridge Point Configuration Data* (en este caso, 2). Esta entidad también se relaciona con la entidad *MAC Bridge Service Profile* mediante el campo *service-profile-ptr* que deberá tener valor 1. Si toma otro valor, irá vinculado a otro tipo de perfil de servicio. Las

relaciones de esta entidad se pueden ver de forma resumida en la Figura 63.

```
gem-interworking-termination-point create instance 2 gem-port-nwk-ctp-conn-  
ptr 2 interwork-option mac-bridge-lan service-profile-ptr 1 interwork-tp-ptr 0  
gal-profile-ptr 0
```

9. Creación del *VLAN Tagging Filter Data*. El número de instancia debe coincidir con el del *MAC Bridge Port Configuration Data* asociado (el segundo en este caso, de ahí que el número de instancia sea 2). En esta entidad, se define el etiquetado VLAN. Se pueden definir hasta 12 etiquetas VLAN. En este caso, se utilizará una sola etiqueta, la 833 con prioridad 7 (la otra etiqueta que puede proporcionar servicio es la 806, como se ha explicado en más de una ocasión en esta memoria). Estos dos parámetros deben coincidir con lo que se haya configurado en la interfaz WAN del router de la ONU (ver Figura 43). En el CLI, a diferencia del TGMS, hay que especificar un valor concreto de prioridad. El TGMS se podía configurar para aceptar cualquier prioridad [4]. Este detalle es fundamental ya que puede ser un error muy fácil de cometer.

```
vlan-tagging-filter-data create instance 2 forward-operation h-vid-a vlan-tag1  
833 vlan-priority1 7 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
```

10. Creación de *Extended VLAN Tagging Operation Config Data*. Esta entidad está asociada al primer *MAC Bridge Port Configuration Data* a través del número de instancia, que coincide con el campo *tp-ptr* del *MAC Bridge Port Configuration Data* (en este caso, 257). El resto de valores se toman por defecto siguiendo los ejemplos del manual [32].

```
extended-vlan-tagging-operation-config-data create instance 257 association-  
type ptp-eth-uni associated-me-ptr 257
```

Una vez creada, se debe llevar a cabo la configuración de esta entidad (por ello se utiliza el parámetro *set instance* seguido del número de instancia). Se debe ejecutar este comando para cada identificador VLAN que se haya utilizado en la entidad *VLAN Tagging Filter Data*. De ahí la relación entre las dos entidades que aparece en la Figura 63.

```
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 833 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 833 treatment-inner-tpid tpid-de-  
copy-from-inner  
exit
```

Llegados a este punto, ya se han configurado todas las entidades necesarias para la creación del servicio.

11. Ahora se deben configurar las reglas VLAN en el OLT y asociarlas al *Port-ID* definido en el paso 3 desde el menú de canal de OLT (*OLT channel configuration*). Al igual que para las entidades, se destacarán los parámetros de configuración más importantes de estos comandos en negrita. El resto se tomarán por defecto de los ejemplos del manual [32].

```
vlan uplink configuration port-id 800 min-cos 0 max-cos 7 de-bit disable  
primary-tag-handling false  
vlan uplink handling port-id 800 primary-vlan none destination datapath c-  
vlan-handling no-change s-vlan-handling no-change new-c-vlan 0 new-s-vlan 0
```

12. Creación del perfil de ancho de banda para el sentido de bajada. Se ha de definir un ancho de banda garantizado (campo *committed-max-bw*) en Kbps y un ancho de banda en exceso (campo *excess-max-bw*) también en Kbps (este campo puede ser 0). Es muy importante que ambos valores sean múltiplos de 64 Kbps. De lo contrario, saltará un mensaje de error. El resto de campo mantendrán los valores por defecto.

```
policing downstream profile committed-max-bw 256000 committed-burst-size 1023 excess-max-bw 51200 excess-burst-size 1023
```

Asignación del perfil de ancho de banda creado anteriormente. Al ejecutar el comando anterior, aparecerán las siguientes líneas:

```
OLT_device_id: 0
OLT_channel_id: 0
downstream_profile_index: 0
```

Este índice de perfil *downstream\_profile\_index* (en este caso es 0 pero a medida que se van creando perfiles se va incrementado), se debe utilizar para la asignación del perfil al *Port-ID* correspondiente (*port-configuration entity port-id*), en este caso 800.

```
policing downstream port-configuration entity port-id 800 ds-profile-index 0
exit
```

- Finalmente, en el menú de configuración del algoritmo DBA (*DBA configuration mode*), se debe asignar el perfil de ancho de banda para el canal de subida. Este perfil se debe vincular al *Alloc-ID* definido en el paso 3 (en este caso 810) y se debe especificar en unidades enteras de Mbps (en el canal de bajada era en múltiplos de 64 Kbps) un ancho de banda garantizado (campo *gr-bw*) y un ancho de banda *Best Effort* (campo *be-bw*), que debe ser igual o mayor que el ancho de banda garantizado. Es importante recalcar que si no se cumplen estos requisitos, el CLI mostrará un mensaje de error. También se debe seleccionar en este punto el tipo de algoritmo DBA: NSR (*Non-Status Reporting*) o SR (*Status Reporting*) *Type 0* (DBRu) o *Type 1* (DBRu). En ambos casos, es el OLT el que realiza la asignación dinámica de ancho de banda pero en el NSR, las ONUs no envían reportes del estado de sus colas [24] de forma periódica al OLT. Se utilizará el algoritmo NSR por lo que el campo *status-report* tomará el valor *nsr*. El resto de parámetros toman los valores por defecto.

```
pon
dba pythagoras 0
```

```
sla 810 service data status-report nsr gr-bw 250 gr-fine 0 be-bw 300 be-fine 0
end
```

Tras realizar todos estos pasos y con las interfaces WAN previamente configuradas (Apartado 4.3.1), el servicio de Internet queda configurado.

Si se ejecuta el comando *event-notify*, aparecen notificaciones sobre el estado de la red a medida que ésta se actualiza. Por ejemplo, al configurar el servicio de Internet, aparecería un mensaje como el de la Figura 64 indicando la configuración IP que recibe la ONU que se esté configurando.

```
Got Attribute Value change notification message
OLT device:0 OLT channel:0 ONU id:1
entity_class: ip-host-config-data
entity_instance: 0x1
Current IP host service address: 10. 0.103.115
Current IP host service submask: 255.255.255. 0
Current IP host service gateway address: 10. 0.103.253
Current Primary DNS service IP address: 157. 88.129. 90
Got Attribute Value change notification message
OLT device:0 OLT channel:0 ONU id:1
entity_class: ip-host-config-data
entity_instance: 0x1
Domain name: tel.uva.es
```

Figura 64: Ejemplo de notificación de la actualización del estado de una ONU

Para finalizar la configuración del servicio de Internet, se analizarán tres posibles situaciones ligeramente distintas. La primera será la configuración de dos servicios en un mismo puerto GEM. La segunda será la configuración de dos servicios en dos puertos GEM y finalmente la tercera será el borrado de servicios. A continuación se describirán con más detalle cada una de las situaciones anteriores:

- Configuración de dos servicios en un mismo puerto GEM (los dos servicios irán configurados en la misma ONU): la configuración del servicio se aplicará a dos identificadores VLAN de forma que dos servicios irán sobre el mismo puerto GEM. Esto puede resultar útil si uno de los servidores a los que está conectada la red mediante uno de esos

identificadores VLAN sufre algún tipo de problema. En ese caso, se podría disponer del servicio en el otro identificador VLAN.

Solo hay que realizar dos pequeñas modificaciones con respecto a los pasos descritos anteriormente. La primera modificación está en el paso 9, en el que se completarán los campos correspondientes al segundo identificador VLAN, es decir, *vlan-tag2 806 vlan-priority2 7*. Se utilizó para el primer identificador VLAN la etiqueta 833 así que en el segundo, se utilizará la etiqueta 806 (son las dos etiquetas que hacen que los servicios puedan configurarse de forma correcta). La segunda diferencia está en que en el paso 10, se debe configurar la entidad *Extended VLAN Tagging Operation Config Data* también para el identificador VLAN 806.

```
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 806 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 806 treatment-inner-tpid tpid-de-
copy-from-inner.
```

- Configuración de dos servicios en dos puertos GEM (los dos servicios irán configurados en la misma ONU): la configuración anterior tiene el inconveniente de que los anchos de banda son exactamente iguales para las dos VLAN. Puede resultar interesante configurar servicios con diferente ancho de banda para poder asignar uno u otro en función de la necesidad de ancho de banda.

Para poder configurar anchos de banda diferentes, la solución es crear servicios independientes vinculados cada uno a un puerto GEM diferente. Por tanto, se trata de replicar la configuración explicada en este apartado utilizando un *Alloc-ID*, un *Port-ID* e identificadores para las instancias de las entidades diferentes. La única entidad que no es necesario replicar es el *MAC Bridge Service Profile*, ya que esta entidad define el tipo de perfil de servicio, que en ambos casos es un puente MAC (*MAC Bridge*). El resto de entidades estarán duplicadas (una para cada servicio) y también será

necesario configurar las reglas VLAN y asignar perfiles de ancho de banda a cada uno de los servicios.

- **Borrado de servicios:** el proceso de borrado de servicios en el CLI se lleva a cabo en dos pasos. En primer lugar, se debe borrar el perfil (o los perfiles) de ancho de banda del canal de bajada asociado al puerto (o puertos) correspondiente. Para ello, desde el menú de canal de OLT (*OLT channel configuration*), se ejecutará el comando *no policing downstream port-configuration entity port-id PORT\_ID* con todos los puertos en los que haya configurados perfiles de ancho de banda. Una vez hecho esto, desde el menú OMCI de la ONU (*OMCI ONU configuration mode*), se ejecutará el comando *ont-data mib-reset* de forma que todas las entidades creadas se borrarán. De esta forma, los servicios creados han sido borrados.

### **4.3.3 Pruebas y resultados de la configuración de un servicio de Internet**

Este apartado no presenta ninguna diferencia con respecto al apartado de resultados del servicio de Internet configurado con el TGMS. Las pruebas realizadas están recogidas en el Apartado 3.3.3.

## **4.4 Configuración del servicio de Internet y vídeo**

En este apartado, se analiza la configuración del servicio de vídeo. Configurar un servicio de vídeo implica configurar también un servicio de Internet ya que el servicio de vídeo, que es de tipo multicast, debe ir asociado a un servicio Ethernet (de Internet) para así poder gestionar el tráfico IGMP [28] (*Internet Group Management Protocol*) asociado [4]. Se describirán los pasos que hay que seguir tanto a la hora de configurar la interfaz del CLI como para configurar los routers integrados en las ONUs.

#### **4.4.1 Configuración del servicio de Internet y Video en los routers de las ONUs**

Como se explicó anteriormente en el Apartado 4.3.1, cuando se configuran servicios con el modo de gestión CLI, es necesario configurar primero los routers de las ONUs y posteriormente la interfaz del CLI. Esto es debido a que no hay una opción para volcar la configuración del CLI a las ONUs sino que el estado de las ONUs se va actualizando a medida que se ejecutan los comandos. Por ello, las interfaces WAN deben haber sido creadas antes de ejecutar los comandos del CLI. De lo contrario, su estado no pasará a *Connected* y no recibirán las direcciones IP.

La manera de configurar el router de cada ONU y por tanto, de configurar las interfaces WAN que permitan la asignación de direcciones IP viene recogida en el Apartado 3.4.2.

#### **4.4.2 Configuración del servicio de Internet y Video en la interfaz del CLI**

Al igual que se mencionó en el Apartado 3.4.1, para disponer de servicio de vídeo, es necesario crear dos servicios, uno de tipo Multicast, ya que los canales se configurarán como direcciones Multicast y otro de tipo Ethernet, para así poder gestionar el tráfico IGMP [28] (*Internet Group Management Protocol*) asociado [4]. Por tanto, el servicio de vídeo es en realidad servicio de Internet y vídeo.

La mayoría de las entidades involucradas en la configuración del servicio de Internet y vídeo fueron explicadas en el Apartado 4.3.2 pero en este caso, hay que utilizar una nueva entidad denominada *Multicast GEM Interworking Termination Point*. Esta instancia representa el punto de unión entre el servicio multicast y la capa GEM. Por tanto, en este punto, los flujos de datos multicast se convierten en paquetes GEM y viceversa. Esta entidad se vincula con las entidades *MAC Bridge Port Configuration Data* y *GEM Port Network CTP*.

La Figura 65, que parte del estándar OMCI (Figura 8.2.2.10 en dicho estándar) [31], está adaptada al caso de nuestra red ya que la figura del estándar presenta algunas entidades que no tienen ningún comando CLI asociado. La Figura 65 define las relaciones que han de seguir las entidades creadas mediante los comandos del CLI para

proporcionar servicio de Internet y vídeo. La parte rodeada con un círculo rojo corresponde al servicio multicast. El resto de entidades son las que forman el servicio de Ethernet (servicio de Internet) y son las mismas que las definidas en la Figura 63.

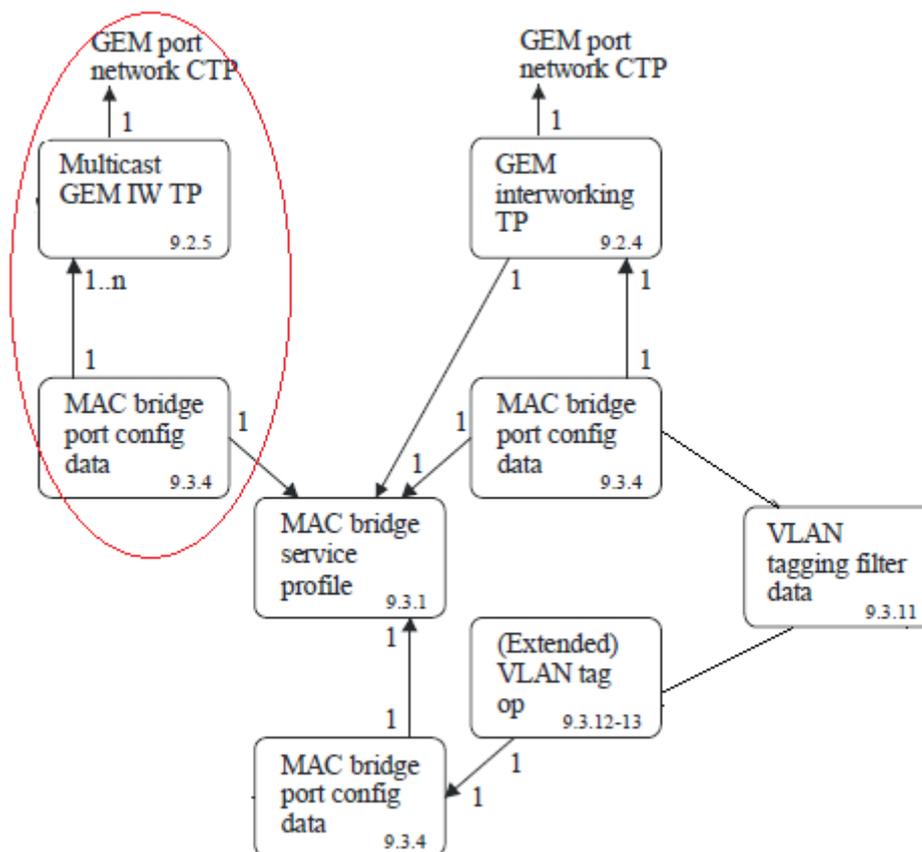


Figura 65: Relación entre las entidades del protocolo MIB para configurar el servicio de Internet y vídeo

A continuación, se mostrarán los comandos necesarios para crear un servicio de Internet y vídeo. Se seguirán de nuevo los ejemplos del manual de scripts del CLI [32], adaptados al caso de estudio de esta red. La mayoría de los comandos son los ya utilizados anteriormente. Se explicarán las diferencias presentes con respecto a la configuración únicamente del servicio de Internet. Estas diferencias corresponden a las entidades rodeadas con un círculo rojo en la Figura 65. Al igual que en el Apartado 4.3.2, los parámetros de configuración más relevantes de las comandos irán en negrita. Este ejemplo de configuración de Internet y vídeo está recogido de forma resumida en el Apéndice A.

1. En primer lugar, se debe crear el canal OMCI y borrar las entidades para evitar conflictos con configuraciones anteriores. Para ello se escribirá el siguiente código:

```
enable  
configure  
olt-device 0  
olt-channel 0  
onu-local 0  
omci-port 0  
exit  
onu-omci 0  
ont-data mib-reset  
exit  
fec direction uplink 0
```

2. Posteriormente, se debe definir un *Alloc-ID* y vincularlo a un puerto:

```
onu-local 0  
alloc-id 810  
exit  
port 800 alloc-id 810
```

3. Desde el menú de canal OMCI, se empiezan a crear las entidades que forman los servicios Ethernet y multicast. Se comienza creando la entidad *T-Cont*. Solamente es necesario crear un *T-Cont* para el servicio Ethernet.

```
onu-omci 0  
t-cont set slot-id 128 t-cont-id 0 alloc-id 810
```

4. La creación del *MAC Bridge Service Profile* y de los dos primeros *MAC Bridge Port Configuration Data* es idéntica a como se hizo para el servicio de Internet.

```
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 0 port-path-cost 1 port-spanning-tree-ind true encaps-method llc lanfcs-ind  
forward
```

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

5. La primera diferencia viene en la creación de un tercer *MAC Bridge Port Configuration Data*. Esta entidad irá vinculada a la entidad *Multicast GEM Interworking Termination* a través del campo *tp-type* (tipo del punto de terminación del *MAC Bridge*), que será de tipo multicast (opción *mc-gem*) y a través del número de instancia de la entidad *Multicast GEM Interworking Termination*, que debe coincidir con el campo *tp-ptr* (en este caso, 3).

```
mac-bridge-pcd create instance 3 bridge-id-ptr 1 port-num 3 tp-type mc-gem tp-ptr 3  
port-priority 0 port-path-cost 1 port-spanning-tree-ind true encaps-method llc lanfcs-ind  
forward
```

6. Creación de la entidad *GEM Port Network CTP* para el servicio Ethernet. El comando es el mismo que para el servicio de Internet.

```
gem-port-network-ctp create instance 2 port-id 800 t-cont-ptr 32768 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
```

7. Creación del *GEM Port Network CTP* para el multicast. Se vincula a la entidad *Multicast GEM Interworking Termination Point*. Para ello, el número de instancia de la entidad *GEM Port Network CTP* (en este caso, 3) debe coincidir con el del campo *gem-port-nwk-ctp-conn-ptr* del *Multicast GEM Interworking Termination Point*. El puerto GEM que define la entidad *GEM Port Network CTP* se asocia a

un puerto (por ejemplo, el 4094). Este puerto no es necesario vincularlo a un *Alloc-ID* porque el servicio de vídeo (servicio multicast) solamente hace uso del canal de bajada y el algoritmo DBA no influye para nada. Por eso, no es necesario asignar perfiles de ancho de banda a este servicio. A diferencia del *GEM Port Network CTP* asociado al servicio Ethernet, en este caso campo *direction* no es *bidirectional*, sino *ani-to-ani* ya que el servicio multicast solo utiliza el canal *Downstream*.

```
gem-port-network-ctp create instance 3 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
```

8. Creación de la entidad *GEM Interworking Termination Point* para el servicio Ethernet de la misma forma que se hizo en el Apartado 4.3.2.

```
gem-interworking-termination-point create instance 2 gem-port-nwk-ctp-conn-ptr 2  
interwork-option mac-bridge-lan service-profile-ptr 1 interwork-tp-ptr 0 gal-profile-ptr  
0
```

9. Creación de la entidad *Multicast GEM Interworking Termination Point*. Se vincula a las entidades *MAC Bridge Port Configuration Data* y *GEM Port Network CTP* (como se ve en la Figura 65) a través del número de instancia y del campo *gem-port-nwk-ctp-conn-ptr*, respectivamente (en ambos casos, el número es 3). El campo *interwork-option* debe ser del tipo *mac-bridge-lan* que en el punto anterior mientras que el campo *service-profile-ptr* en este caso no se usa y tiene como valor defecto 65535. El resto de campos toman los valores por defecto [32].

```
multicast-gem-interworking-termination-point create instance 3 gem-port-nwk-ctp-  
conn-ptr 3 interwork-option mac-bridge service-prof-ptr 65535 interwork-tp-ptr 0 gal-  
prof-ptr 65535 gal-lpbk-config 0
```

10. Configuración de las entidades relacionadas con los identificadores VLAN. No hay ninguna modificación con respecto a lo configurado en el servicio de Internet.

```
vlan-tagging-filter-data create instance 2 forward-operation h-vid-a vlan-tag1 833  
vlan-priority1 7 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-
```

```
tag9 null vlan-priority9 null vlan-tag10 null vlan-priority10 null vlan-tag11 null vlan-  
priority11 null vlan-tag12 null vlan-priority12 null
```

```
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 833 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  
833 treatment-inner-tpid tpid-de-copy-from-inner
```

```
exit
```

11. Finalmente, se lleva a cabo la configuración de las reglas VLAN en el OLT y la asignación de los perfiles de ancho de banda. En estos pasos, no hay que realizar ninguna modificación con respecto al Apartado 4.3.2 por el hecho de haber configurado un servicio multicast.

```
vlan uplink configuration port-id 800 min-cos 0 max-cos 7 de-bit disable primary-tag-  
handling false
```

```
vlan uplink handling port-id 800 primary-vlan none destination datapath c-vlan-  
handling no-change s-vlan-handling no-change new-c-vlan 0 new-s-vlan 0
```

```
policing downstream profile committed-max-bw 256000 committed-burst-size 1023  
excess-max-bw 51200 excess-burst-size 1023
```

```
policing downstream port-configuration entity port-id 800 ds-profile-index 0
```

```
exit
```

```
pon
```

```
dba pythagoras 0
```

```
sla 810 service data status-report nsr gr-bw 250 gr-fine 0 be-bw 300 be-fine 0
```

```
end
```

Una vez realizados todos estos pasos y con las interfaces WAN previamente configuradas (Apartado 4.3.2), queda configurado el servicio de Internet y vídeo. Al

igual que para el servicio de Internet, se puede crear más de un servicio de Internet y vídeo para lo cual basta con duplicar las entidades existentes y utilizar identificadores, puertos y *Alloc-IDs* diferentes. La única entidad que no es necesario replicar es el *MAC Bridge Service Profile*, ya que esta entidad define el tipo de perfil de servicio, que en todos los casos será un puente MAC (*MAC Bridge*).

Antes de concluir este apartado, hay que resaltar una diferencia significativa con el TGMS. En el TGMS era necesario configurar los canales multicast y asociarlos en paquetes ya que el TGMS filtra las direcciones multicast que llegan al OLT y solo deja pasar a las ONUs aquellas que están configuradas [4]. El CLI deja pasar todas las direcciones multicast de forma que no es necesario configurar canales. El único requisito es que las direcciones estén en el rango multicast permitido por la red GPON (224.0.0.2 – 239.255.255.255) [4].

#### **4.4.3 Pruebas y resultados de la configuración de un servicio de Internet y vídeo**

Este apartado no presenta ninguna diferencia con respecto al apartado de resultados del servicio de Internet y vídeo configurado con el TGMS. Las pruebas realizadas están recogidas en el Apartado 3.4.3.

### **4.5 Conclusiones**

El modo de gestión CLI es un sistema de gestión en línea de comandos basados en el protocolo GPON. Tiene una estructura basada en menús. En comparación con el TGMS, resulta más complicada la configuración de la red ya que la interfaz es mucho menos intuitiva y es necesario tener conocimientos del protocolo GPON. La ventaja del uso del CLI es que permite comprender el funcionamiento de este protocolo. El programador debe ser muy cuidadoso a la hora de introducir los comandos ya que cualquier mínimo fallo hace que el comando no sea válido y el CLI muestre un mensaje de error.

La principal diferencia con el TGMS es que el CLI no tiene persistencia por lo que la configuración de la red se pierde con cada reinicio/apagado de la misma. Además, los identificadores de ONU varían cada vez que arranca la red con lo que se hace necesario trabajar con las direcciones MAC en lugar de con los identificadores. Para

facilitar la gestión, en el Capítulo 5 se han realizado una serie de programas en Python y XML que tienen como objetivo paliar estas desventajas.

La forma de configurar servicios a través del CLI se lleva a cabo a través de la interfaz de gestión y control de las ONUs denominada OMCI. OMCI funciona con el protocolo MIB, basado en entidades. Estas entidades, relacionadas entre sí, vienen recogidas en el estándar OMCI [31]. Para la creación de servicios, hay que crear, en primer lugar, un canal OMCI vinculado a una ONU. Posteriormente, se deben crear las entidades para finalmente crear los perfiles de anchos de banda.

# 5

## Gestión de la red GPON utilizando Python y XML

### 5.1 Introducción

En este capítulo, se estudiarán las posibilidades que ofrecen los lenguajes de programación Python [6] y XML [7] para facilitar la configuración de la red GPON.

De cara a poder automatizar y agilizar la configuración de la red, el modo de gestión TGMS [4] no es útil ya que se trata de una interfaz que no funciona con comandos sino mediante la selección de opciones utilizando el ratón. Por tanto, no hay forma de realizar programas que automaticen su manejo. Por el contrario, el CLI [5] funciona mediante comandos (basados en entidades del estándar GPON), con lo que este modo de gestión sí puede ser utilizado para desarrollar programas que permitan configurar y gestionar de forma más rápida la red GPON.

De esta forma, será el CLI el modo de gestión con el que se trabajará en este capítulo. El lenguaje de programación Python [6] permite desarrollar programas denominados scripts (documentos que contienen instrucciones escritas en códigos de programación) para automatizar y agilizar el manejo de la red. Gracias a estos scripts, cualquier persona que maneje la red pueda configurarla sin necesidad de conocer en profundidad los comandos del CLI y las especificaciones del protocolo GPON.

Este capítulo se divide en tres partes. En el Apartado 5.2 se analizará cómo acceder y controlar el CLI desde Python. En el Apartado 5.3, se explicará el script de configuración general de servicios denominado *configuracion\_red\_GPON.py*, con los métodos (también denominados funciones) de los que consta, descritos en los diferentes subapartados. Finalmente, en el Apartado 5.4, se analizarán el resto de scripts, que

permiten agilizar ciertos aspectos relevantes de la gestión de la red GPON. Todos los scripts están en el Apéndice B comentados de forma oportuna.

## 5.2 Acceso al CLI desde Python

El entorno de desarrollo integrado utilizado para la programación en Python será *Spyder* en su versión Python 3.5 [33]. La principal funcionalidad de Python utilizada en los scripts es el módulo *telnetlib* [34]. Este módulo permite establecer conexiones de tipo Telnet como las que se realizan desde el símbolo del sistema de Windows (cmd) así como enviar comandos y recibir las respuestas del servidor. De esta forma, es posible acceder el CLI, introducir los comandos necesarios para gestionar la red y recibir las respuestas del CLI. Se debe seleccionar desde *Spyder* “Ejecutar -> Configuración -> Ajustes de Ejecución -> Ejecutar en una terminal de comandos del sistema”.

Para acceder al CLI es necesario crear una instancia de la clase *Telnet* del módulo *telnetlib*, que se debe importar al principio del programa junto al resto de módulos que se vayan a utilizar en el código. Se deberá especificar el nombre de la instancia (*tn*), la dirección IP y puerto del OLT (IP 172.26.128.38, puerto 4551 - definidos en el Capítulo 4) al que se realiza la conexión y un tiempo de espera máximo para la conexión en segundos (en este ejemplo, 5 segundos). La dirección IP debe ir entrecomillada debido a los puntos. La línea de código que permite el acceso al CLI “*tn = telnetlib.Telnet("172.26.128.38",4551,5)*” [34].

Una vez realizado el acceso, la forma de enviar comandos será con el uso del método (también se puede llamar función) *Telnet.write()* [34]. El comando o comandos CLI dirigidos al OLT se definirán mediante una cadena como puede “*comandos = "configure \n olt-device 0 \n olt-channel 0 \n show serial-number allocated"*”. Cada comando CLI debe ir separado por *\n* al final del comando. La cadena definida por el comando CLI se incluirá en el método del modo “*tn.write(comandos.encode('ascii') + b"\n")*”. Es necesario pasar a código ASCII la cadena mediante el método *encode('ascii')* así como introducir *b"\n"* para que no se produzcan problemas con el último comando ejecutado. Si se va a invocar varias veces el método *Telnet.write()* de forma seguida y con cadenas largas que impliquen varios comandos de CLI, es recomendable introducir un pequeño tiempo de espera (0,1–0,2 segundos es suficiente) entre cada llamada al

método. La línea de código que hace eso es *time.sleep(0.1)* y sirve para no saturar los buffers de la conexión. Es necesario por lo tanto importar el módulo *time*.

Para recibir los datos que aparecerían en la terminal del CLI se utiliza el método *Telnet.read\_very\_eager()* [34]. Estos datos se decodifican con el método *decode()* y se guardan en una variable. Esta variable se guarda en un fichero de texto con el método *open()*, que toma como argumentos el nombre del fichero y la opción de apertura. La opción '*r*' abre el fichero solo para lectura. La opción '*w*' abre un fichero para escribir solamente. Sobrescribe el fichero si ya existe y si no existe, se crea uno nuevo para escritura. La opción '*a*' también abre el fichero para escritura. Si el fichero existe, escribe los datos al final del mismo. En caso de no existir, lo crea y escribe los datos. Estas son las tres opciones que se utilizarán en los scripts. Finalmente, una vez abierto el fichero de la forma que se considere oportuna, se escriben los datos con el método *write()*. A continuación, se recogen las líneas de código que llevan a cabo estos pasos:

```
datos = tn.read_very_eager().decode()
outfile = open(nombre_archivo, 'a')
outfile.write(datos_perfil)
```

Otro método importante que se utilizará en los scripts es *Telnet.interact()* [34], que al ser invocado, muestra el terminal del CLI con todos los comandos ejecutados. Una vez invocado, se puede seguir configurando la red pero desde el terminal del CLI que acaba de aparecer. El script de Python no se seguirá ejecutando. Este método se utilizará para comprobar al final de un script que el proceso llevado a cabo ha sido el esperado o para visualizar algún tipo de información.

Como se describió en el Capítulo 4, la principal desventaja que presenta el modo de gestión CLI es la falta de persistencia. Cada vez que se apaga la red, la configuración que había en ese momento se pierde y al encenderla de nuevo, la red está sin configurar. Para solucionar este inconveniente, se utilizará el lenguaje XML [7] (*eXtensible Markup Language*), lenguaje de marcado extensible, en combinación con Python. Se creará un fichero XML en el directorio de trabajo de Python (donde están también los scripts), en el que se irá actualizando el estado de la red a medida que se vaya configurando. Este fichero estará organizado por ONUs y recogerá los servicios configurados dentro de su perfil así como sus parámetros. Cuando se produzca un nuevo arranque de la red, el

programador podrá recuperar la última configuración que había en la red, si así lo desea. Esta funcionalidad se explicará más en detalle en el Apartado 5.3.7.

## 5.3 Programa de configuración general de la red GPON

En este apartado, se estudiará el script que permite la configuración general de servicios y perfiles asociados en la red GPON y denominado con el nombre *configuracion\_red\_GPON.py*. Se analizarán los aspectos más relevantes del programa. El código comentado línea por línea está recogido en el Apéndice B.

Este programa permite a la persona que lo ejecuta configurar servicios en las diferentes ONUs de una manera muy sencilla, sin necesidad de conocer aspectos del CLI [5] ni del protocolo GPON [3]. Simplemente debe elegir el tipo de servicio e introducir los parámetros del mismo.

Es importante subrayar que se da por hecho que la configuración de los routers de las ONUs es la correcta (ya ha sido explicada en diferentes apartados a lo largo de la presente memoria) puesto que en estos momentos no se puede configurar mediante un script de Python.

### 5.3.1 Estructura general del programa

En primer lugar, el programa comprueba en el método *get\_ID\_ONU()* el número de ONUs conectadas a la red y almacena el número de serie de dichas ONUs (en la Figura 62 aparecen los números de serie de las 4 ONUs que pueden conectarse a la red). El número de serie está formado por los campos *Vendor ID* y *Vendor Specific* (este a su vez está formado por los últimos 8 valores hexadecimales de la dirección MAC) [10]. Estos números de serie se almacenarán en un lista (en Python se utilizan listas, no existen vectores) denominada por simplicidad *MAC*. La posición del número de serie de la ONU en dicha lista será el índice de la ONU. Este parámetro, explicado en el Capítulo 4, es de suma importancia y varía cada vez que arranca la red. Por ello, el programa lo primero que hace es la comprobación de ONUs conectadas y del índice de ONU que tienen. Además, este método comprueba si el fichero XML (explicado más adelante en el capítulo) que almacena las configuraciones existe. En caso de no existir, lo crea.

Una vez realizado este paso, el programa comienza a interactuar con el usuario, que deberá introducir el número de serie de la ONU que quiere configurar. Si el número de serie no es correcto, se le seguirá pidiendo hasta que introduzca un número de serie correspondiente a una ONU conectada a la red. Cuando el usuario introduzca un número de serie válido, se le podrán presentar dos menús de opciones dependiendo de si dicha ONU ya está configurada o no.

Para saber si una ONU está configurada o no, el programa comprobará si existe algún fichero en el directorio de trabajo con extensión *.txt* y con el número de serie de la ONU. Esta comprobación es posible con la importación de los métodos *os.path as path* y *os* que permiten saber si existe o no un fichero en el directorio de trabajo. Por ejemplo, en caso de existir el fichero “*Servicio\_Internet+Video\_ONU\_MAC\_54-4c-52-49-5b-01-f7-28*”, el programa sabría que la ONU con número de serie 54-4c-52-49-5b-01-f7-28 tiene configurado servicio de Internet y vídeo. Estos ficheros con extensión *.txt* se deben eliminar manualmente al apagarse la red ya que se pierde la configuración por la falta de persistencia del CLI. Sin embargo, los últimos parámetros de configuración sí quedarán recogidos en el fichero XML para poder recuperarlos la próxima vez que se encienda la red, si el usuario así lo desea. Es muy importante borrar esos ficheros de extensión *.txt* cada vez que se apague la red ya que de no hacerlo, al encender de nuevo la red, el programa considerará que la ONU está configurada cuando realmente no lo estará.

Es importante recalcar la diferencia entre los ficheros de texto que se generan automáticamente y el fichero XML. Los ficheros de texto permiten saber si una ONU está configurada y en caso de estarlo, conocer el tipo de servicio y sus parámetros ya que recogen los comandos del CLI introducidos para la configuración del servicio. Estos ficheros se deben eliminar cuando se apaga la red ya que se pierde la configuración. Por el contrario, el fichero XML (que se actualiza constantemente) guarda los parámetros de configuración de todas las ONUs conectadas. La función de este fichero es poder recuperar las configuraciones cuando se vuelve a encender la red.

Se comenzará analizando el caso de que la ONU no esté configurada. Si la ONU no está configurada, el usuario podrá realizar diferentes opciones:

- *Configurar servicio de Internet*. En este caso, el programa llamará al método *servicio\_Internet()*, que se explicará en el Apartado 5.3.2.

- Configurar servicio de Internet y vídeo. En este caso, el programa llamará al método *servicio\_Video()*, que se explicará en el Apartado 5.3.3.
- Cargar la última configuración activa. La ONU no está configurada y se desea recuperar la última configuración que tuvo antes de que se apagara la red. Este método se explicará en el Apartado 5.3.7.
- Volver al menú principal. Por menú principal se denomina al punto en el que se pide al usuario el número de serie de la ONU a configurar.
- Finalizar el programa.

En caso de que la ONU esté configurada, las opciones son las siguientes:

- Ver la configuración. Se llamará al método *ver\_configuracion()*, explicado en el Apartado 5.3.4.
- Borrar la configuración. Se llamará al método *borrar\_configuracion()*, explicado en el Apartado 5.3.5.
- Modificar la configuración. Se llamará al método *modificar\_configuracion()*, explicado en el Apartado 5.3.6.
- Volver al menú principal. Por menú principal se denomina al punto en el que se pide al usuario el número de serie de la ONU a configurar.
- Finalizar el programa.

Una vez finalizada la llamada a una función o método, el programa siempre permitirá al usuario finalizar el programa o continuar y volver al menú principal. El diagrama de flujo del programa es el mostrado en la Figura 66.

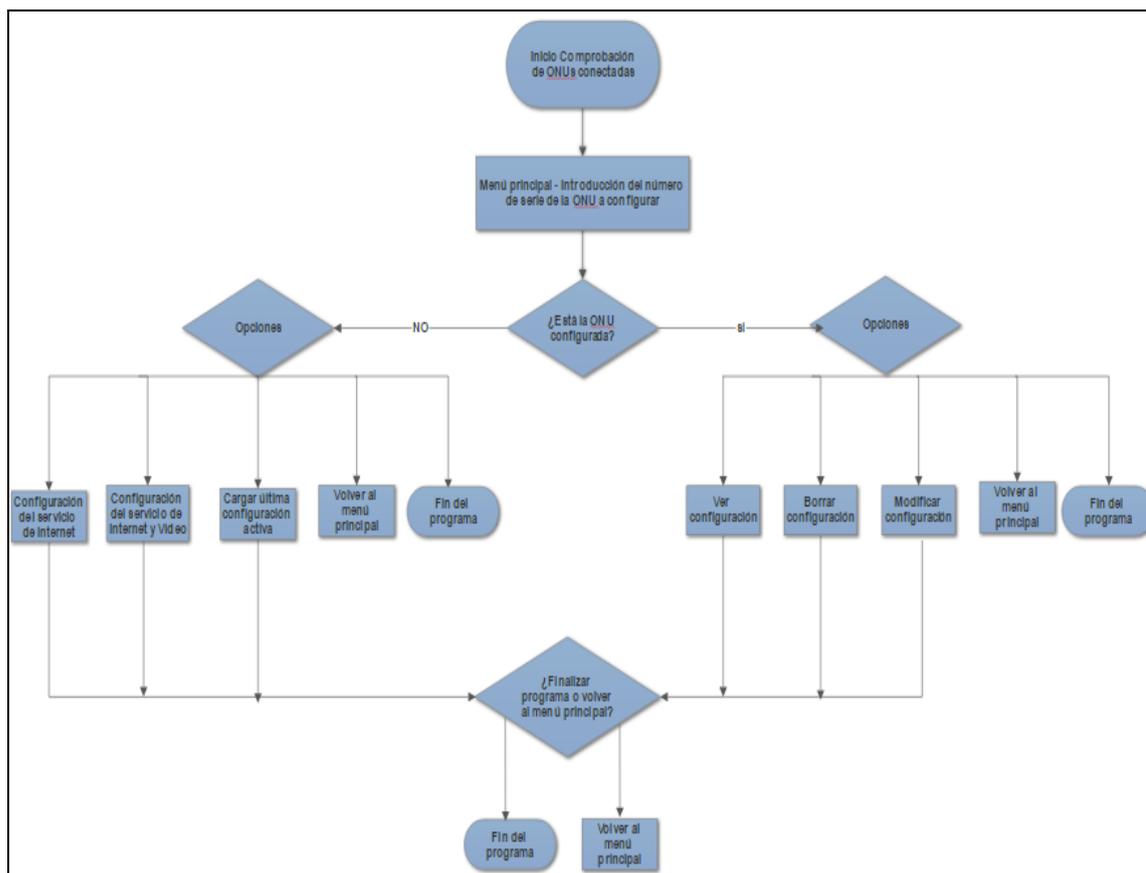


Figura 66: Diagrama de flujo del script *configuracion\_red\_GPON.py*

### 5.3.2 Configuración del servicio de Internet

La configuración del servicio de Internet se lleva a cabo en la función *servicio\_Internet(ID\_ONU,MAC\_ONU,num\_servicios\_Internet)*. Este método toma 3 argumentos: el identificador de la ONU (*ID\_ONU*), el número de serie (*MAC\_ONU*) definido, entre otros parámetros, por la dirección MAC y el número de servicios a configurar (*num\_servicios\_Internet*). El programa solo permite que el número de servicios sea 0 (no se configura ningún servicio), 1 o 2, ya que 2 son los identificadores VLAN operativos para esta red (806 y 833).

Esta función define los comandos del CLI en forma de cadenas y los escribe con la función *Telnet.write()*. Los comandos son los mismos que los explicados en el Apartado 4.3.2. Con la ayuda de variables y bucles, se asignan valores adecuados para los números de instancia y el resto de índices necesarios para la configuración. En el código que aparece en el Apéndice B, se explica de forma detallada esta forma de proceder.

Los parámetros de configuración, es decir, etiqueta VLAN, ancho de banda *Downstream* garantizado y en exceso y ancho de banda *Upstream* garantizado y *Best Effort*, son pedidos al usuario por pantalla. En caso de que el valor no sea correcto, el programa lanza un mensaje de aviso y vuelve a pedir el valor.

Una vez configurado el servicio, se notifica al usuario que el proceso se ha completado con éxito y se vuelcan los datos que aparecerían en el CLI a un fichero de texto en el que se especifica el tipo de servicio (Internet) y el número de serie (que se toma del argumento de la función). Por ejemplo, este fichero podría tener como nombre “*Servicio\_Internet\_ONU\_MAC\_54-4c-52-49-5b-01-f6-90.txt*”. Al crearse este fichero, el programa sabe que la ONU con ese número de serie tiene configurado un servicio de Internet. Al apagar la red, este fichero debe ser eliminado manualmente.

Este método también actualiza el fichero XML con los parámetros que se acaban de configurar. En el Apartado 5.3.7, se explicará la estructura de este fichero.

### **5.3.3 Configuración del servicio de Internet y Vídeo**

La configuración del servicio de Internet y vídeo se lleva a cabo en la función *servicio\_Video(ID\_ONU,MAC\_ONU,num\_servicios\_Video)*. Este método toma 3 argumentos, el identificador de la ONU, el número de serie (definido, entre otros parámetros, por la dirección MAC) y el número de servicios a configurar. El programa solo permite que el número de servicios sea 0 (no se configura ningún servicio), 1 o 2, ya que 2 son los identificadores VLAN operativos para esta red (806 y 833).

Esta función define los comandos del CLI en forma de cadenas y los escribe con la función *Telnet.write()*. Los comandos son los mismos que los explicados en el Apartado 4.4.2. Con la ayuda de variables y bucles, se asignan valores adecuados para los números de instancia y el resto de índices necesarios para la configuración. En el código que aparece en el Apéndice B, se explica de forma detallada esta forma de proceder.

Los parámetros de configuración que el usuario debe introducir son los mismos que en el servicio de Internet ya que el servicio de tipo multicast, que es el asociado al

vídeo, no tiene ancho de banda asociado y solamente es necesario definir los anchos de banda del servicio Ethernet (servicio de Internet) así como el identificador VLAN.

Una vez configurado el servicio, se notifica al usuario que el proceso se ha completado con éxito. Se guardan los datos que aparecerían en el CLI y se almacenan en un fichero de texto en el que se especifica el tipo de servicio (Internet y vídeo) y el número de serie (que se toma del argumento de la función). Por ejemplo, este fichero podría tener como nombre “*Servicio\_Internet+Video\_ONU\_MAC\_54-4c-52-49-5b-01-f6-d8.txt*”. Al crearse este fichero, el programa sabe que la ONU con ese número de serie tiene configurado servicio de Internet y vídeo. Al apagar la red, este fichero debe ser eliminado manualmente.

Este método también actualiza el fichero XML con los parámetros que se acaban de configurar. En el Apartado 5.3.7, se explicará la estructura de este fichero.

### 5.3.4 Ver configuración

La función *ver\_configuracion(MAC\_ONU)* permite ver la configuración asociada a una ONU. Este método toma como argumento el número de serie de la ONU (la variable se denomina *MAC\_ONU* por simplicidad ya la diferencia entre los números de serie de las 4 ONUs es debida a la dirección MAC). El programa busca en el fichero XML el número de serie y muestra por pantalla al usuario el tipo de servicio, el número de servicios configurados y los parámetros de configuración asociados a dicho número de serie. Un ejemplo de la configuración mostrada a través de esta función es la Figura 67.

```
Tipo de servicio: Internet
Número de servicios: 1
Parámetros:
El identificador VLAN del servicio es 833
El ancho de banda garantizado Downstream en Kbps del servicio es 200000
El ancho de banda en exceso Downstream en Kbps del servicio es 0
El ancho de banda garantizado Upstream en Mbps del servicio es 200
El ancho de banda Best Effort Upstream en Mbps del servicio es 200
```

Figura 67: Ejemplo de configuración de los servicios configurados en una ONU

### 5.3.5 Borrar configuración

La función *borrar\_configuracion(ID\_ONU,MAC\_ONU,nombre\_archivo)* permite borrar la configuración asociada a una ONU. Toma como parámetros el identificador de ONU y su número de serie así como el nombre del fichero de texto donde está guardada

la configuración, es decir, donde está el historial de comandos del CLI que fueron necesarios para realizar el proceso de configuración.

Se abre el fichero de texto en modo lectura y se busca mediante un bucle el puerto o puertos asociados a los servicios configurados. Cada vez que se encuentre un puerto, se borrará el perfil de ancho de banda descendente asociado a dicho puerto. Una vez realizado este paso, se borrarán las entidades creadas en el canal OMCI asociado a la ONU en cuestión. Finalmente, se eliminará el fichero de texto (ahora la ONU está sin configurar) con el método *remove()*.

### **5.3.6 Modificar configuración**

La función *modificar\_configuracion(ID\_ONU,MAC\_ONU,nombre\_archivo)* permite modificar la configuración de una ONU. Toma como argumentos el identificador de ONU, el número de serie y el nombre del fichero de texto donde están guardados los comandos CLI del proceso de configuración.

En primer lugar, la función borra la configuración de la ONU de la misma forma que se hace en la función *borrar\_configuracion()* y tras esto, el programa pide al usuario el nuevo de tipo de servicio que desea configurar. En caso de que seleccione servicio de Internet, se llamará al método *servicio\_Internet()* y en caso de que escoja servicio de Internet y vídeo, invocará al método *servicio\_Video()*. La configuración del nuevo servicio se realiza desde esas funciones así como la actualización del fichero XML y de los ficheros de texto que guardan los comandos CLI.

### **5.3.7 Cargar configuración utilizando XML**

La función *cargar\_configuracion(ID\_ONU,MAC\_ONU)* permite recuperar la última configuración activa de una ONU antes de que la red fuera apagada. Toma como argumentos el identificador y el número de serie de la ONU.

Para poder recuperar la configuración, se utiliza un fichero XML. Es necesario importar un módulo para la creación y modificación de este fichero: *import xml.etree.cElementTree as etree* (transparente al usuario) [35]. Este módulo permitirá trabajar con el fichero como si fuera un árbol con diferentes niveles que parten de un elemento raíz. El fichero se crea, en caso de no existir, en el método *get\_ID\_ONU()* y

está organizado en diferentes niveles. El primer nivel es general y se denomina *RedGPON* (elemento raíz). El segundo nivel está formado por las ONUs. El valor de atributo de este segundo nivel es el número de serie de las ONUs y es lo que permite acceder a una ONU u otra (*ONU MAC*). El tercer nivel está formado por el tipo de servicio (*Servicio tipo*), con un valor de atributo que puede ser “*Internet*” o “*Internet+Video*”. Finalmente, en el último nivel, están los parámetros de configuración de los servicios en forma de elementos. El valor de estos elementos es el valor del parámetro asociado [36]. La Figura 68 muestra un ejemplo del fichero de configuración XML. En este ejemplo, se puede apreciar cómo solo la ONU con número de serie *54-4c-52-49-5b-01-f6-d8* está configurada.

```

<?xml version="1.0"?>
- <RedGPON>
  <ONU MAC="54-4c-52-49-5b-01-f6-d8"/>
  <ONU MAC="54-4c-52-49-5b-01-f6-90"/>
  - <ONU MAC="54-4c-52-49-5b-01-f7-28">
    - <Servicio tipo="Internet+Video">
      <VLAN_ID>833</VLAN_ID>
      <BW_Down_GR>19968</BW_Down_GR>
      <BW_Down_Excess>0</BW_Down_Excess>
      <BW_Up_GR>20</BW_Up_GR>
      <BW_Up_BE>20</BW_Up_BE>
    </Servicio>
    - <Servicio tipo="Internet+Video">
      <VLAN_ID>806</VLAN_ID>
      <BW_Down_GR>19968</BW_Down_GR>
      <BW_Down_Excess>0</BW_Down_Excess>
      <BW_Up_GR>20</BW_Up_GR>
      <BW_Up_BE>20</BW_Up_BE>
    </Servicio>
  </ONU>
  <ONU MAC="54-4c-52-49-5b-01-f7-30"/>
</RedGPON>

```

Figura 68: Fichero de configuración XML

Este fichero XML se debe actualizar en todos los métodos del programa en los que se produzcan cambios en la configuración (*servicio\_Internet()*, *servicio\_Video()*, *borrar\_configuracion()*, *modificar\_configuracion()*), ya que tiene que contener la configuración activa en todas las ONUs para que cuando se apague la red y se vuelva a encender, se pueda recuperar la última configuración que había en las ONUs.

Al encender de nuevo la red, el usuario introducirá el número de serie de la ONU cuya última configuración quiere recuperar y seleccionará la opción “*Cargar la última configuración activa*”. La función *cargar\_configuracion(ID\_ONU,MAC\_ONU)* obtiene del fichero XML el número y tipo de servicios y sus parámetros de configuración. Posteriormente, lleva a cabo el proceso de configuración de la misma manera que en las funciones *servicio\_Internet()* o *servicio\_Video()* (dependiendo del tipo de servicio que hubiera configurado) con la única diferencia de que en este caso, no es necesario pedir los parámetros por pantalla. Al finalizar el proceso de cargar la configuración, se muestra un aviso de que ha resultado con éxito y se le da la posibilidad al usuario de ver la configuración cargada (para lo cual se llama al método *ver\_configuracion()*).

## 5.4 Otros scripts para la gestión de la red GPON

En este apartado, se analizarán el resto de programas que permiten agilizar la gestión de la red GPON. La ejecución de estos scripts evita tener que introducir las direcciones y puertos de la conexión así como las claves de acceso y otros comandos. Los scripts comentados están recogidos en el Apéndice B.

### 5.4.1 Acceso al CLI

El script *acceso\_CLI.py* permite acceder directamente al menú con privilegios (*Privileged Mode*) del CLI.

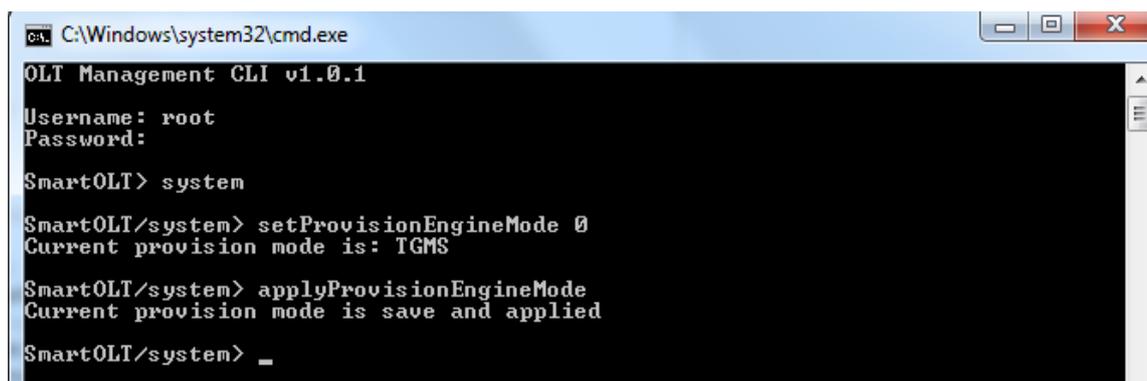
El programa realiza la conexión de tipo Telnet a la dirección IP del OLT (172.26.128.38) y puerto de acceso al CLI (4551) a través de la clase *Telnet* del módulo *telnetlib* [34]. Posteriormente, se introducen las claves de acceso y los comandos necesarios (ya definidos en el Capítulo 4) con el método *Telnet.write()*. Finalmente, el método *Telnet.interact()* permite que aparezca el terminal del CLI, desde el que el usuario puede introducir los comandos CLI. De esta forma, se evita que el usuario tenga que introducir las claves de acceso y los comandos cada vez que quiera acceder al CLI.

### 5.4.2 Activación de los modos de gestión TGMS / CLI

Los scripts *activar\_TGMS.py* y *activar\_CLI.py* permiten activar un modo u otro de gestión de la red GPON simplemente mediante la ejecución del script.

Son scripts muy similares. El programa realiza la conexión de tipo Telnet a la dirección IP del OLT (172.26.128.38) y puerto de configuración del modo de gestión (4555) a través de la clase *Telnet* del módulo *telnetlib* [34]. Posteriormente, se introducen las claves de acceso y los comandos necesarios (ya definidos en el Capítulo 4) con el método *Telnet.write()*. El comando *setProvisionEngineMode* seguido de un 0 activa el TGMS, por que aparecerá en el script *activar\_TGMS.py* mientras que seguido de un 1 activa el CLI y por tanto, aparecerá en el script *activar\_CLI.py*. Esta es la única diferencia entre ambos scripts.

Simplemente con la ejecución del script correspondiente, se activa el modo de gestión seleccionado (el proceso de activación tarda 1-2 minutos en completarse). Si el usuario desea ver los comandos introducidos durante el proceso, puede utilizar la función *Telnet.interact()* de forma que en el ejemplo del TGMS, aparecería el terminal con los comandos, tal y como se muestra en la Figura 69.



```
C:\Windows\system32\cmd.exe
OLT Management CLI v1.0.1
Username: root
Password:
SmartOLT> system
SmartOLT/system> setProvisionEngineMode 0
Current provision mode is: TGMS
SmartOLT/system> applyProvisionEngineMode
Current provision mode is save and applied
SmartOLT/system> _
```

Figura 69: Activación del modo de gestión TGMS

### 5.4.3 Mostrar modo de gestión activo

El script *mostrar\_modos\_funcionamiento.py* permite comprobar cuál es el modo de gestión de la red GPON que está activo en ese momento. Puede resultar útil si al encender la red, no se recuerda cuál era el modo activo en el momento del apagado.

El script es muy similar al anterior ya que se realiza la misma conexión Telnet. La diferencia es que el comando utilizado es *showProvisionEngineMode*. En este caso, sí es necesario usar el método *Telnet.interact()* [34] ya que es necesario ver en el terminal el modo activo. La Figura 70 muestra el ejemplo del resultado de la ejecución de este script. En este caso, el modo de gestión activo es el CLI.

```

C:\Windows\system32\cmd.exe
OLT Management CLI v1.0.1
Username: root
Password:
SmartOLT> system
SmartOLT/system> showProvisionEngineMode
Current provision mode is: CLI
SmartOLT/system>
    
```

Figura 70: Visualización del modo de gestión de la red GPON activo

#### 5.4.4 Mostrar ONUs conectadas

El script *mostrar\_ONUs\_conectadas.py* permite saber el número de ONUs conectadas a la red GPON y conocer su número de serie y el identificador de ONU asociado. Resulta útil para detectar si hay algún problema de conexión en alguna ONU así como para conocer el identificador de las ONUs, parámetro fundamental para la configuración de servicios en las ONUs y que varía cada vez que se enciende la red.

El programa realiza la conexión de tipo Telnet a la dirección IP del OLT (172.26.128.38) y puerto de acceso al CLI (4551) a través de la clase *Telnet* del módulo *telnetlib* [34]. Posteriormente, se introducen las claves de acceso y los comandos necesarios (ya definidos en el Capítulo 4). Para conocer las ONUs que están conectadas a la red, su número de serie y los identificadores de ONU asociados, hay que introducir desde el menú de canal de OLT (*OLT channel configuration*) el comando *show serial-number allocated*. La introducción de las claves de acceso y los comandos se realiza con el método *Telnet.write()*. Finalmente, se debe utilizar el método *Telnet.interact()* para que aparezca el terminal del CLI y poder visualizar la información de las ONUs tal y como aparece en la Figura 62.

## 5.5 Conclusiones

En toda red de comunicaciones se busca agilizar y automatizar el proceso de gestión y configuración de la misma. En esta red GPON, el modo de gestión TGMS no es útil para desarrollar estos avances ya que su interfaz se maneja mediante la selección de opciones con el ratón. Por el contrario, el modo de gestión CLI sí permite desarrollar

programas para facilitar la labor de gestión y configuración ya que se basa en comandos fáciles de automatizar.

Gracias al lenguaje de programación Python [6], se ha podido desarrollar un programa general que permite configurar servicios, tanto de Internet como de Internet y vídeo, en cualquier ONU conectada a la red. El programa detecta automáticamente las ONUs que se encuentran conectadas y el usuario solamente tiene que introducir el número de serie de la ONU y seleccionar el tipo de servicio que desea configurar. No es necesario que el usuario conozca los comandos del CLI ni el protocolo GPON ya que lo único que el programa le pide son los parámetros del servicio a configurar. Además, el usuario puede en todo momento ver, borrar y modificar la configuración de las ONUs.

Este programa también permite paliar de algún modo la principal desventaja del CLI, la falta de persistencia. Para ello, se utilizará el lenguaje XML [7]. La configuración de las ONUs se irá guardando en un fichero XML y se actualizará con cada modificación. Cuando se produzca un nuevo arranque en la red, el usuario podrá elegir recuperar la última configuración activa ya que el programa podrá recuperar los parámetros de configuración de dicho fichero de forma automática y transparente a través de un script.

Además de este programa de configuración general, se pueden crear scripts que faciliten otras tareas de gestión de la red como son: acceso directo al CLI sin necesidad de introducir contraseñas, activación de cualquiera de los modos de gestión (TGMS/CLI), mostrar el modo de gestión activo y mostrar las ONUs conectadas y su información más relevante. Estos scripts permiten ahorrar mucho tiempo a la persona que está manejando la red GPON.



# 6

## Conclusiones y Líneas Futuras

### 6.1 Conclusiones

En este Trabajo Fin de Grado, se ha llevado a cabo la configuración de servicios en la red de acceso GPON situada en el laboratorio de Comunicaciones Ópticas (2L007). Tras un análisis a nivel físico de la topología y los componentes que forman la red, la principal motivación de este estudio ha sido investigar las posibilidades de configuración que ofrece la red.

Para llevar a cabo esta tarea, en primer lugar se ha realizado un análisis a nivel físico de la red GPON. Este análisis, recogido en el Capítulo 2, ha permitido conocer la topología en árbol de la red, con un elemento central denominado OLT (*Optical Line Termination*), que se situaría en las dependencias del operador, y varios equipos de usuario denominados ONUs/ONTs (*Optical Network Unit/Terminal*). En esta red, se dispone de 4 ONUs aunque se podrían llegar a conectar hasta 256 al OLT. Se utiliza un *splitter* o divisor óptico para hacer que la señal óptica llegue a las diferentes ONUs. De esta forma, se generan dos canales de comunicación definidos en el protocolo GPON, es decir, el canal descendente (*Downstream*), que va desde el OLT hacia las ONUs y en el que se pueden alcanzar tasas de hasta 2,488 Gbps y el canal ascendente (*Upstream*), que va de las ONUs hacia el OLT, con tasas de hasta 1,244 Gbps.

Una vez realizado el análisis a nivel físico, el estudio se centró en la configuración de servicios. Se han configurado dos tipos de servicios, servicio de Internet y servicio de Internet y vídeo.

Para llevar a cabo esta tarea, dentro del OLT existen dos modos de gestión de la red GPON: el TGMS (*TELNET GPON Management System*) y el CLI (*Command Line Interface*). En primer lugar, se analizó el TGMS en el Capítulo 3. Este modo consiste en

un sistema web de gestión con una interfaz muy visual e intuitiva. Permite gestionar y configurar la red abstrayéndose por completo del protocolo GPON. La forma de configurar servicios se basa en la definición de anchos de banda y mapeados VLAN que se asocian a los servicios. Finalmente, uno o más servicios se agrupan en perfiles de usuario que posteriormente se asignan a las ONUs.

En el Capítulo 4, se analizó la configuración de la red a través del otro modo de gestión, el CLI. El CLI es una interfaz a la cual se accede mediante una conexión Telnet y que está basada en línea de comandos. Estos comandos están relacionados con el protocolo GPON de manera que trabajar con este modo de gestión permite comprender el funcionamiento del protocolo. La configuración de servicios se lleva a cabo a través de la interfaz de gestión y control de ONUs denominada OMCI (*ONU Management and Control Interface*). OMCI funciona con el protocolo MIB (*Management Information Base*), que se basa en entidades. La definición de estas entidades mediante comandos CLI permite la configuración de los servicios.

Finalmente, en el Capítulo 5, se han desarrollado una serie de programas con los lenguajes de programación Python y XML para agilizar y automatizar la gestión de la red y la configuración de servicios en la misma. Esto es posible utilizando el modo de gestión CLI ya que el TGMS no permite esta automatización de los procesos al no estar basado en comandos. Se ha desarrollado un programa general en Python que permite configurar servicios en las ONUs conectadas a la red de una forma muy sencilla, es decir, el usuario solamente debe seleccionar la ONU que quiere configurar e introducir los parámetros del servicio. Este programa palía en cierto modo el principal problema del CLI, esto es, la falta de persistencia, ya que al apagar la red se pierde la configuración. Para ello, se utiliza el lenguaje XML mediante un fichero que guarda las últimas configuraciones de las ONUs y permite su posterior recuperación. Además de este programa de configuración global, se han realizado scripts que con simplemente ejecutarlos, llevan a cabo otras tareas de gestión. En concreto tareas para el acceso directo al CLI, activación de un modo de gestión u otro, visualización del modo de gestión activo y visualización de las ONUs conectadas a la red.

## 6.2 Líneas Futuras

La implementación de sistemas de gestión que permiten controlar la red de acceso GPON de manera más eficiente mediante el uso de tecnologías más vanguardistas y con posibilidades de desarrollo ha propiciado la aparición de nuevas líneas de investigación, que se pueden seguir una vez concluido este Trabajo de Fin de Grado.

Estas líneas futuras de trabajo se centran en el uso de *OpenFlow* [37]. *OpenFlow* es un protocolo que permite a un servidor determinar el camino de reenvío de paquetes que debería seguir en una red de switches.

Este protocolo, utilizado en esta red, podría permitir controlar la configuración de la red GPON mediante un controlador que actuara directamente sobre OLT usando dicho estándar *OpenFlow*. El objetivo sería instalar un controlador *OpenFlow* que sea capaz de enviar órdenes (con datos de configuración de servicios y ONUs) y programar en el lado del OLT un agente *OpenFlow*, cuya misión sea traducir estos mensajes *OpenFlow* al lenguaje que entiende el OLT, es decir, a comandos CLI. Este agente se puede programar en Python y haría de intermediario, es decir, sería el encargado de comunicarse con el controlador *OpenFlow* por un lado, traducir la información a comandos CLI y comunicarse con el CLI para volcar los datos de configuración sobre la red GPON. Este último paso se va a llevar a cabo utilizando programa de configuración general ya desarrollado en Python (Apartado 5.3). También se podría utilizar el protocolo *OpenFlow* para el control de las ONUs. Para ello, habría que valorar la opción de controlar los routers de las ONUs a través de comandos que se puedan automatizar con programas en Python. De esta forma, se podría programar un agente *OpenFlow* que se comunicara tanto con el controlador *OpenFlow* como con las ONUs.



# 7

## Bibliografía

- [1] Página web oficial de la Escuela Técnica Superior de Ingenieros de Telecomunicación (ETSIT) de la Universidad de Valladolid. <http://www.tel.uva.es/>
- [2] Página web oficial de TELNET Redes Inteligentes. <http://www.telnet-ri.es/>
- [3] International Telecommunication Union, “G.984.1: Gigabit-capable passive optical networks (GPON): General characteristics”. Disponible en: <http://www.itu.int/rec/T-REC-G.984.1-200803-I>
- [4] “TGMS: Manual de Usuario”, TELNET Redes Inteligentes
- [5] “Manual CLI Académico”, TELNET Redes Inteligentes
- [6] Página web oficial del lenguaje de programación Python. <https://www.python.org/>
- [7] Página web del desarrollador de XML. <https://www.w3.org/XML/>
- [8] A. García Yagüe, “GPON: Introducción y conceptos generales”, TELNET Redes Inteligentes, Noviembre 2012.
- [9] “Red de Operador. Arquitectura y Servicios”, TELNET Redes Inteligentes.
- [10] “SmartOLT: Guía de Instalación y Configuración”, TELNET Redes Inteligentes.
- [11] Hoja de Producto, “SmartOLT 350”, TELNET Redes Inteligentes.
- [12] Hoja de Producto, “WaveAccess 3021”, TELNET Redes Inteligentes.
- [13] “WaveAccess Series 30xx/40xx. Manual de Usuario Avanzado”, TELNET – Redes Inteligentes.

- [14] International Telecommunication Union, “G.984.2: Gigabit-capable Passive Optical Networks (G-PON): Physical Media Dependent (PMD) layer specification. Amendment 1”. Disponible en: <https://www.itu.int/rec/T-REC-G.984.2-200602-I!Amd1/es>
- [15] I. de Miguel Jiménez y N. Merayo Álvarez, “Componentes de los Sistemas de Comunicaciones Ópticas”.
- [16] International Telecommunication Union, “G.652: Characteristics of a single-mode optical fibre and cable”. Disponible en: <https://www.itu.int/rec/T-REC-G.652-201611-I/es>
- [17] N. Merayo Álvarez y J.C. Aguado Manzano, “Comprobación de Instalaciones de fibra óptica con OTDR”.
- [18] Página web oficial de Fibercom. <http://www.fibercom.es/inicio/>
- [19] Página web oficial de Haktronics. <https://hacktronics.co.in/>
- [20] Página web de Virtual Box. <https://www.virtualbox.org/>
- [21] IEEE Std 802.1Q - 2014, “IEEE Standard for Local and metropolitan area networks — Bridges and Bridged Networks”. Disponible en: <http://standards.ieee.org/about/get/802/802.1.html>
- [22] J.V. Capella Hernández, “Características y configuración básica de VLANs”, Universidad Politécnica de Valencia.
- [23] RFC 2131, “DHCP (Dynamic Host Configuration Protocol)”, March 1997. Disponible en: <http://www.rfc-base.org/rfc-2131.html>
- [24] M. Feknous, A. Gravey, B. Le Guyader and S. Gosellin, “Status Reporting versus Non Status Reporting Dynamic Bandwidth Allocation”
- [25] M. L. Bote Lorenzo, “Ingeniería de Protocolos en Redes Telemáticas – La capa de red: direccionamiento y reenvío”, E.T.S.I. de Telecomunicación, Universidad de Valladolid, 2016.

- [26] Página web oficial de AMD (Advanced Micro Devices). <https://www.amd.com/es/home>
- [27] Página web oficial de HP (Hewlett-Packard). <http://www8.hp.com/es/es/home.html>
- [28] RFC 3376, “IGMP (Internet Group Management Protocol)”, October 2002. Disponible en: <https://www.rfc-editor.org/info/rfc3376>
- [29] VLC Media Player. Disponible en: <https://www.videolan.org/vlc/>
- [30] Documentación VLC Media Player. Disponible en: <https://wiki.videolan.org/Documentation:Documentation/>
- [31] International Telecommunication Union, “G.988: ONU management and control interface (OMCI) specification.” Octubre 2012. Disponible en: <https://www.itu.int/rec/T-REC-G.988-201210-I>
- [32] “Scripts CLI Académico”, TELNET Redes Inteligentes
- [33] “Spyder – Documentation”. Disponible en: <http://pythonhosted.org/spyder/>
- [34] “Python – Documentation – The Python Standard Library: *telnetlib* – Telnet Client”. Disponible en: <https://docs.python.org/3.5/library/telnetlib.html>
- [35] “Python – Documentation – The Python Standard Library: *xml.etree.ElementTree* — The ElementTree XML API”. Disponible en: <https://docs.python.org/3.5/library/xml.etree.elementtree.html>
- [36] “Área de Programación y Desarrollo. Manual de XML”. Disponible en: <http://www.mundolinux.info/que-es-xml.htm>
- [37] Página web del protocolo *OpenFlow*. <https://www.opennetworking.org/en/sdn-resources/openflow>



# Apéndice A: Ficheros de configuración del CLI

En este apéndice, se recogen dos ejemplos de ficheros de configuración que recogen los comandos que hay que introducir en el CLI para la configuración de servicios. En ambos casos, se configurará el servicio en la ONU con identificador de ONU 0; la forma de operar en las ONUs con otro identificador es análoga. Estos ejemplos han sido analizados paso por paso en el Capítulo 4. En este apéndice, se mostrará la lista de comandos sin entrar en tanto detalle. Las relaciones entre las entidades creadas vienen explicadas el Capítulo 4. Los comentarios aparecen en negrita tras el carácter #; el resto de líneas son comandos CLI.

El primer ejemplo es el de configuración del servicio de Internet y el segundo ejemplo es el de configuración del servicio de Internet y vídeo.

## Ejemplo de configuración del servicio de Internet en la interfaz del CLI

**# Comandos necesarios para empezar a configurar el servicio de Internet en la ONU con id 0 utilizando el VLAN id 833. Se toma como punto de partida el menú con privilegios (Privileged Mode).**

```
configure
olt-device 0
olt-channel 0
onu-local 0
```

**# Creación en la ONU con id 0 el canal OMCI 0, siguiendo la convención de utilizar el mismo identificador para la ONU y su canal OMCI asociado**

```
omci-port 0
exit
onu-omci 0
```

**# Borrado todas las MIB presentes para que no haya problemas con posibles configuraciones anteriores**

```
ont-data mib-reset
exit
```

**# Activación del FEC en el enlace ascendente para la ONU con onu-id 0**

```
fec direction uplink 0
onu-local 0
```

**# Creación del Alloc-ID 810 en la ONU con onu-id 0**

```
alloc-id 810
```

exit

**# Asignación del Alloc-ID 810 al Port-ID 800**

port 800 alloc-id 810

**# Desde el menu onu-omci 0 se crearán las entidades necesarias para el correcto funcionamiento del servicio de Internet**

onu-omci 0

**# Asignación del Alloc-ID 810 con T-Cont 0.**

t-cont set slot-id 128 t-cont-id 0 alloc-id 810

**# Creación de MAC Bridge Service Profile.**

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

**# Creación del primer MAC Bridge Port Configuration Data.**

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

**# Creación del segundo Mac Bridge Port Configuration Data.**

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

**# Creación del GEM Port Network CTP para el Port-ID 800.**

gem-port-network-ctp create instance 2 port-id 800 t-cont-ptr 32768 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

**# Creación del GEM Interworking Termination Point.**

gem-interworking-termination-point create instance 2 gem-port-nwk-ctp-conn-ptr 2 interwork-option mac-bridge-lan service-profile-ptr 1 interwork-tp-ptr 0 gal-profile-ptr 0

**# Creación del VLAN Tagging Filter Data con el identificador VLAN 833.**

vlan-tagging-filter-data create instance 2 forward-operation h-vid-a vlan-tag1 833 vlan-priority1 7 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

**# Creación de Extended VLAN Tagging Operation Config Data, que será configurado en el paso posterior. Sirve para gestionar los identificadores VLAN.**

extended-vlan-tagging-operation-config-data create instance 257 association-type ptp-eth-uni associated-me-ptr 257

**# Configuración de Extended VLAN Tagging Operation Config Data. Se debe configurar para cada identificador VLAN.**

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 833 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 833 treatment-inner-tpid tpid-de-
copy-from-inner
```

**# Salida al menú del canal 0 de la OLT 0**

```
exit
```

**# Configuración de las reglas VLAN en la OLT asociadas al puerto 800**

```
vlan uplink configuration port-id 800 min-cos 0 max-cos 7 de-bit disable primary-tag-handling
false
vlan uplink handling port-id 800 primary-vlan none destination datapath c-vlan-handling no-
change s-vlan-handling no-change new-c-vlan 0 new-s-vlan 0
```

**# Creación del perfil de ancho de banda en el sentido de bajada.**

```
policing downstream profile committed-max-bw 256000 committed-burst-size 1023 excess-max-
bw 51200 excess-burst-size 1023
```

**# Asignación del perfil de ancho de banda recién creado al Port-ID 800 (el ds-profile-index se mostrará en el paso anterior).**

```
policing downstream port-configuration entity port-id 800 ds-profile-index 0
exit
```

**# Acceso al menú de la red PON.**

```
pon
```

**# Selección del menú del algoritmo DBA de la PON 0.**

```
dba pythagoras 0
```

**# Asignación del ancho de banda en el sentido de subida para el Alloc-ID 810.**

```
sla 810 service data status-report nsr gr-bw 50 gr-fine 0 be-bw 100 be-fine 0
end
```

## Ejemplo de configuración del servicio de Internet y vídeo en la interfaz del CLI

**# Comandos necesarios para empezar a configurar el servicio de Internet y vídeo en la ONU con id 0 utilizando el VLAN id 833. Se toma como punto de partida el menú con privilegios (Privileged Mode).**

```
enable
configure
olt-device 0
olt-channel 0
onu-local 0
```

**# Creación en la ONU con id 0 el canal OMCI 0, siguiendo la convención de utilizar el mismo identificador para la ONU y su canal OMCI asociado**

```
omci-port 0
exit
onu-omci 0
```

**# Borrado todas las MIB presentes para que no haya problemas con posibles configuraciones anteriores**

```
ont-data mib-reset
exit
```

**# Activación del FEC en el enlace ascendente para la ONU con onu-id 0**

```
fec direction uplink 0
onu-local 0
```

**# Creación del Alloc-ID 810 en la ONU con onu-id 0**

```
alloc-id 810
exit
```

**# Asignación del Alloc-ID 810 al Port-ID 800**

```
port 800 alloc-id 810
```

**# Desde el menu onu-omci 0 se crearán las entidades necesarias para el correcto funcionamiento del servicio de Internet y vídeo**

```
onu-omci 0
```

**# Asignación del Alloc-ID 810 con T-Cont 0.**

```
t-cont set slot-id 128 t-cont-id 0 alloc-id 810
```

**# Creación del Mac Bridge Service Profile**

```
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
```

**# Creación de los Mac Bridge Port Configuration Data. Los dos primeros corresponden al servicio Ethernet (de Internet). El tercero sirve para configurar el servicio multicast (de video).**

```
126
```

```
mac-bridge-pcd create instance 1 bridge-id-ptr 1 port-num 1 tp-type lan tp-ptr 257 port-  
priority 0 port-path-cost 1 port-spanning-tree-ind true encap-method llc lanfcs-ind forward
```

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

```
mac-bridge-pcd create instance 3 bridge-id-ptr 1 port-num 3 tp-type mc-gem tp-ptr 3 port-  
priority 0 port-path-cost 1 port-spanning-tree-ind true encap-method llc lanfcs-ind forward
```

**# Creación del Gem Port Network CTP para el servicio Ethernet (servicio de Internet)**

```
gem-port-network-ctp create instance 2 port-id 800 t-cont-ptr 32768 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
```

**# Creación del GEM Port Network CTP para el servicio multicast utilizando el Port-ID 4094.**

```
gem-port-network-ctp create instance 3 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
```

**# Creación del GEM Interworking Termination Point para el servicio Ethernet.**

```
gem-interworking-termination-point create instance 2 gem-port-nwk-ctp-conn-ptr 2  
interwork-option mac-bridge-lan service-profile-ptr 1 interwork-tp-ptr 0 gal-profile-ptr 0
```

**# Creación de Multicast GEM Interworking Termination Point (para el servicio multicast).**

```
multicast-gem-interworking-termination-point create instance 3 gem-port-nwk-ctp-conn-ptr 3  
interwork-option mac-bridge service-prof-ptr 65535 interwork-tp-ptr 0 gal-prof-ptr 65535  
gal-lpbk-config 0
```

**# Creación del VLAN Tagging Filter Data con el identificador VLAN 833.**

```
vlan-tagging-filter-data create instance 2 forward-operation h-vid-a vlan-tag1 833 vlan-priority1  
7 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
```

**# Creación de Extended VLAN Tagging Operation Config Data, que será configurado en el paso posterior. Sirve para gestionar los identificadores VLAN.**

```
extended-vlan-tagging-operation-config-data create instance 257 association-type ptp-eth-uni  
associated-me-ptr 257
```

**# Configuración de Extended VLAN Tagging Operation Config Data. Se debe configurar para cada identificador VLAN.**

```
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 833 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 833 treatment-inner-tpid tpid-de-  
copy-from-inner
```

**# Salida al menú del canal 0 de la OLT 0**

```
exit
```

**# Configuración de las reglas VLAN en la OLT asociadas al puerto 800**

```
vlan uplink configuration port-id 800 min-cos 0 max-cos 7 de-bit disable primary-tag-handling
false
vlan uplink handling port-id 800 primary-vlan none destination datapath c-vlan-handling no-
change s-vlan-handling no-change new-c-vlan 0 new-s-vlan 0
```

**# Creación del perfil de ancho de banda en el sentido de bajada.**

```
policing downstream profile committed-max-bw 256000 committed-burst-size 1023 excess-max-
bw 51200 excess-burst-size 1023
```

**# Asignación del perfil de ancho de banda recién creado al Port-ID 800 (el ds-profile-index se mostrará en el paso anterior).**

```
policing downstream port-configuration entity port-id 800 ds-profile-index 0
exit
```

**# Acceso al menú de la red PON.**

```
pon
```

**# Selección del menú del algoritmo DBA de la PON 0.**

```
dba pythagoras 0
```

**# Asignación del ancho de banda en el sentido de subida para el Alloc-ID 810.**

```
sla 810 service data status-report nsr gr-bw 50 gr-fine 0 be-bw 100 be-fine 0
end
```

## Apéndice B: Scripts de Python

En este apéndice, se recogen los scripts de Python explicados en el Capítulo 5. El código está comentado (las líneas comentadas comienzan por el carácter #) de forma oportuna en los diferentes scripts. Pulsando la tecla Ctrl + botón izquierdo en el nombre del script, se mostrará el código del script correspondiente: *configuracion\_red\_GPON.py*, *acceso\_CLI.py*, *activar\_CLI.py*, *activar\_TGMS.py*, *mostrar\_modos\_funcionamiento.py* y *mostrar\_ONUs\_conectadas.py*.

### Script *configuracion\_red\_GPON.py*

```
# Lista de módulos importados para el correcto funcionamiento del programa
# Módulo telnetlib: permite establecer la conexión Telnet e introducir los comandos
import telnetlib

# Módulo time: permite establecer un tiempo entre la escritura de comandos
# para que no se sobrecargue el CLI. Cada escritura de un comando irá seguida
# de un time.sleep().
import time

# Módulo math: permite realizar operaciones matemáticas con los anchos de banda
import math

# Módulos os y path: permite comprobar la existencia de ficheros en el directorio de trabajo
import os.path as path
import os

# Módulo necesario para la creación y modificación del fichero XML. Este fichero permitirá
# recuperar las configuraciones de las ONUs al apagar la red o al cambiar de modo de
# gestión (hay que recordar que el CLI NO TIENE PERSISTENCIA en los datos configurados).
import xml.etree.cElementTree as etree

# !!!! IMPORTANTE !!!!!!!
# El campo que permite diferenciar a las ONUs es el número de serie, que está
# formado por los campos Vendor ID y Vendor Specific (este a su vez está formado
# por los últimos 8 valores hexadecimales de la dirección MAC). Por tanto, lo
```

```
# que diferencia a los números de serie de las ONUs de la red es la dirección MAC.
# Por simplicidad, en los comentarios de todo el programa se utilizará el término
# dirección MAC para referirse al número de serie.

# Función get_ID_ONU(): permite detectar las ONUs conectadas a la red GPON a través
# de su dirección MAC. Asimismo, asocia a cada dirección MAC el identificador de ONU,
# parámetro muy importante en el resto del programa y que varía cada vez que se enciende
# la red o se activa el modo de gestión CLI. También se crea en esta función, en caso de
# no existir, el fichero XML que permite cargar configuraciones anteriores
def get_ID_ONU():

    # 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)

# Declaración y escritura del comando que permite ver las ONUs conectadas a
# la red así como sus direcciones MAC
comandos = "configure \n olt-device 0 \n olt-channel 0 \n show serial-number allocated \n"
tn.write(comandos.encode('ascii') + b"\n")
time.sleep(0.1)

# Lectura de los datos (tanto enviados como recibidos) del CLI y volcado en
# un fichero de texto para su posterior análisis
data = tn.read_very_eager().decode()

# Se abre el fichero con modo de escritura, de esta forma cada vez que cambie
# el estado de la red el fichero se sobrescribirá con la información nueva
outfile = open('IDs_ONUs.txt', 'w')

# Se escriben los datos procedentes de la escritura de los comandos de arriba
outfile.write(data)

# Ciere del fichero
outfile.close()

# Creación del vector que almacenará las direcciones MAC de las ONUs conectadas
MAC = []

# Se abre el archivo anterior en modo lectura
outfile = open('IDs_ONUs.txt', 'r')

# Se almacenan todas las líneas del fichero con la función readlines()
lines = outfile.readlines()

# Bucle que recorre cada línea del fichero
for line in lines:

    # Se almacenan todas las palabras de cada línea
    palabras = line.split()

    # Bucle que recorre cada palabra de la línea
    for p in palabras:

```

```
# Si los 18 primeros caracteres de una palabra coinciden con los indicados,
# se trata de una direccion MAC -> Se ha detectado una ONU
if p[:18]=='54-4c-52-49-5b-01-':
    # Se añade la MAC al vector. La posición en la que se añade indica
    # el identificador de la ONU.
    MAC.append(p)

# FORMA DE BUSCAR ONUs ONU A ONU -> Menos eficiente y hay que añadir
# código
# si se conecta alguna ONU más a la red
# if p=='54-4c-52-49-5b-01-f6-90':
#     MAC[id_ONU] = '54-4c-52-49-5b-01-f6-90'
#     id_ONU = id_ONU + 1
# if p=='54-4c-52-49-5b-01-f7-30':
#     MAC[id_ONU] = '54-4c-52-49-5b-01-f7-30'
#     id_ONU = id_ONU + 1
# if p=='54-4c-52-49-5b-01-f6-d8':
#     MAC[id_ONU] = '54-4c-52-49-5b-01-f6-d8'
#     id_ONU = id_ONU + 1
# if p=='54-4c-52-49-5b-01-f7-28':
#     MAC[id_ONU] = '54-4c-52-49-5b-01-f7-28'
#     id_ONU = id_ONU + 1

# Cierre del archivo
outfile.close()

# Si existe un fichero con este nombre en el directorio de trabajo, sale de la función
# devolviendo el vector de direcciones MAC
if path.exists("configuracionGPON.xml"):
    return MAC
```

```

# Si no existe, se ha de crear el fichero XML

else:

    # Se define el elemento raíz

    root = etree.Element("RedGPON")

    # Se añaden las ONUs (el nº de ONUs es la longitud del vector de direcciones MAC)

    i=0

    while i<len(MAC):

        # Cada ONU se añade con su dirección MAC como un subelemento del elemento raíz

        ONU = etree.SubElement(root,"ONU", MAC=MAC[i])

        i=i+1

# FORMA DE CREAR EL ÁRBOL ONU A ONU -> Menos eficiente y hay que añadir
código

# si se conecta alguna ONU más a la red

# ONU1 = etree.SubElement(root,"ONU", MAC='54-4c-52-49-5b-01-f6-90')
# ONU2 = etree.SubElement(root,"ONU", MAC='54-4c-52-49-5b-01-f7-30')
# ONU3 = etree.SubElement(root,"ONU", MAC='54-4c-52-49-5b-01-f6-d8')
# ONU4 = etree.SubElement(root,"ONU", MAC='54-4c-52-49-5b-01-f7-28')

# Se crea el árbol XML y se escribe en el fichero con el nombre indicado

tree = etree.ElementTree(root)

tree.write("configuracionGPON.xml")

# Se devuelve el vector de direcciones MAC

return MAC

# Función servicio_Internet(ID_ONU,MAC_ONU,num_servicios_Internet): permite configurar
# servicios de Internet y toma como parámetros el identificador de la ONU
# (posición en el vector de direcciones MAC), la MAC y el número de servicios a configurar.
# Guarda la configuración actual en un fichero txt (esta configuración es temporal y se

```

```
# mantiene mientras no se apague la red ni se cambie al modo de gestión TGMS).
# La función también actualiza el fichero XML para recuperar configuraciones
# cuando se apaga la red o se cambia al TGMS.
def servicio_Internet(ID_ONU,MAC_ONU,num_servicios_Internet):

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

    # Dependiendo del número de servicios, se asignan tantos valores como sean necesarios
    # a los vectores anteriormente creados
    i=0
    while i < num_servicios_Internet:
        # 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+2)
        tcont_ID.append(i)
        puntero.append(32768+i)
        ds_profile_index.append(i)
        i=i+1

    # Flags que marcarán la salida de los diferentes bucles cuando el parámetro
```

```

# requerido haya sido introducido de forma válida

true_VLAN = 0

true_BW_Downstream_GR = 0

true_BW_Downstream_Excess = 0

true_BW_Upstream_GR = 0

true_BW_Upstream_BE = 0

# Creación de los vectores en los que se almacenarán los parámetros
# de configuración que el USUARIO deberá introducir: identificador de VLAN,
# ancho de banda Downstream garantizado y en exceso y ancho de banda Upstream
# garantizado y Best Effort

VLAN_ID = []

BW_Downstream_GR = []

BW_Downstream_Excess = []

BW_Upstream_GR = []

BW_Upstream_BE = []

# En los sucesivos bucles, se irán pidiendo los parámetros de configuración al usuario.

# Primer bucle: se piden el identificador VLAN que corresponda a cada servicio
# La VLAN 833 conecta con un servidor DHCP que asigna la dirección mientras que
# la VLAN 806 recibe una IP de forma estática (no la tiene que introducir el usuario)
i=0

while i < num_servicios_Internet:

    num_servicio = str(i+1)

    true_VLAN = 0

    cont=0

    # Las etiquetas VLAN van de 1 a 4094. Si el usuario introduce un valor fuera de ese
    # rango, se le volverá a pedir que introduzca el valor. Hay que recordar que la red
    # solo ofrece servicio en las VLAN 833 y 806

    while true_VLAN == 0:

```

```
if cont == 0:

    print("Identificador VLAN [1-4094] para el servicio " + num_servicio + " (833 -
Servidor DHCP, 806 - Dirección IP estática): ")

    else:

        print("El identificador VLAN no es válido. Vuelva a probar:")

    cont = cont+1

    entrada = input()

    entrada = int(entrada)

    if entrada > 0 and entrada < 4095:

        VLAN_ID.append(entrada)

        print("El identificador VLAN para el servicio " + num_servicio + " es:", VLAN_ID[i])

        i=i+1

        true_VLAN = 1

print("\n")

# Segundo bucle: se pide el ancho de banda garantizado en sentido Downstream.
# Este ancho de banda se introduce en Kbps y debe estar entre 0 y 2488000 (aunque
# a efectos prácticos no tiene sentido meter más de 600000 Kbps, aproximadamente).
# El CLI solo admite múltiplos de 64 Kbps de manera que el programa trunca el
# valor del usuario para que coincida con un múltiplo de 64 Kbps.

i=0

while i < num_servicios_Internet:

    num_servicio = str(i+1)

    true_BW_Downstream_GR = 0

    cont=0

    while true_BW_Downstream_GR == 0:

        if cont == 0:

            print("Ancho de banda Downstream garantizado en Kbps [0-2488000] para el servicio
" + num_servicio + ": ")
```

```

else:
    print("Ancho de banda no válido. Vuelva a probar:")

cont=cont+1

entrada = input()

entrada = int(entrada)

entrada = entrada / 64

entrada = math.floor(entrada)

entrada = entrada * 64

if entrada > -1 and entrada < 2488000:

    BW_Downstream_GR.append(entrada)

    print("El ancho de banda Downstream garantizado en Kbps es:",
BW_Downstream_GR[i])

    i=i+1

    true_BW_Downstream_GR = 1

# Tercer bucle: se pide el ancho de banda en exceso en sentido Downstream.
# Este ancho de banda se introduce en Kbps y debe estar entre 0 y 2488000 (aunque
# a efectos prácticos no tiene sentido que la suma del garantizado y exceso sea
# mayor que 600000 Kbps, aproximadamente). El CLI solo admite múltiplos de 64 Kbps
# de manera que el programa trunca el valor del usuario para que coincida con
# un múltiplo de 64 Kbps.

i=0

while i < num_servicios_Internet:

    num_servicio = str(i+1)

    true_BW_Downstream_Excess = 0

    cont=0

    while true_BW_Downstream_Excess == 0:

        if cont == 0:

            print("Ancho de banda Downstream en exceso en Kbps [0-2488000] para el servicio "
+ num_servicio + ": ")

        else:

            print("Ancho de banda no válido. Vuelva a probar:")

```

```
cont=cont+1

entrada = input()
entrada = int(entrada)
entrada = entrada / 64
entrada = math.floor(entrada)
entrada = entrada * 64

if entrada > -1 and entrada < 2488000:

    BW_Downstream_Excess.append(entrada)

    print("El ancho de banda Downstream en exceso en Kbps es:",
BW_Downstream_Excess[i])

    i=i+1

    true_BW_Downstream_Excess = 1

print("\n")

# Cuarto bucle: se pide el ancho de banda garantizado en sentido Upstream.
# Este ancho de banda se introduce en Mbps y debe estar entre 0 y 1244 (aunque
# a efectos prácticos no tiene sentido meter más de 600 Mbps, aproximadamente).
i=0

while i < num_servicios_Internet:

    num_servicio = str(i+1)

    true_BW_Upstream_GR = 0

    cont=0

    while true_BW_Upstream_GR == 0:

        if cont == 0:

            print("Ancho de banda Upstream garantizado en Mbps [0-1244] para el servicio " +
num_servicio + ": ")

        else:

            print("Ancho de banda no válido. Vuelva a probar:")

        cont=cont+1
```

```

    entrada = input()

    entrada = int(entrada)

    if entrada > -1 and entrada < 1245:

        BW_Upstream_GR.append(entrada)

        print("El ancho de banda Upstream garantizado en Mbps es:", BW_Upstream_GR[i])

        i=i+1

        true_BW_Upstream_GR = 1

print("\n")

# Quinto bucle: se pide el ancho de banda garantizado Best Effort Upstream.
# Este ancho de banda se introduce en Mbps, debe estar entre 0 y 1244 (aunque
# a efectos prácticos no tiene sentido meter más de 600 Mbps, aproximadamente) y
# tiene que ser igualo mayor que el ancho de banda Upstream garantizado (el valor Best
# Effort es el mayor ancho de banda que podrá recibir la ONU)

i=0

while i < num_servicios_Internet:

    num_servicio = str(i+1)

    true_BW_Upstream_BE = 0

    cont=0

    while true_BW_Upstream_BE == 0:

        if cont == 0:

            print("Ancho de banda Upstream BE en Mbps [0-1244] para el servicio " +
num_servicio + ": ")

        else:

            print("Ancho de banda no válido. Vuelva a probar:")

        cont=cont+1

        entrada = input()

        entrada = int(entrada)

        if entrada > -1 and entrada < 1245 and entrada >= BW_Upstream_GR[i]:

            BW_Upstream_BE.append(entrada)

```

```
print("El ancho de banda Upstream BE en Mbps es:", BW_Upstream_BE[i])

i=i+1

true_BW_Upstream_BE = 1

print("\n")

# 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)

# A la hora de introducir variables en la declaración de cadenas, aunque sean números,
# se deben introducir en forma de string. Por ello, aparece a lo largo del programa muchas
# ocasiones la función str(), que convierte una variable en string
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)

i=0

# Se crean, dentro del menú de la ONU a configurar, tantos Alloc-ID como servicios

while i < num_servicios_Internet:

    allocID = "alloc-id " + str(alloc_ID[i]).strip('[]') + " \n"

    tn.write(allocID.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1

# El comando exit hace salir hacia el menú anterior del CLI en la estructura de menús

salir = "exit \n"

tn.write(salir.encode('ascii') + b"\n")

time.sleep(0.2)

# Cada Alloc-ID irá asociado a un puerto (por convención, se utiliza el mismo identificador)

# Estos Alloc-IDs y puertos no se pueden utilizar para otras ONUs ni para otros servicios

i=0

while i < num_servicios_Internet:

    portalloc = "port " + str(port_ID[i]).strip('[]') + " alloc-id " + str(alloc_ID[i]).strip('[]') + "
\n"

    tn.write(portalloc.encode('ascii') + b"\n")

    time.sleep(0.2)

```

```
i=i+1

# Desde el menú de configuración del canal OMCI de comunicación, se crean las entidades
# MIB que forman el servicio de Internet. Estas entidades vienen en el estándar GPON.

omci = "onu-omci " + ID_ONU + " \n"
tn.write(omci.encode('ascii') + b"\n")
time.sleep(0.2)

# Creación de las entidades T-Cont (colas). Cada servicio está asociado con un T-Cont
# y está vinculado a un Alloc-ID

i=0
while i < num_servicios_Internet:
    tcont = "t-cont set slot-id 128 t-cont-id " + str(tcont_ID[i]).strip('[]') + " alloc-id " +
str(alloc_ID[i]).strip('[]') + " \n"
    tn.write(tcont.encode('ascii') + b"\n")
    time.sleep(0.2)
    i=i+1

# Creación de MAC Bridge Service Profile. Se asociará esta entidad con los MAC Bridge
# Port Configuration Data a través del bridge-group-id en los siguientes comandos.

macservice = "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 \n"

# Creación del primer MAC Bridge Port Configuration Data. Irá asociado a la entidad
# extended-vlan-tagging-operation-config-data. Para ello, el tp-type (tipo del punto
# de terminación del MAC Bridge) ha de ser lan y el puntero tp-ptr debe tener el mismo
# valor que el nº de instancia de la entidad extended-vlan-tagging-operation-config-data.

macbridge1 = "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 \n"

tn.write(macservice.encode('ascii') + b"\n")
```

```

time.sleep(0.2)

tn.write(macbridge1.encode('ascii') + b"\n")

time.sleep(0.2)

# Creación de los restantes Mac Bridge Port Configuration Data. Irán asociado a la entidades
# VLAN-tagging-filter-data mediante los números de instancia. También irán asociados a los
# GEM Interworking Termination Point mediante el tp-type (tipo gem) y el tp-ptr, que
# tiene que coincidir con el número de instancia del GEM Interworking Termination Point
i=0

while i < num_servicios_Internet:

    macbridge2 = "mac-bridge-pcd create instance " + str(num_instancia[i]).strip('[]') + "
bridge-id-ptr 1 port-num " + str(num_instancia[i]).strip('[]') + " tp-type gem tp-ptr " +
str(num_instancia[i]).strip('[]') + " port-priority 0 port-path-cost 1 port-spanning-tree-ind true
encap-method llc lanfcs-ind forward \n"

    tn.write(macbridge2.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1

# Creación de los GEM Port Network CTP, que irán asociado a los puertos especificados
# anteriormente. Los identificadores tienen que coincidir con el gem-port-nwk-ct-conn-ptr
# de los GEM Interworking Termination Point.
i=0

while i < num_servicios_Internet:

    gemport = "gem-port-network-ctp create instance " + str(num_instancia[i]).strip('[]') + "
port-id " + str(port_ID[i]).strip('[]') + " t-cont-ptr " + str(puntero[i]).strip('[]') + " direction
bidirectional traffic-mgnt-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.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1

# Creación de los GEM Interworking Termination Point (en este punto, se produce la
# transformación de flujo de bytes a tramas GEM y viceversa). Estas entidades se vinculan

```

```
# a los GEM Port Network CTP a través del campo gem-port-nwk-ctp-conn-ptr, que debe coincidir

# con el número de instancia que utilizado en el GEM Port Network CTP. Estas entidades también

# se asocian con los MAC Bridge Port Configuration. Para ello, hay que seleccionar como

# interwork-option mac-bridge-lan y el campo service-profile-ptr debe ser un 1.

# El número de instancia debe ser el mismo que el tp-ptr del MAC Bridge Point Configuration Data.

i=0

while i < num_servicios_Internet:

    geminterworking = "gem-interworking-termination-point create instance " +
str(num_instancia[i]).strip('[]') + " gem-port-nwk-ctp-conn-ptr " + str(num_instancia[i]).strip('[]')
+ " interwork-option mac-bridge-lan service-profile-ptr 1 interwork-tp-ptr 0 gal-profile-ptr 0 \n"

    tn.write(geminterworking.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1

# Creación de los VLAN Tagging Filter Data con los identificadores VLAN definidos arriba.

# Los números de instancia deben coincidir con los de los MAC Bridge Port Configuration Data asociados.

i=0

while i < num_servicios_Internet:

    vlantagging = "vlan-tagging-filter-data create instance " + str(num_instancia[i]).strip('[]') + "
forward-operation h-vid-a vlan-tag1 " + str(VLAN_ID[i]).strip('[]') + " vlan-priority1 7 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 \n"

    tn.write(vlantagging.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1

# Creación de la entidad Extended VLAN Tagging Operation Config Data, que será configurada en el paso posterior.
```

```

# Sirve para gestionar los identificadores VLAN. Esta entidad está asociada al primer MAC
Bridge Port Configuration Data

# a través del número de instancia, que coincide con el tp-ptr del MAC Bridge Port
Configuration Data.

extendedvlan = "extended-vlan-tagging-operation-config-data create instance 257 association-
type ptp-eth-uni associated-me-ptr 257 \n"

tn.write(extendedvlan.encode('ascii') + b"\n")

time.sleep(0.2)

# Configuración de la entidadExtended VLAN Tagging Operation Config Data. Se debe
configurar

# para cada identificador VLAN.

i=0

while i < num_servicios_Internet:

    extendedvlanconf = "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 " + str(VLAN_ID[i]).strip('[]') + " 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 " + str(VLAN_ID[i]).strip('[]') + " treatment-inner-tpid tpid-de-copy-from-
inner\n"

    tn.write(extendedvlanconf.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1

# El comando exit hace salir hacia el menú anterior del CLI en la estructura de menús

salir = "exit \n"

tn.write(salir.encode('ascii') + b"\n")

time.sleep(0.2)

# Configuración de las reglas VLAN en la OLT asociadas a los puertos definidos al principio
de la función

i=0

while i < num_servicios_Internet:

    reglasvlan = "vlan uplink configuration port-id " + str(port_ID[i]).strip('[]') + " min-cos 0
max-cos 7 de-bit disable primary-tag-handling false \n vlan uplink handling port-id " +

```

```
str(port_ID[i]).strip('[]') + " primary-vlan none destination datapath c-vlan-handling no-change s-  
vlan-handling no-change new-c-vlan 0 new-s-vlan 0 \n"
```

```
tn.write(reglasvlan.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

```
i=i+1
```

```
# Creación de los perfiles de ancho de banda en sentido Downstream con los parámetros
```

```
# definidos por el usuario anteriormente.
```

```
i=0
```

```
while i < num_servicios_Internet:
```

```
    perfiledownstreamconf = "policing downstream profile committed-max-bw " +  
str(BW_Downstream_GR[i]).strip('[]') + " committed-burst-size 1023 excess-max-bw " +  
str(BW_Downstream_Excess[i]).strip('[]') + " excess-burst-size 1023 \n"
```

```
    tn.write(perfiledownstreamconf.encode('ascii') + b"\n")
```

```
    time.sleep(0.2)
```

```
    i=i+1
```

```
# Nombre del fichero en el que se guardará la configuración del servicio de Internet
```

```
nombre_archivo = 'Servicio_Internet_ONU_MAC_' + MAC_ONU + '.txt'
```

```
# Al crear los perfiles de ancho de banda en sentido de bajada, el CLI devuelve un
```

```
# índice de perfil. Para asociar los perfiles creados, se deben utilizar esos índices
```

```
# de perfil y asociarlos a los puertos definidos anteriormente. Para ello, se vuelcan los datos
```

```
# enviados y recibidos del CLI en un fichero, del que se extraerán esos índices.
```

```
datos_perfil = tn.read_very_eager().decode()
```

```
outfile = open(nombre_archivo, 'a')
```

```
outfile.write(datos_perfil)
```

```
outfile.close()
```

```
outfile = open(nombre_archivo, 'r')
```

```
lines = outfile.readlines()
```

```
true_ds_profile = 0
```

```

# Se busca el índice de perfil que primero aparezca en el fichero.

i=0

for i in range (0,500):
    if true_ds_profile == 1:
        break
    j = str(i)
    cadena = 'downstream_profile_index: ' + j + ' '
    for line in lines:
        if cadena in line:
            j = int(j)
            ds_profile_index[0] = j
            true_ds_profile = 1

outfile.close()

# El índice encontrado corresponde al primer servicio. En caso de haber más, el índice
# de perfil será ese aumentado en una unidad.

i=0

for i in range (0,num_servicios_Internet):
    ds_profile_index[i] = ds_profile_index[0] + i

# Asignación de los perfiles de ancho de banda Downstream a los puertos correspondientes
# mediante los índices de perfil buscados anteriormente.

i=0

while i < num_servicios_Internet:
    perfiledownstreamassign = "policing downstream port-configuration entity port-id " +
str(port_ID[i]).strip('[]') + " ds-profile-index " + str(ds_profile_index[i]).strip('[]') + " \n"
    tn.write(perfiledownstreamassign.encode('ascii') + b"\n")
    time.sleep(0.2)
    i=i+1

```

```
# El comando exit hace salir hacia el menú anterior del CLI en la estructura de menús
salir = "exit \n"
tn.write(salir.encode('ascii') + b"\n")
time.sleep(0.2)

# Desde el menú de configuración del algoritmo DBA, se definirían los perfiles de
# ancho de banda en sentido Upstream
dba = "pon \n dba pythagoras 0 \n "
tn.write(dba.encode('ascii') + b"\n")
time.sleep(0.2)

# Cada perfil upstream irá asociado a un Alloc-ID y estará configurado con los
# parámetros que haya definido el usuario.
i=0
while i < num_servicios_Internet:
    perfilupstream = "sla " + str(alloc_ID[i]).strip('[]') + " service data status-report nsr gr-bw "
+ str(BW_Upstream_GR[i]).strip('[]') + " gr-fine 0 be-bw " + str(BW_Upstream_BE[i]).strip('[]')
+ " be-fine 0 \n"
    tn.write(perfilupstream.encode('ascii') + b"\n")
    time.sleep(0.2)
    i=i+1

# El comando end hace salir directamente al modo privilegiado en la estructura
# de menús del CLI
final = "end \n"
tn.write(final.encode('ascii') + b"\n")
time.sleep(0.2)

# Se vuelcan todos los datos en el fichero definido anteriormente (la opción 'a'
# hace que los datos se añadan al final del fichero) de forma que el fichero recogerá
```

```
# toda la configuración del servicio de Internet

data = tn.read_very_eager().decode()

outfile = open(nombre_archivo, 'a')

outfile.write(data)

outfile.write("\n\n\n")

outfile.close()

# Una vez configurado el servicio, se muestra un mensaje al usuario

print("Servicio de Internet configurado. \n")

# Actualización del fichero XML con los datos configurados

doc = etree.parse("configuracionGPON.xml")

# Extracción del elemento raíz

redGPON = doc.getroot()

# Obtención del índice correspondiente a la ONU que se está configurando.

# Para ello, se hace una comparación de las diferentes direcciones MAC con

# la de la ONU a configurar.

index=0

true_index=0

while true_index==0:

    for attr,value in redGPON[index].items():

        if value == MAC_ONU:

            true_index=1

            break

        index=index+1

# Se borra cualquier configuración anterior que en este caso, al configurar un

# nuevo servicio, se va a desechar

num_servicios = len(redGPON[index])

i=0
```

```
while i<num_servicios:

    redGPON[index].remove(redGPON[index][0])

    i=i+1

# Se actualiza la parte del árbol correspondiente a la ONU en cuestión con los
# parámetros que ha definido el usuario. Se utiliza la función SubElement para
# crear el servicio así como los parámetros del mismo.

i=0

while i < num_servicios_Internet:

    Servicio = etree.SubElement(redGPON[index], "Servicio", tipo='Internet')

    VLAN = etree.SubElement(redGPON[index][i], "VLAN_ID").text =
str(VLAN_ID[i]).strip('[]')

    BW_Down_GR = etree.SubElement(redGPON[index][i], "BW_Down_GR").text =
str(BW_Downstream_GR[i]).strip('[]')

    BW_Down_Excess = etree.SubElement(redGPON[index][i], "BW_Down_Excess").text =
str(BW_Downstream_Excess[i]).strip('[]')

    BW_Up_GR = etree.SubElement(redGPON[index][i], "BW_Up_GR").text =
str(BW_Upstream_GR[i]).strip('[]')

    BW_Up_BE = etree.SubElement(redGPON[index][i], "BW_Up_BE").text =
str(BW_Upstream_BE[i]).strip('[]')

    i=i+1

# Finalmente se crea el nuevo árbol actualizado y se escribe en el fichero

doc = etree.ElementTree(redGPON)

doc.write("configuracionGPON.xml")

return

# Función servicio_Video(ID_ONU,MAC_ONU,num_servicios_Video): permite configurar
# servicios de Internet + Vídeo (el servicio multicast del vídeo debe ir sobre uno
# Ethernet para gestionar el tráfico IMGP asociado) y toma como parámetros el identificador de
la ONU
```

```

# (posición en el vector de direcciones MAC), la MAC y el número de servicios a configurar.
# Guarda la configuración actual en un fichero txt (esta configuración es temporal y se
# mantiene mientras no se apague la red ni se cambie al modo de gestión TGMS).
# La función también actualiza el fichero XML para recuperar configuraciones
# cuando se apaga la red o se cambia al TGMS.
def servicio_Video(ID_ONU,MAC_ONU,num_servicios_Video):

    # Creación de los vectores en los que se almacenarán los parámetros
    # internos de configuración

    port_ID = []
    alloc_ID = []
    tcont_ID = []
    num_instancia = []
    puntero = []
    ds_profile_index = []

    # Dependiendo del número de servicios, se asignan tantos valores como sean necesarios
    # a los vectores anteriormente creados

    i=0
    while i < num_servicios_Video:

        # 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)

        ds_profile_index.append(i)

        i=i+1

```

```
# Flags que marcarán la salida de los diferentes bucles cuando el parámetro
# requerido haya sido introducido de forma válida
true_VLAN = 0
true_BW_Downstream_GR = 0
true_BW_Downstream_Excess = 0
true_BW_Upstream_GR = 0
true_BW_Upstream_BE = 0

# Creación de los vectores en los que se almacenarán los parámetros
# de configuración que el USUARIO deberá introducir: identificador de VLAN,
# ancho de banda Downstream garantizado y en exceso y ancho de banda Upstream
# garantizado y Best Effort
VLAN_ID = []
BW_Downstream_GR = []
BW_Downstream_Excess = []
BW_Upstream_GR = []
BW_Upstream_BE = []

# En los sucesivos bucles, se irán pidiendo los parámetros de configuración al usuario.
# IMPORTANTE: EN EL CLI, NO HAY QUE DEFINIR ANCHOS DE BANDA PARA EL
SERVICIO MULTICAST.
# EL ANCHO DE BANDA UTILIZADO POR EL VÍDEO ES EL DEFINIDO EN EL
SERVICIO ETHERNET.

# Primer bucle: se piden el identificador VLAN que corresponda a cada servicio
# La VLAN 833 conecta con un servidor DHCP que asigna la dirección mientras que
# la VLAN 806 recibe una IP de forma estática (no la tiene que introducir el usuario)
i=0
while i < num_servicios_Video:
```

```

num_servicio = str(i+1)

true_VLAN = 0

cont=0

# Las etiquetas VLAN van de 1 a 4094. Si el usuario introduce un valor fuera de ese
# rango, se le volverá a pedir que introduzca el valor. Hay que recordar que la red
# solo ofrece servicio en las VLAN 833 y 806

while true_VLAN == 0:

    if cont == 0:

        print("Identificador VLAN [1-4094] para el servicio " + num_servicio + " (833 -
Servidor DHCP, 806 - Dirección IP estática): ")

    else:

        print("El identificador VLAN no es válido. Vuelva a probar:")

    cont = cont+1

    entrada = input()

    entrada = int(entrada)

    if entrada > 0 and entrada < 4095:

        VLAN_ID.append(entrada)

        print("El identificador VLAN para el servicio " + num_servicio + " es:", VLAN_ID[i])

        i=i+1

        true_VLAN = 1

print("\n")

# Segundo bucle: se pide el ancho de banda garantizado en sentido Downstream.
# Este ancho de banda se introduce en Kbps y debe estar entre 0 y 2488000 (aunque
# a efectos prácticos no tiene sentido meter más de 600000 Kbps, aproximadamente).
# El CLI solo admite múltiplos de 64 Kbps de manera que el programa trunca el
# valor del usuario para que coincida con un múltiplo de 64 Kbps.

i=0

while i < num_servicios_Video:

    num_servicio = str(i+1)

```

```
true_BW_Downstream_GR = 0

cont=0

while true_BW_Downstream_GR == 0:

    if cont == 0:

        print("Ancho de banda Downstream garantizado en Kbps [0-2488000] para el servicio
" + num_servicio + ": ")

    else:

        print("Ancho de banda no válido. Vuelva a probar:")

    cont=cont+1

    entrada = input()

    entrada = int(entrada)

    entrada = entrada / 64

    entrada = math.floor(entrada)

    entrada = entrada * 64

    if entrada > -1 and entrada < 2488000:

        BW_Downstream_GR.append(entrada)

        print("El ancho de banda Downstream garantizado en Kbps es:",
BW_Downstream_GR[i])

        i=i+1

        true_BW_Downstream_GR = 1

# Tercer bucle: se pide el ancho de banda en exceso en sentido Downstream.
# Este ancho de banda se introduce en Kbps y debe estar entre 0 y 2488000 (aunque
# a efectos prácticos no tiene sentido que la suma del garantizado y exceso sea
# mayor que 600000 Kbps, aproximadamente). El CLI solo admite múltiplos de 64 Kbps
# de manera que el programa trunca el valor del usuario para que coincida con
# un múltiplo de 64 Kbps.

i=0

while i < num_servicios_Video:

    num_servicio = str(i+1)
```

```

true_BW_Downstream_Excess = 0

cont=0

while true_BW_Downstream_Excess == 0:

    if cont == 0:

        print("Ancho de banda Downstream en exceso en Kbps [0-2488000] para el servicio "
+ num_servicio + ": ")

    else:

        print("Ancho de banda no válido. Vuelva a probar:")

    cont=cont+1

    entrada = input()

    entrada = int(entrada)

    entrada = entrada / 64

    entrada = math.floor(entrada)

    entrada = entrada * 64

    if entrada > -1 and entrada < 2488000:

        BW_Downstream_Excess.append(entrada)

        print("El ancho de banda Downstream en exceso en Kbps es:",
BW_Downstream_Excess[i])

        i=i+1

        true_BW_Downstream_Excess = 1

print("\n")

# Cuarto bucle: se pide el ancho de banda garantizado en sentido Upstream.
# Este ancho de banda se introduce en Mbps y debe estar entre 0 y 1244 (aunque
# a efectos prácticos no tiene sentido meter más de 600 Mbps, aproximadamente).
i=0
while i < num_servicios_Video:

    num_servicio = str(i+1)

    true_BW_Upstream_GR = 0

    cont=0

```

```
while true_BW_Upstream_GR == 0:
    if cont == 0:
        print("Ancho de banda Upstream garantizado en Mbps [0-1244] para el servicio " +
num_servicio + ": ")
    else:
        print("Ancho de banda no válido. Vuelva a probar:")
    cont=cont+1
    entrada = input()
    entrada = int(entrada)
    if entrada > -1 and entrada < 1245:
        BW_Upstream_GR.append(entrada)
        print("El ancho de banda Upstream garantizado en Mbps es:", BW_Upstream_GR[i])
        i=i+1
        true_BW_Upstream_GR = 1

print("\n")

# Quinto bucle: se pide el ancho de banda garantizado Best Effort Upstream.
# Este ancho de banda se introduce en Mbps, debe estar entre 0 y 1244 (aunque
# a efectos prácticos no tiene sentido meter más de 600 Mbps, aproximadamente) y
# tiene que ser igualo mayor que el ancho de banda Upstream garantizado (el valor Best
# Effort es el mayor ancho de banda que podrá recibir la ONU)
i=0
while i < num_servicios_Video:
    num_servicio = str(i+1)
    true_BW_Upstream_BE = 0
    cont=0
    while true_BW_Upstream_BE == 0:
        if cont == 0:
```

```

        print("Ancho de banda Upstream BE en Mbps [0-1244] para el servicio " +
num_servicio + ": ")

    else:

        print("Ancho de banda no válido. Vuelva a probar:")

    cont=cont+1

    entrada = input()

    entrada = int(entrada)

    if entrada > -1 and entrada < 1245 and entrada >= BW_Upstream_GR[i]:

        BW_Upstream_BE.append(entrada)

        print("El ancho de banda Upstream BE en Mbps es:", BW_Upstream_BE[i])

        i=i+1

        true_BW_Upstream_BE = 1

print("\n")

# 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)

# A la hora de introducir variables en la declaración de cadenas, aunque sean números,
# se deben introducir en forma de string. Por ello, aparece a lo largo del programa muchas
# ocasiones la función str(), que convierte una variable en string
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 Ethernet
# ¡¡¡IMPORTANTE!!! El servicio multicast ocupa solo el canal descendente y no necesita la
# asignación de un Alloc-ID para el acceso al canal ascendente de transmisión ni parámetros
de SLA.

i=0

while i < num_servicios_Video:

    allocID = "alloc-id " + str(alloc_ID[i]).strip('[]') + " \n"

    tn.write(allocID.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1
```

```

# El comando exit hace salir hacia el menú anterior del CLI en la estructura de menús.

salir = "exit \n"

tn.write(salir.encode('ascii') + b"\n")

time.sleep(0.2)

# Cada Alloc-ID irá asociado a un puerto (por convención, se utiliza el mismo identificador).
# Estos Alloc-IDs y puertos no se pueden utilizar para otras ONUs ni para otros servicios.
# El servicio multicast irá sobre el puerto 4094 (podría ser otro) pero como se explicó arriba,
# no necesita la asignación de un Alloc-ID

i=0

while i < num_servicios_Video:

    portalloc = "port " + str(port_ID[i]).strip('[]') + " alloc-id " + str(alloc_ID[i]).strip('[]') + "
\n"

    tn.write(portalloc.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1

# Desde el menú de configuración del canal OMCI de comunicación, se crean las entidades
# MIB que forman el servicio de Internet y Vídeo. Estas entidades vienen en el estándar
GPON.

omci = "onu-omci " + ID_ONU + " \n"

tn.write(omci.encode('ascii') + b"\n")

time.sleep(0.2)

# Creación de las entidades T-Cont (colas). Cada servicio Ethernet está asociado
# con un T-Cont y está vinculado a un Alloc-ID

i=0

while i < num_servicios_Video:

    tcont = "t-cont set slot-id 128 t-cont-id " + str(tcont_ID[i]).strip('[]') + " alloc-id " +
str(alloc_ID[i]).strip('[]') + " \n"

    tn.write(tcont.encode('ascii') + b"\n")

    time.sleep(0.2)

```

i=i+1

# Creación de MAC Bridge Service Profile. Se asociará esta entidad con los MAC Bridge

# Port Configuration Data a través del bridge-group-id en los siguientes comandos.

```
macservice = "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 \n"
```

# Creación del primer MAC Bridge Port Configuration Data. Irá asociado a la entidad

# extended-vlan-tagging-operation-config-data. Para ello, el tp-type (tipo del punto

# de terminación del MAC Bridge) ha de ser lan y el puntero tp-ptr debe tener el mismo

# valor que el nº de instancia de la entidad extended-vlan-tagging-operation-config-data.

# Esta entidad está vinculada a los servicios Ethernet.

```
macbridge1 = "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 \n"
```

```
tn.write(macservice.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

```
tn.write(macbridge1.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

# Creación del segundo MAC Bridge Port Configuration Data. Esta entidad está

# vinculada al servicio multicast. Se asocia a la entidad Multicast GEM Interworking

# Termination Poing. Para ello, el tipo de puntero (tp-type) ha de ser de tipo multicast

# (mc-gem) y el tp-ptr ha de coincidir con el nº de instancia la entidad Multicast GEM

# Interworking Termination Poing.

```
macbridge2 = "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 encap-method llc lanfcs-ind
forward \n"
```

```
tn.write(macbridge2.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

# Creación de los restantes Mac Bridge Port Configuration Data. Irán asociado a la entidades

```

# VLAN-tagging-filter-data mediante los números de instancia. También irán asociados a los
# GEM Interworking Termination Point mediante el tp-type (tipo gem) y el tp-ptr, que
# tiene que coincidir con el número de instancia del GEM Interworking Termination Point.
# Estas entidades están vinculadas a los servicios de tipo Ethernet.

i=0

while i < num_servicios_Video:

    macbridge3 = "mac-bridge-pcd create instance " + str(num_instancia[i]).strip('[]') + "
bridge-id-ptr 1 port-num " + str(num_instancia[i]).strip('[]') + " tp-type gem tp-ptr " +
str(num_instancia[i]).strip('[]') + " port-priority 0 port-path-cost 1 port-spanning-tree-ind true
encap-method llc lanfcs-ind forward \n"

    tn.write(macbridge3.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1

# 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)

# Creación de los GEM Port Network CTP, que irán asociado a los puertos especificados
# anteriormente para los servicios Ethernet. Los identificadores tienen que coincidir
# con el gem-port-nwk-ct-conn-ptr de los GEM Interworking Termination Point.

i=0

while i < num_servicios_Video:

```

```
gemport = "gem-port-network-ctp create instance " + str(num_instancia[i]).strip('[]') + "  
port-id " + str(port_ID[i]).strip('[]') + " t-cont-ptr " + str(puntero[i]).strip('[]') + " 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 \n"
```

```
tn.write(gemport.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

```
i=i+1
```

```
# Creación del Multicast GEM Interworking Termination Point (en este punto, se produce la  
# transformación de flujo de bytes a tramas GEM y viceversa). Esta entidad está vinculada al  
# servicio multicast. Se vincula al GEM Port Network CTP de tipo multicast mediante el campo  
# gem-port-nwk-ctp-conn-ptr. También se asocia con el MAC Bridge Port Configuration Data  
de
```

```
# tipo multicast. Para ello, hay que seleccionar como interwork-option mac-bridge y el campo  
# service-profile-ptr debe tener el valor que se indica en la declaración del comando.
```

```
# El número de instancia debe ser el mismo que el tp-ptr del MAC Bridge Point Configuration  
Data asociado.
```

```
multicast_geminterworking = "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 \n"
```

```
tn.write(multicast_geminterworking.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

```
# Creación de los GEM Interworking Termination Point (en este punto, se produce la  
# transformación de flujo de bytes a tramas GEM y viceversa). Estas entidades se vinculan  
# a los GEM Port Network CTP a través del campo gem-port-nwk-ctp-conn-ptr, que debe  
coincidir
```

```
# con el número de instancia que utilizado en el GEM Port Network CTP. Estas entidades  
también
```

```
# se asocian con los MAC Bridge Port Configuration. Para ello, hay que seleccionar como  
# interwork-option mac-bridge-lan y el campo service-profile-ptr debe ser un 1.
```

```
# El número de instancia debe ser el mismo que el tp-ptr del MAC Bridge Point Configuration  
Data.
```

```
i=0
```

```

while i < num_servicios_Video:

    geminterworking = "gem-interworking-termination-point create instance " +
str(num_instancia[i]).strip('[]') + " gem-port-nwk-ctp-conn-ptr " + str(num_instancia[i]).strip('[]')
+ " interwork-option mac-bridge-lan service-profile-ptr 1 interwork-tp-ptr 0 gal-profile-ptr 0 \n"

    tn.write(geminterworking.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1

# Creación de los VLAN Tagging Filter Data con los identificadores VLAN definidos arriba.

# Los números de instancia deben coincidir con los de los MAC Bridge Port Configuration
Data asociados.

i=0

while i < num_servicios_Video:

    vlantagging = "vlan-tagging-filter-data create instance " + str(num_instancia[i]).strip('[]') + "
forward-operation h-vid-a vlan-tag1 " + str(VLAN_ID[i]).strip('[]') + " vlan-priority1 7 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 \n"

    tn.write(vlantagging.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1

# Creación de la entidad Extended VLAN Tagging Operation Config Data, que será
configurada en el paso posterior.

# Sirve para gestionar los identificadores VLAN. Esta entidad está asociada al primer MAC
Bridge Port Configuration Data

# a través del número de instancia, que coincide con el tp-ptr del MAC Bridge Port
Configuration Data.

extendedvlan = "extended-vlan-tagging-operation-config-data create instance 257 association-
type ptp-eth-uni associated-me-ptr 257 \n"

tn.write(extendedvlan.encode('ascii') + b"\n")

time.sleep(0.2)

# Configuración de la entidadExtended VLAN Tagging Operation Config Data. Se debe
configurar

# para cada identificador VLAN.

```

```
i=0
```

```
while i < num_servicios_Video:
```

```
    extendedvlanconf = "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 " + str(VLAN_ID[i]).strip('[]') + " 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 " + str(VLAN_ID[i]).strip('[]') + " treatment-inner-tpid tpid-de-copy-from-
inner\n"
```

```
    tn.write(extendedvlanconf.encode('ascii') + b"\n")
```

```
    time.sleep(0.2)
```

```
    i=i+1
```

```
# El comando exit hace salir hacia el menú anterior del CLI en la estructura de menús
```

```
salir = "exit \n"
```

```
tn.write(salir.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

```
# Configuración de las reglas VLAN en la OLT asociadas a los puertos definidos al principio
de la función.
```

```
# Estos puertos son los vinculados a los servicios de tipo Ethernet.
```

```
i=0
```

```
while i < num_servicios_Video:
```

```
    reglasvlan = "vlan uplink configuration port-id " + str(port_ID[i]).strip('[]') + " min-cos 0
max-cos 7 de-bit disable primary-tag-handling false \n vlan uplink handling port-id " +
str(port_ID[i]).strip('[]') + " primary-vlan none destination datapath c-vlan-handling no-change s-
vlan-handling no-change new-c-vlan 0 new-s-vlan 0 \n"
```

```
    tn.write(reglasvlan.encode('ascii') + b"\n")
```

```
    time.sleep(0.2)
```

```
    i=i+1
```

```
# Creación de los perfiles de ancho de banda en sentido Downstream con los parámetros
```

```
# definidos por el usuario anteriormente.
```

```
i=0
```

```

while i < num_servicios_Video:

    perfiledownstreamconf = "policing downstream profile committed-max-bw " +
str(BW_Downstream_GR[i]).strip('[]') + " committed-burst-size 1023 excess-max-bw " +
str(BW_Downstream_Excess[i]).strip('[]') + " excess-burst-size 1023 \n"

    tn.write(perfiledownstreamconf.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1

# Nombre del fichero en el que se guardará la configuración del servicio de Internet + Video
nombre_archivo = 'Servicio_Internet+Video_ONU_MAC_' + MAC_ONU + '.txt'
# Al crear los perfiles de ancho de banda en sentido de bajada, el CLI devuelve un
# índice de perfil. Para asociar los perfiles creados, se deben utilizar esos índices
# de perfil y asociarlos a los puertos definidos anteriormente. Para ello, se vuelcan los datos
# enviados y recibidos del CLI en un fichero, del que se extraerán esos índices.

datos_perfil = tn.read_very_eager().decode()

outfile = open(nombre_archivo, 'a')

outfile.write(datos_perfil)

outfile.close()

outfile = open(nombre_archivo, 'r')

lines = outfile.readlines()

true_ds_profile = 0

# Se busca el índice de perfil que primero aparezca en el fichero.

i=0

for i in range (0,500):

    if true_ds_profile == 1:

        break

    j = str(i)

    cadena = 'downstream_profile_index: ' + j + ' '

    for line in lines:

        if cadena in line:

            j = int(j)

            ds_profile_index[0] = j

```

```
true_ds_profile = 1

outfile.close()

# El índice encontrado corresponde al primer servicio. En caso de haber más, el índice
# de perfil será ese aumentado en una unidad.
i=0
for i in range (0,num_servicios_Video):
    ds_profile_index[i] = ds_profile_index[0] + i

# Asignación de los perfiles de ancho de banda Downstream a los puertos correspondientes
# mediante los índices de perfil buscados anteriormente. Estos anchos de banda se asignan
# a los servicios de tipo Ethernet. Para el servicio multicast, no hay que definir anchos
# de banda.
i=0
while i < num_servicios_Video:
    perfildownstreamassign = "policing downstream port-configuration entity port-id " +
str(port_ID[i]).strip('[]') + " ds-profile-index " + str(ds_profile_index[i]).strip('[]') + " \n"
    tn.write(perfildownstreamassign.encode('ascii') + b"\n")
    time.sleep(0.2)
    i=i+1

# El comando exit hace salir hacia el menú anterior del CLI en la estructura de menús
salir = "exit \n"
tn.write(salir.encode('ascii') + b"\n")
time.sleep(0.2)

# Desde el menú de configuración del algoritmo DBA, se definirían los perfiles de
# ancho de banda en sentido Upstream
dba = "pon \n dba pythagoras 0 \n "
```

```

tn.write(dba.encode('ascii') + b"\n")

time.sleep(0.2)

# Cada perfil upstream irá asociado a un Alloc-ID y estará configurado con los
# parámetros que haya definido el usuario. Como ya se se explicó anteriormente,
# solo los servicios de tipo Ethernet tienen asociados Alloc-IDs; el de tipo
# multicast, no.
i=0
while i < num_servicios_Video:
    perfilupstream = "sla " + str(alloc_ID[i]).strip('[]') + " service data status-report nsr gr-bw "
+ str(BW_Upstream_GR[i]).strip('[]') + " gr-fine 0 be-bw " + str(BW_Upstream_BE[i]).strip('[]')
+ " be-fine 0 \n"

    tn.write(perfilupstream.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1

# El comando end hace salir directamente al modo privilegiado en la estructura
# de menú del CLI
final = "end \n"
tn.write(final.encode('ascii') + b"\n")
time.sleep(0.2)

# Se vuelcan todos los datos en el fichero definido anteriormente (la opción 'a'
# hace que los datos se añadan al final del fichero) de forma que el fichero recogerá
# toda la configuración del servicio de Internet + Vídeo
data = tn.read_very_eager().decode()
outfile = open(nombre_archivo, 'a')
outfile.write(data)
outfile.write("\n\n\n")
outfile.close()

# Una vez configurado el servicio, se muestra un mensaje al usuario

```

```
print("Servicio de Internet+VÍdeo configurado. \n")

# Actualización del fichero XML con los datos configurados
doc = etree.parse("configuracionGPON.xml")

# Extracción del elemento raíz
redGPON = doc.getroot()

# Obtención del índice correspondiente a la ONU que se está configurando.
# Para ello, se hace una comparación de las diferentes direcciones MAC con
# la de la ONU a configurar.
index=0
true_index=0
while true_index==0:
    for attr,value in redGPON[index].items():
        if value == MAC_ONU:
            true_index=1
            break
    index=index+1

# Se borra cualquier configuración anterior que en este caso, al configurar un
# nuevo servicio, se va a desechar.
num_servicios = len(redGPON[index])
i=0
while i<num_servicios:
    redGPON[index].remove(redGPON[index][0])
    i=i+1

# Se actualiza la parte del árbol correspondiente a la ONU en cuestión con los
# parámetros que ha definido el usuario. Se utiliza la función SubElement para
# crear el servicio así como los parámetros del mismo.
```

```

i=0

while i < num_servicios_Video:

    Servicio = etree.SubElement(redGPON[index], "Servicio", tipo='Internet+Video')

    VLAN = etree.SubElement(redGPON[index][i], "VLAN_ID").text =
str(VLAN_ID[i]).strip('[]')

    BW_Down_GR = etree.SubElement(redGPON[index][i], "BW_Down_GR").text =
str(BW_Downstream_GR[i]).strip('[]')

    BW_Down_Excess = etree.SubElement(redGPON[index][i], "BW_Down_Excess").text =
str(BW_Downstream_Excess[i]).strip('[]')

    BW_Up_GR = etree.SubElement(redGPON[index][i], "BW_Up_GR").text =
str(BW_Upstream_GR[i]).strip('[]')

    BW_Up_BE = etree.SubElement(redGPON[index][i], "BW_Up_BE").text =
str(BW_Upstream_BE[i]).strip('[]')

    i=i+1

# Finalmente se crea el nuevo árbol modificado y se escribe en el fichero XML

doc = etree.ElementTree(redGPON)

doc.write("configuracionGPON.xml")

return

# Función modificar_configuracion(ID_ONU,MAC_ONU,nombre_archivo): esta función
permite
# cambiar la configuración existente en una ONU. Toma como argumento el identificador
# de ONU, la dirección MAC y el nombre del archivo donde está la configuración que será
# reemplazada. La función en primer lugar borra la configuración existente tomando datos de este
# fichero y posteriormente, llama a las funciones de servicio_Internet o servicio_Video
# dependiendo de lo que el usuario haya seleccionado para la nueva configuración.
def modificar_configuracion(ID_ONU,MAC_ONU,nombre_archivo):

    # 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 abre en modo lectura el fichero en el que está la configuración antigua
outfile = open(nombre_archivo, 'r')
# Se almacenan todas las líneas del fichero
lines = outfile.readlines()
port_ID = 0

# En este bucle, se buscarán los puertos configurados y se borrarán los perfiles
# de anchos de banda asociados a estos puertos. En las funciones de servicio_Internet
# y servicio_Video que se llamarán posteriormente, se borrarán las entidades MIB, de forma
# que los servicios quedarán borrados por completo.
k=0
for k in range (0,10000):
    j = str(k)
```

```

# Se recorren todas las líneas del fichero hasta encontrar los puertos configurados
cadena = 'configuration entity port-id ' + j + ' '

for line in lines:
    if cadena in line:
        j = int(j)
        port_ID = j
        port_ID = str(port_ID)
        # Cada vez que se encuentra un puerto configurado, se borra el perfil de ancho
        # de banda asociado
        borrar_perfil = "configure \n olt-device 0 \n olt-channel 0 \n no policing downstream
port-configuration entity port-id " + port_ID + " \n end \n"
        tn.write(borrar_perfil.encode('ascii') + b"\n")
        time.sleep(0.1)

outfile.close()

# Finalmente se borra el fichero de configuración
os.remove(nombre_archivo)

# Actualización del archivo XML con los datos borrados (se actualizarán en las
# funciones correspondientes de Internet o Vídeo)
doc = etree.parse("configuracionGPON.xml")
# Extracción del elemento raíz
redGPON = doc.getroot()

# Obtención del índice correspondiente a la ONU cuya configuración se va a borrar.
# Para ello, se hace una comparación de las diferentes direcciones MAC con
# la de la ONU en cuestión.
index=0
true_index=0
while true_index==0:
    for attr,value in redGPON[index].items():

```

```
    if value == MAC_ONU:
        true_index=1
        break
    index=index+1

# Se borra la configuración de dicha ONU
num_servicios = len(redGPON[index])
i=0
while i<num_servicios:
    redGPON[index].remove(redGPON[index][0])
    i=i+1

# Finalmente se crea el nuevo árbol modificado y se escribe en el fichero XML
doc = etree.ElementTree(redGPON)
doc.write("configuracionGPON.xml")

# Una vez borrada la configuración, se pide al usuario que seleccione la nueva
# configuración que desea para la ONU. Dos opciones: servicio de Internet y
# servicio de Internet + Video.
true_opcion = 0
while true_opcion == 0:
    print("\n¿Qué servicio quiere crear ahora en esta ONU? Escoja la opción:")
    print("1 - Servicio de Internet")
    print("2 - Servicio de Internet + Vídeo")
    opcion = input()
    opcion = int(opcion)
    # Si el usuario no selecciona una opción válida, se le vuelve a pedir.
    if opcion == 1 or opcion == 2:
        true_opcion = 1
    else:
```

```

print("Opción no válida. Vuelva a probar.")

# En caso de seleccionar el servicio de Internet, se le pide el nº de servicios.
# Este nº podría ser mayor que 2 pero realmente, no tiene sentido configurar más de 2
# servicios puesto que solo hay 2 VLAN que proporcionen servicio a esta red.
if opcion == 1:
    print("\nConfiguración del servicio de Internet:")
    cont=0
    true_Internet = 0
    while true_Internet == 0:
        if cont == 0:
            print("Número de servicios de datos:")
        else:
            print("Número de servicios no válido. Vuelva a probar:")
        cont = cont+1
        num_servicios_Internet = input()
        num_servicios_Internet = int(num_servicios_Internet)
        if num_servicios_Internet > -1 and num_servicios_Internet < 3:
            true_Internet = 1

    print("\n")
    # En caso de que el usuario seleccione un número de servicios de Internet que no sea 0,
    # se llama a la función de servicio_Internet para configurar la ONU con los nuevos
    # parámetros
    if num_servicios_Internet > 0:
        servicio_Internet(ID_ONU,MAC_ONU,num_servicios_Internet)

# En caso de seleccionar el servicio de Internet+Video, se le pide el nº de servicios.
# Este nº podría ser mayor que 2 pero realmente, no tiene sentido configurar más de 2
# servicios puesto que solo hay 2 VLAN que proporcionen servicio a esta red.
elif opcion == 2:

```

```
print("\nConfiguración del servicio de Internet + Vídeo:")
cont=0
true_Video = 0
while true_Video == 0:
    if cont == 0:
        print("Número de servicios de vídeo:")
    else:
        print("Número de servicios no válido. Vuelva a probar:")
    cont = cont+1
    num_servicios_Video = input()
    num_servicios_Video = int(num_servicios_Video)
    if num_servicios_Video > -1 and num_servicios_Video < 3:
        true_Video = 1

print("\n")

# En caso de que el usuario seleccione un número de servicios de Internet + Vídeo que no
sea 0,

# se llama a la función de servicio_Internet para configurar la ONU con los nuevos
parámetros

if num_servicios_Video > 0:
    servicio_Video(ID_ONU,MAC_ONU,num_servicios_Video)

return

# Función borrar_configuracion(ID_ONU,MAC_ONU,nombre_archivo): esta función permite
# borrar la configuración de una ONU. Toma como argumento el identificador
# de ONU, la dirección MAC y el nombre del archivo donde está la configuración que
# será borrada.

def borrar_configuracion(ID_ONU,MAC_ONU,nombre_archivo):
```

```

# 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 abre en modo lectura el fichero en el que está la configuración a borrar.
outfile = open(nombre_archivo, 'r')

# Se almacenan todas las líneas del fichero.
lines = outfile.readlines()
port_ID = 0

# En este bucle, se buscarán los puertos configurados y se borrarán los perfiles
# de anchos de banda asociados a estos puertos. Posteriormente, se borrarán las entidades
# que forman los servicios.
k=0
for k in range (0,10000):

```

```
j = str(k)

# Se recorren todas las líneas del fichero hasta encontrar los puertos configurados.
cadena = 'configuration entity port-id ' + j + ' '
for line in lines:
    if cadena in line:
        j = int(j)
        port_ID = j
        port_ID = str(port_ID)
        # Cada vez que se encuentra un puerto configurado, se borra el perfil de ancho
        # de banda asociado.
        borrar_perfil = "configure \n olt-device 0 \n olt-channel 0 \n no policing downstream
port-configuration entity port-id " + port_ID + " \n end \n"
        tn.write(borrar_perfil.encode('ascii') + b"\n")
        time.sleep(0.1)

outfile.close()
ID_ONU = str(ID_ONU)
# Tras borrar los perfiles de ancho de banda, se borra las entidades MIB presentes en
# el canal OMCI asociado a la ONU.
borrar_MIB = "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
end \n"
tn.write(borrar_MIB.encode('ascii') + b"\n")
time.sleep(2)
# Finalmente se borra el fichero de configuración.
os.remove(nombre_archivo)
# Una vez borrado, se muestra un mensaje informando al usuario.
print("\nLa configuración de la ONU con número de serie " + MAC_ONU + " ha sido
borrada.\n")

# Actualización del archivo XML con los datos borrados
```

```

doc = etree.parse("configuracionGPON.xml")

# Extracción del elemento raíz
redGPON = doc.getroot()

# Obtención del índice correspondiente a la ONU cuya configuración se va a borrar.
# Para ello, se hace una comparación de las diferentes direcciones MAC con
# la de la ONU en cuestión.
index=0
true_index=0
while true_index==0:
    for attr,value in redGPON[index].items():
        if value == MAC_ONU:
            true_index=1
            break
        index=index+1

# Se borra la configuración de dicha ONU
num_servicios = len(redGPON[index])
i=0
while i<num_servicios:
    redGPON[index].remove(redGPON[index][0])
    i=i+1

# Finalmente se crea el nuevo árbol modificado y se escribe en el fichero XML
doc = etree.ElementTree(redGPON)
doc.write("configuracionGPON.xml")

return

# Función ver_configuracion(ID_ONU,MAC_ONU,nombre_archivo): esta función permite
# ver la configuración presente en una ONU. Toma como argumento el identificador

```

# de ONU, la dirección MAC y el nombre del archivo donde está la configuración.

# Para mostrar esta configuración, se utiliza el fichero XML que se va actualizando

# con las llamadas a cada función.

```
def ver_configuracion(MAC_ONU):
```

```
    # Extracción del elemento raíz del fichero XML.
```

```
    doc = etree.parse("configuracionGPON.xml")
```

```
    redGPON = doc.getroot()
```

```
    # Obtención del índice correspondiente a la ONU cuya configuración se va a mostrar.
```

```
    # Para ello, se hace una comparación de las diferentes direcciones MAC con
```

```
    # la de la ONU en cuestión.
```

```
    index=0
```

```
    true_index=0
```

```
    while true_index == 0:
```

```
        for attr,value in redGPON[index].items():
```

```
            if value == MAC_ONU:
```

```
                true_index=1
```

```
                break
```

```
            index = index+1
```

```
    # Obtención del tipo de servicio: Internet o Internet + Video.
```

```
    # Para ello, se obtiene el valor que toma el subelemento Servicio.
```

```
    for attr,value in redGPON[index][0].items():
```

```
        if value == 'Internet':
```

```
            tipo_servicio = value
```

```
        elif value == 'Internet+Video':
```

```
            tipo_servicio = value
```

```
    # Obtención del nº de servicios que había configurados en base a la longitud
```

```

# del vector de subelementos de la ONU.

num_servicios = len(redGPON[index])

print("\n")

# Se muestra el tipo de servicio, el número de servicios y los parámetros que los forman.

print("Tipo de servicio: " + tipo_servicio)

print("Número de servicios:", num_servicios)

print("Parámetros:\n")

# Para mostrar los parámetros, se accede a los subelementos de los servicios.

i=0

while i<num_servicios:

    print("El identificador VLAN del servicio", i+1,"es " + redGPON[index][i][0].text)

    print("El ancho de banda garantizado Downstream en Kbps del servicio", i+1, "es " +
redGPON[index][i][1].text)

    print("El ancho de banda en exceso Downstream en Kbps del servicio", i+1, "es " +
redGPON[index][i][2].text)

    print("El ancho de banda garantizado Upstream en Mbps del servicio", i+1, "es " +
redGPON[index][i][3].text)

    print("El ancho de banda Best Effort Upstream en Mbps del servicio", i+1, "es " +
redGPON[index][i][4].text)

    print("\n")

    i=i+1

return

# Función cargar_configuracion(ID_ONU,MAC_ONU): esta función permite recuperar y
# cargar la configuración que había en una ONU antes de que la red fuera apagada o
# se cambiara al modo de gestión TGMS. Con esta función, se solventa el problema de la
# no persistencia del CLI. Toma como parámetros el ID de la ONU y la dirección MAC. Utiliza
# los datos guardados en el fichero XML para cargar la configuración que se había perdido.
def cargar_configuracion(ID_ONU,MAC_ONU):

    # Extracción del elemento raíz del fichero XML.

```

```
doc = etree.parse("configuracionGPON.xml")
redGPON = doc.getroot()

# Obtención del índice correspondiente a la ONU cuya configuración se va a cargar.
# Para ello, se hace una comparación de las diferentes direcciones MAC con
# la de la ONU en cuestión.
index=0
true_index=0
while true_index == 0:
    for attr,value in redGPON[index].items():
        if value == MAC_ONU:
            true_index=1
            break
        index = index+1

# Obtención del nº de servicios que había configurados en base a la longitud
# del vector de subelementos de la ONU.
num_servicios = len(redGPON[index])

# Este if obtiene el tipo de servicio (si es que había alguno) configurado.
# Para ello, se obtiene el valor que toma el subelemento Servicio.
if num_servicios>0:
    for attr,value in redGPON[index][0].items():
        if value == 'Internet':
            tipo_servicio = value
        elif value == 'Internet+Video':
            tipo_servicio = value
    elif num_servicios == 0:
        # En caso de que no hubiera servicios configurados, se informa con un mensaje
        # y se retorna al main del programa.
```

```

print("\nLa ONU con número de serie " + MAC_ONU + " no estaba configurada.\n")

return

# Si el servicio configurado anteriormente era de Internet, se muestra un mensaje informando
de ello.

if tipo_servicio == 'Internet':

    print("\nLa ONU con número de serie " + MAC_ONU + " tenía configurado anteriormente
servicio de Internet.")

# A partir de aquí se reproduce la función servicio_Internet con la DIFERENCIA
# de que los parámetros ya no se piden al usuario, sino que se toman del
# del fichero XML.

# ¡IMPORTANTE! Las líneas que siguen son iguales que las de la función servicio_Internet
# PARA LA EXPLICACIÓN DE LAS SIGUIENTES LÍNEAS, VER FUNCIÓN
servicio_Internet

port_ID = []
alloc_ID = []
tcont_ID = []
num_instancia = []
puntero = []
ds_profile_index = []

i=0

while i < num_servicios:

    port_ID.append(200+100*ID_ONU+i)
    alloc_ID.append(200+100*ID_ONU+i)
    num_instancia.append(i+2)
    tcont_ID.append(i)
    puntero.append(32768+i)
    ds_profile_index.append(i)

    i=i+1

```

```
print("\n")

host = "172.26.128.38"
port = "4551"
password1 = "TLNT25"
password2 = "TLNT145"
enable = "enable"

tn = telnetlib.Telnet(host,port,1)
tn.write(password1.encode('ascii') + b"\n")
time.sleep(0.2)
tn.write(enable.encode('ascii') + b"\n")
time.sleep(0.2)
tn.write(password2.encode('ascii') + b"\n")
time.sleep(0.2)

ID_ONU = str(ID_ONU)

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)

i=0
while i < num_servicios:
    allocID = "alloc-id " + str(alloc_ID[i]).strip('[]') + " \n"
    tn.write(allocID.encode('ascii') + b"\n")
    time.sleep(0.2)
    i=i+1
```

```

salir = "exit \n"

tn.write(salir.encode('ascii') + b"\n")

time.sleep(0.2)

i=0

while i < num_servicios:

    portalloc = "port " + str(port_ID[i]).strip('[]') + " alloc-id " + str(alloc_ID[i]).strip('[]') + "
\n"

    tn.write(portalloc.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1

omci = "onu-omci " + ID_ONU + " \n"

tn.write(omci.encode('ascii') + b"\n")

time.sleep(0.2)

i=0

while i < num_servicios:

    tcont = "t-cont set slot-id 128 t-cont-id " + str(tcont_ID[i]).strip('[]') + " alloc-id " +
str(alloc_ID[i]).strip('[]') + " \n"

    tn.write(tcont.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1

macservice = "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 \n"

macbridge1 = "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 \n"

tn.write(macservice.encode('ascii') + b"\n")

time.sleep(0.2)

tn.write(macbridge1.encode('ascii') + b"\n")

```

```
time.sleep(0.2)
```

```
i=0
```

```
while i < num_servicios:
```

```
    macbridge2 = "mac-bridge-pcd create instance " + str(num_instancia[i]).strip('[]') + "  
bridge-id-ptr 1 port-num " + str(num_instancia[i]).strip('[]') + " tp-type gem tp-ptr " +  
str(num_instancia[i]).strip('[]') + " port-priority 0 port-path-cost 1 port-spanning-tree-ind true  
encap-method llc lanfcs-ind forward \n"
```

```
    tn.write(macbridge2.encode('ascii') + b"\n")
```

```
    time.sleep(0.2)
```

```
    i=i+1
```

```
i=0
```

```
while i < num_servicios:
```

```
    gemport = "gem-port-network-ctp create instance " + str(num_instancia[i]).strip('[]') + "  
port-id " + str(port_ID[i]).strip('[]') + " t-cont-ptr " + str(puntero[i]).strip('[]') + " direction  
bidirectional traffic-mgnt-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.encode('ascii') + b"\n")
```

```
    time.sleep(0.2)
```

```
    i=i+1
```

```
i=0
```

```
while i < num_servicios:
```

```
    geminterworking = "gem-interworking-termination-point create instance " +  
str(num_instancia[i]).strip('[]') + " gem-port-nwk-ctp-conn-ptr " + str(num_instancia[i]).strip('[]')  
+ " interwork-option mac-bridge-lan service-profile-ptr 1 interwork-tp-ptr 0 gal-profile-ptr 0 \n"
```

```
    tn.write(geminterworking.encode('ascii') + b"\n")
```

```
    time.sleep(0.2)
```

```
    i=i+1
```

```
i=0
```

```
while i < num_servicios:
```

```

vlantagging = "vlan-tagging-filter-data create instance " + str(num_instancia[i]).strip('[]')
+ " forward-operation h-vid-a vlan-tag1 " + redGPON[index][i][0].text + " vlan-priority1 7 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 \n"

```

```

tn.write(vlantagging.encode('ascii') + b"\n")

```

```

time.sleep(0.2)

```

```

i=i+1

```

```

extendedvlan = "extended-vlan-tagging-operation-config-data create instance 257
association-type pptp-eth-uni associated-me-ptr 257 \n"

```

```

tn.write(extendedvlan.encode('ascii') + b"\n")

```

```

time.sleep(0.2)

```

```

i=0

```

```

while i < num_servicios:

```

```

    extendedvlanconf = "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 " + redGPON[index][i][0].text + " 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 " + redGPON[index][i][0].text + " treatment-inner-tpid tpid-de-copy-from-
inner\n"

```

```

    tn.write(extendedvlanconf.encode('ascii') + b"\n")

```

```

    time.sleep(0.2)

```

```

    i=i+1

```

```

salir = "exit \n"

```

```

tn.write(salir.encode('ascii') + b"\n")

```

```

time.sleep(0.2)

```

```

i=0

```

```

while i < num_servicios:

```

```

    reglasvlan = "vlan uplink configuration port-id " + str(port_ID[i]).strip('[]') + " min-cos 0
max-cos 7 de-bit disable primary-tag-handling false \n vlan uplink handling port-id " +

```

```
str(port_ID[i]).strip('[]') + " primary-vlan none destination datapath c-vlan-handling no-change s-  
vlan-handling no-change new-c-vlan 0 new-s-vlan 0 \n"
```

```
tn.write(reglasvlan.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

```
i=i+1
```

```
i=0
```

```
while i < num_servicios:
```

```
    perfildownstreamconf = "policing downstream profile committed-max-bw " +  
redGPON[index][i][1].text + " committed-burst-size 1023 excess-max-bw " +  
redGPON[index][i][2].text + " excess-burst-size 1023 \n"
```

```
tn.write(perfildownstreamconf.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

```
i=i+1
```

```
nombre_archivo = 'Servicio_Internet_ONU_MAC_' + MAC_ONU + '.txt'
```

```
datos_perfil = tn.read_very_eager().decode()
```

```
outfile = open(nombre_archivo, 'a')
```

```
outfile.write(datos_perfil)
```

```
outfile.close()
```

```
outfile = open(nombre_archivo, 'r')
```

```
lines = outfile.readlines()
```

```
true_ds_profile = 0
```

```
i=0
```

```
for i in range (0,500):
```

```
    if true_ds_profile == 1:
```

```
        break
```

```
    j = str(i)
```

```
    cadena = 'downstream_profile_index: ' + j + ''
```

```
    for line in lines:
```

```
        if cadena in line:
```

```

        j = int(j)
        ds_profile_index[0] = j
        true_ds_profile = 1

outfile.close()

i=0
for i in range (0,num_servicios):
    ds_profile_index[i] = ds_profile_index[0] + i

i=0
while i < num_servicios:
    perfildownstreamassign = "policing downstream port-configuration entity port-id " +
str(port_ID[i]).strip('[]') + " ds-profile-index " + str(ds_profile_index[i]).strip('[]') + " \n"
    tn.write(perfildownstreamassign.encode('ascii') + b"\n")
    time.sleep(0.2)
    i=i+1

salir = "exit \n"
tn.write(salir.encode('ascii') + b"\n")
time.sleep(0.2)

dba = "pon \n dba pythagoras 0 \n "
tn.write(dba.encode('ascii') + b"\n")
time.sleep(0.2)

i=0
while i < num_servicios:
    perfilupstream = "sla " + str(alloc_ID[i]).strip('[]') + " service data status-report nsr gr-bw
" + redGPON[index][i][3].text + " gr-fine 0 be-bw " + redGPON[index][i][4].text + " be-fine 0
\n"
    tn.write(perfilupstream.encode('ascii') + b"\n")

```

```
time.sleep(0.2)

i=i+1

final = "end \n"
tn.write(final.encode('ascii') + b"\n")
time.sleep(0.2)

data = tn.read_very_eager().decode()
outfile = open(nombre_archivo, 'a')
outfile.write(data)
outfile.write("\n\n\n")
outfile.close()

# Aquí finaliza la configuración del servicio de Internet con los datos tomados
# del fichero XML para recuperar la configuración anterior. Una vez establecida esta
# nueva configuración, se ha vuelto a crear el fichero txt con los datos del CLI para
# su utilización en otras funciones. Es importante recordar que estos ficheros txt deben
# ser borados manualmente cada vez que se apaga la red o se cambia al TGMS.

# Se muestra un mensaje al usuario informando que la configuración se ha restablecido
# correctamente y se le pregunta si desea ver dicha configuración.
print("El servicio de Internet configurado anteriormente ha sido restablecido.\n")
print("¿Desea ver la configuración cargada? Responda SI o NO")
true_respuesta=0
while true_respuesta == 0:
    respuesta = input()
    # En caso de que quiera ver la configuración, se llama a la función ver_configuración
    if respuesta == "SI":
        ver_configuracion(MAC_ONU)
        true_respuesta = 1
```

```

elif respuesta == "NO":
    print("\n")
    true_respuesta = 1
else:
    print("Vuelva a introducir su respuesta (SI o NO)")

# Si el servicio configurado anteriormente era de Internet + Video, se muestra un mensaje
informando de ello.

elif tipo_servicio == 'Internet+Video':
    print("\nLa ONU con número de serie " + MAC_ONU + " tenía configurado anteriormente
servicio de Internet y Video.")

# A partir de aquí se reproduce la función servicio_Video con la DIFERENCIA
# de que los parámetros ya no se piden al usuario, sino que se toman del
# del fichero XML.

# ¡IMPORTANTE! Las líneas que siguen son iguales que las de la función servicio_Video
# PARA LA EXPLICACIÓN DE LAS SIGUIENTES LÍNEAS, VER FUNCIÓN
servicio_Video

port_ID = []
alloc_ID = []
tcont_ID = []
num_instancia = []
puntero = []
ds_profile_index = []
i=0

while i < num_servicios:
    port_ID.append(200+100*ID_ONU+i)
    alloc_ID.append(200+100*ID_ONU+i)
    num_instancia.append(i+3)
    tcont_ID.append(i)

```

```
puntero.append(32768+i)
ds_profile_index.append(i)
i=i+1

host = "172.26.128.38"
port = "4551"
password1 = "TLNT25"
password2 = "TLNT145"
enable = "enable"

tn = telnetlib.Telnet(host,port,1)
tn.write(password1.encode('ascii') + b"\n")
time.sleep(0.2)
tn.write(enable.encode('ascii') + b"\n")
time.sleep(0.2)
tn.write(password2.encode('ascii') + b"\n")
time.sleep(0.2)

ID_ONU = str(ID_ONU)

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)

i=0
while i < num_servicios:
    allocID = "alloc-id " + str(alloc_ID[i]).strip('[]') + " \n"
    tn.write(allocID.encode('ascii') + b"\n")
    time.sleep(0.2)
```

```

    i=i+1

salir = "exit \n"
tn.write(salir.encode('ascii') + b"\n")
time.sleep(0.2)

i=0
while i < num_servicios:
    portalloc = "port " + str(port_ID[i]).strip('[]') + " alloc-id " + str(alloc_ID[i]).strip('[]') + "
\n"
    tn.write(portalloc.encode('ascii') + b"\n")
    time.sleep(0.2)
    i=i+1

omci = "onu-omci " + ID_ONU + " \n"
tn.write(omci.encode('ascii') + b"\n")
time.sleep(0.2)

i=0
while i < num_servicios:
    tcont = "t-cont set slot-id 128 t-cont-id " + str(tcont_ID[i]).strip('[]') + " alloc-id " +
str(alloc_ID[i]).strip('[]') + " \n"
    tn.write(tcont.encode('ascii') + b"\n")
    time.sleep(0.2)
    i=i+1

macservice = "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 \n"

macbridge1 = "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 \n"

tn.write(macservice.encode('ascii') + b"\n")

```

```
time.sleep(0.2)
```

```
tn.write(macbridge1.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

```
macbridge2 = "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 encap-method llc lanfcs-ind  
forward \n"
```

```
tn.write(macbridge2.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

```
i=0
```

```
while i < num_servicios:
```

```
macbridge3 = "mac-bridge-pcd create instance " + str(num_instancia[i]).strip('[]') + "  
bridge-id-ptr 1 port-num " + str(num_instancia[i]).strip('[]') + " tp-type gem tp-ptr " +  
str(num_instancia[i]).strip('[]') + " port-priority 0 port-path-cost 1 port-spanning-tree-ind true  
encap-method llc lanfcs-ind forward \n"
```

```
tn.write(macbridge3.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

```
i=i+1
```

```
gemport_multicast = "gem-port-network-ctp create instance 2 port-id 4094 t-cont-ptr 0  
direction ani-to-uni traffic-mgnt-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)
```

```
i=0
```

```
while i < num_servicios:
```

```
gemport = "gem-port-network-ctp create instance " + str(num_instancia[i]).strip('[]') + "  
port-id " + str(port_ID[i]).strip('[]') + " t-cont-ptr " + str(puntero[i]).strip('[]') + " direction  
bidirectional traffic-mgnt-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.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

```
i=i+1
```

```
multicast_geminterworking = "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 \n"
```

```
tn.write(multicast_geminterworking.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

```
i=0
```

```
while i < num_servicios:
```

```
geminterworking = "gem-interworking-termination-point create instance " +
str(num_instancia[i]).strip('[]') + " gem-port-nwk-ctp-conn-ptr " + str(num_instancia[i]).strip('[]')
+ " interwork-option mac-bridge-lan service-profile-ptr 1 interwork-tp-ptr 0 gal-profile-ptr 0 \n"
```

```
tn.write(geminterworking.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

```
i=i+1
```

```
i=0
```

```
while i < num_servicios:
```

```
vlantagging = "vlan-tagging-filter-data create instance " + str(num_instancia[i]).strip('[]')
+ " forward-operation h-vid-a vlan-tag1 " + redGPON[index][i][0].text + " vlan-priority1 7
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 \n"
```

```
tn.write(vlantagging.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

```
i=i+1
```

```
extendedvlan = "extended-vlan-tagging-operation-config-data create instance 257
association-type pptp-eth-uni associated-me-ptr 257 \n"
```

```
tn.write(extendedvlan.encode('ascii') + b"\n")
```

```
time.sleep(0.2)
```

```
i=0
```

```
while i < num_servicios:

    extendedvlanconf = "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 " + redGPON[index][i][0].text + " 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 " + redGPON[index][i][0].text + " treatment-inner-tpid tpid-de-copy-from-
inner\n"

    tn.write(extendedvlanconf.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1

salir = "exit \n"

tn.write(salir.encode('ascii') + b"\n")

time.sleep(0.2)

i=0

while i < num_servicios:

    reglasvlan = "vlan uplink configuration port-id " + str(port_ID[i]).strip('[]') + " min-cos 0
max-cos 7 de-bit disable primary-tag-handling false \n vlan uplink handling port-id " +
str(port_ID[i]).strip('[]') + " primary-vlan none destination datapath c-vlan-handling no-change s-
vlan-handling no-change new-c-vlan 0 new-s-vlan 0 \n"

    tn.write(reglasvlan.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1

i=0

while i < num_servicios:

    perfildownstreamconf = "policing downstream profile committed-max-bw " +
redGPON[index][i][1].text + " committed-burst-size 1023 excess-max-bw " +
redGPON[index][i][2].text + " excess-burst-size 1023 \n"

    tn.write(perfildownstreamconf.encode('ascii') + b"\n")

    time.sleep(0.2)

    i=i+1
```

```

nombre_archivo = 'Servicio_Internet+Video_ONU_MAC_' + MAC_ONU + '.txt'
datos_perfil = tn.read_very_eager().decode()
outfile = open(nombre_archivo, 'a')
outfile.write(datos_perfil)
outfile.close()
outfile = open(nombre_archivo, 'r')
lines = outfile.readlines()
true_ds_profile = 0
i=0
for i in range (0,100):
    if true_ds_profile == 1:
        break
    j = str(i)
    cadena = 'downstream_profile_index: ' + j + ''
    for line in lines:
        if cadena in line:
            j = int(j)
            ds_profile_index[0] = j
            true_ds_profile = 1
outfile.close()

i=0
for i in range (0,num_servicios):
    ds_profile_index[i] = ds_profile_index[0] + i

i=0
while i < num_servicios:
    perfiledownstreamassign = "policing downstream port-configuration entity port-id " +
str(port_ID[i]).strip('[]') + " ds-profile-index " + str(ds_profile_index[i]).strip('[]') + " \n"

```

```
tn.write(perfildownstreamassign.encode('ascii') + b"\n")
time.sleep(0.2)
i=i+1

salir = "exit \n"
tn.write(salir.encode('ascii') + b"\n")
time.sleep(0.2)

dba = "pon \n dba pythagoras 0 \n "
tn.write(dba.encode('ascii') + b"\n")
time.sleep(0.2)

i=0
while i < num_servicios:
    perfilupstream = "sla " + str(alloc_ID[i]).strip('[]') + " service data status-report nsr gr-bw
" + redGPON[index][i][3].text + " gr-fine 0 be-bw " + redGPON[index][i][4].text + " be-fine 0
\n"
    tn.write(perfilupstream.encode('ascii') + b"\n")
    time.sleep(0.2)
    i=i+1

final = "end \n"
tn.write(final.encode('ascii') + b"\n")
time.sleep(0.2)

data = tn.read_very_eager().decode()
outfile = open(nombre_archivo, 'a')
outfile.write(data)
outfile.write("\n\n\n")
outfile.close()
```

```

# Aquí finaliza la configuración del servicio de Internet + Video con los datos tomados
# del fichero XML para recuperar la configuración anterior. Una vez establecida esta
# nueva configuración, se ha vuelto a crear el fichero txt con los datos del CLI para
# su utilización en otras funciones. Es importante recordar que estos ficheros txt deben
# ser borrados manualmente cada vez que se apaga la red o se cambia al TGMS.

# Se muestra un mensaje al usuario informando que la configuración se ha restablecido
# correctamente y se le pregunta si desea ver dicha configuración.

print("El servicio de Internet y Video configurado anteriormente ha sido restablecido.\n")
print("¿Desea ver la configuración cargada? Responda SI o NO")

true_respuesta=0

while true_respuesta == 0:
    respuesta = input()

    # En caso de que quiera ver la configuración, se llama a la función ver_configuración
    if respuesta == "SI":
        ver_configuracion(MAC_ONU)
        true_respuesta = 1
    elif respuesta == "NO":
        print("\n")
        true_respuesta = 1
    else:
        print("Vuelva a introducir su respuesta (SI o NO)")

return

# FUNCIÓN MAIN DEL PROGRAMA

# Desde aquí, se irán llamando a las funciones definidas arriba en función de lo que
# escoja el usuario.

# En primer lugar, se le da la bienvenida al programa de gestión y se le pide la dirección MAC
# de la ONU. Este será siempre el primer paso: el usuario deberá introducir la dirección MAC de

```

```
# la ONU que quiere manejar y a partir irá seleccionando qué hacer.
print("Bienvenido al programa de gestión de la red GPON.")
print("A continuación, introduzca la número de serie de la ONU y la acción que desea llevar a
cabo.\n")
# La función get_ID_ONU() es llamada al principio del programa para almacenar en el vector
# MAC las direcciones MAC de todas las ONUs que están conectadas a la red GPON.
MAC = get_ID_ONU()

# Este flag marca el fin del programa. Cuando se activa, se finaliza el bucle y
# se sale del programa.
true_FIN = 0
while true_FIN == 0:
    cont=0
    true_MAC = 0
    # Cada vez que se termina de realizar una acción con una ONU, se vuelve a este punto
    # en el que se pide de nuevo una dirección MAC para trabajar con una ONU.
    # La dirección MAC que introduce el usuario se compara con las del vector MAC. Mientras
    # no introduzca una reconocida por el programa, se le seguirá pidiendo de nuevo la MAC.
    # Esto es lo que se denominará menú principal.
    while true_MAC == 0:
        if cont == 0:
            print("Introduzca el número de serie de la ONU que va a configurar: ")
        else:
            print("Número de serie no reconocido. Introdúzcalo de nuevo: ")
        direccion_MAC = input()
        cont = cont+1
    # Bucle que recorre el vector en el que se almacenan las direcciones MAC
    for i in range(0,len(MAC)):
        if direccion_MAC == MAC[i]:
            print("\nEl número de serie ha sido reconocida correctamente.")
```

```

true_MAC = 1

ID_ONU = i

# Estos son los nombres que toman los archivos de configuracion, en caso de existir.
nombre_archivo1 = 'Servicio_Internet_ONU_MAC_' + direccion_MAC + '.txt'
nombre_archivo2 = 'Servicio_Internet+Video_ONU_MAC_' + direccion_MAC + '.txt'
# Si existe nombre_archivo1, la ONU tiene configurado un servicio de Internet.
if path.exists(nombre_archivo1):
    print("\nLa ONU con número de serie " + direccion_MAC + " ya tiene configurado servicio
de Internet.\n")

# EL programa permite al usuario ver, modificar o borrar la configuración
# del servicio de Internet así volver al menú principal o finalizar el programa.
true_opcion = 1
true_eleccion = 0
while true_eleccion == 0:
    print("¿Desea ver, cambiar o borrar la configuración? También puede volver al menú
principal o finalizar el programa.")
    print("0 - Volver al menú principal")
    print("1 - Ver configuración")
    print("2 - Cambiar configuración")
    print("3 - Borrar configuración")
    print("4 - Finalizar el programa")
    cambio = input()
    # Si introduce un 0, el programa vuelve al menú principal donde se pide
    # la dirección MAC de la ONU sobre la que se quiere trabajar.
    if cambio == "0":
        print("\nHa seleccionado volver al menú principal.\n")
        true_eleccion = 1
    # Si introduce un 1, llama a la función ver_configuracion.
    elif cambio == "1":
        ver_configuracion(direccion_MAC)
    # Si introduce un 2, llama a la función modificar_configuracion.

```

```
elif cambio == "2":
    modificar_configuracion(ID_ONU,direccion_MAC,nombre_archivo1)
# Si introduce un 3, llama a la función borrar_configuracion.
elif cambio == "3":
    borrar_configuracion(ID_ONU,direccion_MAC,nombre_archivo1)
# Si introduce un 4, el programa finaliza.
elif cambio == "4":
    print("La configuración de la red GPON ha finalizado.")
    true_eleccion = 1
    true_FIN = 1
# Si introduce un número no reconocido, se le pide que vuelva a introducirlo.
else:
    print("\nOpción no válida. Vuelva a intentarlo.\n")
# Una vez se hayan ejecutado las funciones de ver, modificar o borrar
# configuración, se le pregunta al usuario si desea seguir configurando la red o no.
if cambio == "1" or cambio == "2" or cambio == "3":
    true_eleccion = 1
    true_respuesta=0
    while true_respuesta == 0:
        print("¿Desea continuar configurando la red? Responda SI o NO")
        respuesta = input()
        # En caso de responder NO, el programa finaliza.
        if respuesta == "NO":
            print("La configuración de la red GPON ha finalizado.")
            true_respuesta = 1
            true_FIN = 1
        elif respuesta == "SI":
            # Si la respuesta es SI, el programa volverá al punto donde se
            # pide la dirección MAC de la ONU a configurar.
            print("\n")
```

```

        true_respuesta = 1

        break

    else:

        # Si la respuesta no es válida, se le vuelve a preguntar.

        print("Vuelva a introducir su respuesta (SI o NO)")

# Si existe nombre_archivo2, la ONU tiene configurado un servicio de Internet + Video.

elif path.exists(nombre_archivo2):

    print("\nLa ONU con número de serie " + direccion_MAC + " ya tiene configurado servicio
de Internet y vídeo.\n")

    # EL programa permite al usuario ver, modificar o borrar la configuración

    # del servicio de Internet + Video así como volver al menú principal o

    # finalizar el programa.

    true_opcion = 1

    true_eleccion = 0

    while true_eleccion == 0:

        print("¿Desea ver, cambiar o borrar la configuración? También puede volver al menú
principal o finalizar el programa.")

        print("0 - Volver al menú principal")

        print("1 - Ver configuración")

        print("2 - Cambiar configuración")

        print("3 - Borrar configuración")

        print("4 - Finalizar el programa")

        cambio = input()

        # Si introduce un 0, el programa vuelve al menú principal donde se pide

        # la dirección MAC de la ONU sobre la que se quiere trabajar.

        if cambio == "0":

            print("\nHa seleccionado volver al menú principal.\n")

            true_eleccion = 1

        # Si introduce un 1, llama a la función ver_configuracion.

        elif cambio == "1":

            ver_configuracion(direccion_MAC)

```

```
# Si introduce un 2, llama a la función modificar_configuracion.
elif cambio == "2":
    modificar_configuracion(ID_ONU,direccion_MAC,nombre_archivo2)
# Si introduce un 3, llama a la función borrar_configuracion.
elif cambio == "3":
    borrar_configuracion(ID_ONU,direccion_MAC,nombre_archivo2)
# Si introduce un 1, finaliza el programa.
elif cambio == "4":
    print("La configuración de la red GPON ha finalizado.")
    true_eleccion = 1
    true_FIN = 1
# Si introduce un número no reconocido, se le pide que vuelva a introducirlo.
else:
    print("\nOpción no válida. Vuelva a intentarlo.\n")
# Una vez se hayan ejecutado las funciones de ver, modificar o borrar
# configuración, se le pregunta al usuario si desea seguir configurando la red o no.
if cambio == "1" or cambio == "2" or cambio == "3":
    true_eleccion = 1
    true_respuesta=0
    while true_respuesta == 0:
        print("¿Desea continuar configurando la red? Responda SI o NO")
        respuesta = input()
        # En caso de responder NO, el programa finaliza.
        if respuesta == "NO":
            print("La configuración de la red GPON ha finalizado.")
            true_respuesta = 1
            true_FIN = 1
        # Si la respuesta es SI, el programa volverá al punto donde se
        # pide la dirección MAC de la ONU a configurar.
        elif respuesta == "SI":
```

```

        print("\n")

        true_respuesta = 1

    else:

        # Si la respuesta no es válida, se le vuelve a preguntar.

        print("Vuelva a introducir su respuesta (SI o NO)")

# En caso de que no existan los archivos de configuración, la ONU está sin configurar.
# Por tanto, se marcará a 0 el flag true_opcion y el programa entrará en el siguiente bucle.
else:

    true_opcion = 0

# Se define la variable opción, que permitirá en el siguiente bucle llamar a una función u otra
# dependiendo de la elección del usuario. Se le asigna un valor por defecto que no
# coincida con ninguna opción de las que llaman a otras funciones (ahora mismo, los
# valores de 0 a 4 están utilizados). Ver opciones más abajo.

opcion = 5

# Si el programa entra en este bucle, quiere decir que la ONU está sin configurar. Por tanto,
# el usuario podrá configurar para dicha ONU servicio de Internet, servicio de Internet +
Video,

# cargar la última configuración activa (antes de que se apagara la red o se cambiara al
TGMS),

# volver al menú principal o finalizar el programa.

while true_opcion == 0:

    print("\n¿Qué desea configurar en dicha ONU? Escoja la opción:")

    print("0 - Volver al menú principal")

    print("1 - Servicio de Internet")

    print("2 - Servicio de Internet + Vídeo")

    print("3 - Cargar la última configuración activa")

    print("4 - Finalizar el programa")

    opcion = input()

    opcion = int(opcion)

    if opcion == 0 or opcion == 1 or opcion == 2 or opcion == 3 or opcion == 4:

```

```
    true_opcion = 1
else:
    # Si el usuario introduce un nº de opción no válido, se le vuelve a preguntar.
    print("Opción no válida. Vuelva a probar.")

# Si introduce un 0, el programa vuelve al menú principal donde se pide
# la dirección MAC de la ONU sobre la que se quiere trabajar.
if opcion == 0:
    print("\nHa seleccionado volver al menú principal.\n")
# Si el usuario introduce un 1, se va a configurar servicio de Internet.
elif opcion == 1:
    print("\nConfiguración del servicio de Internet:")
    cont=0
    true_Internet = 0
    # Se le pide el nº de servicios. Este nº podría ser mayor que 2 pero realmente,
    # no tiene sentido configurar más de 2 servicios puesto que solo hay 2 VLAN
    # que proporcionen servicio a esta red.
    while true_Internet == 0:
        if cont == 0:
            print("Número de servicios de datos:")
        else:
            print("Número de servicios no válido. Vuelva a probar:")
        cont = cont+1
        num_servicios_Internet = input()
        num_servicios_Internet = int(num_servicios_Internet)
        if num_servicios_Internet > -1 and num_servicios_Internet < 3:
            true_Internet = 1

    print("\n")
# En caso de que el número de servicios sea mayor que 0, se llamará a la función
```

```

# servicio_Internet.

if num_servicios_Internet > 0:
    servicio_Internet(ID_ONU,direccion_MAC,num_servicios_Internet)
# Tanto si se han configurado 1 o más servicios de Internet, como si no se ha
# configurado ninguno, el programa pregunta al usuario si desea seguir configurando la red.
true_respuesta=0
while true_respuesta == 0:
    print("¿Desea continuar configurando la red? Responda SI o NO")
    respuesta = input()
    if respuesta == "NO":
        print("La configuración de la red GPON ha finalizado.")
        true_respuesta = 1
        true_FIN = 1
    elif respuesta == "SI":
        print("\n")
        true_respuesta = 1
    else:
        print("Vuelva a introducir su respuesta (SI o NO)")

# Si el usuario introduce un 2, se va a configurar servicio de Internet+Video.
elif opcion == 2:
    print("\nConfiguración del servicio de Vídeo:")
    cont=0
    true_Video = 0
    # Se le pide el nº de servicios. Este nº podría ser mayor que 2 pero realmente,
    # no tiene sentido configurar más de 2 servicios puesto que solo hay 2 VLAN
    # que proporcionen servicio a esta red.
    while true_Video == 0:
        if cont == 0:
            print("Número de servicios de vídeo:")
        else:

```

```
    print("Número de servicios no válido. Vuelva a probar:")
    cont = cont+1
    num_servicios_Video = input()
    num_servicios_Video = int(num_servicios_Video)
    if num_servicios_Video > -1 and num_servicios_Video < 3:
        true_Video = 1

print("\n")
# En caso de que el número de servicios sea mayor que 0, se llamará a la función
# servicio_Video.
if num_servicios_Video > 0:
    servicio_Video(ID_ONU,direccion_MAC,num_servicios_Video)
# Tanto si se han configurado 1 o más servicios de Internet+Video, como si no se ha
# configurado ninguno, el programa pregunta al usuario si desea seguir configurando la red.
true_respuesta=0
while true_respuesta == 0:
    print("¿Desea continuar configurando la red? Responda SI o NO")
    respuesta = input()
    if respuesta == "NO":
        print("La configuración de la red GPON ha finalizado.")
        true_respuesta = 1
        true_FIN = 1
    elif respuesta == "SI":
        print("\n")
        true_respuesta = 1
    else:
        print("Vuelva a introducir su respuesta (SI o NO)")

# Si el usuario introduce un 3, se va a cargar la última configuración activa (si
# es que había alguna) antes de apagar la red o cambiar al modo de gestión TGMS
```

```
elif opcion == 3:
    cargar_configuracion(ID_ONU,direccion_MAC)
    # Una vez ejecutada la función, el programa pregunta al usuario si desea seguir
    configurando la red.
    true_respuesta=0
    while true_respuesta == 0:
        print("¿Desea continuar configurando la red? Responda SI o NO")
        respuesta = input()
        if respuesta == "NO":
            print("La configuración de la red GPON ha finalizado.")
            true_respuesta = 1
            true_FIN = 1
        elif respuesta == "SI":
            print("\n")
            true_respuesta = 1
        else:
            print("Vuelva a introducir su respuesta (SI o NO)")

# Si el usuario introduce un 4, finaliza el programa.
elif opcion == 4:
    print("La configuración de la red GPON ha finalizado.")
    true_FIN = 1
```

## Script *acceso\_CLI.py*

```
# Este script sirve para acceder directamente al menú con privilegios del CLI
# sin necesidad introducir las claves de acceso cada vez que se entre al CLI.
# Se ha de importar el módulo telnetlib para la conexión Telnet
import telnetlib

# Se define la dirección IP del host y el puerto de la conexión
host = "172.26.128.38"
port = "4551"

# Se definen las claves de acceso
password1 = "TLNT25"
password2 = "TLNT145"

# Se define el comando enable que permitirá entrar en el menú con privilegios
enable = "enable"

# Ahora se realiza la conexión y se introducen los comandos. Si hubiera un error,
# saltaría la excepción y se mostraría un mensaje de error.
try:
    # Conexión Telnet al host y puerto anteriormente definidos. Si pasaran más
    # de 5 segundos sin establecer conexión, aparecería un mensaje de error.
    tn = telnetlib.Telnet(host,port,5)
    # Se escriben los comandos
    tn.write(password1.encode('ascii') + b"\n")
    tn.write(enable.encode('ascii') + b"\n")
    tn.write(password2.encode('ascii') + b"\n")
    # La función interact() es la que permite interactuar con la terminal del CLI
    tn.interact()

except:
    print("Error")
```

## Script *activar\_CLI.py*

```
# Este script sirve para activar el CLI como modo de gestión de la red GPON
# Se ha de importar el módulo telnetlib para la conexión Telnet y el módulo time
import telnetlib
import time

# Se define la dirección IP del host y el puerto de la conexión
host = "172.26.128.38"
port = "4555"

# Se define el usuario y el password
user = "root"
password = "root"

# Ahora se realiza la conexión y se introducen los comandos. Si hubiera un error,
# saltaría la excepción y se mostraría un mensaje de error.
try:
    # Conexión Telnet al host y puerto anteriormente definidos. Si pasaran más
    # de 5 segundos sin establecer conexión, aparecería un mensaje de error.
    tn = telnetlib.Telnet(host,port,5)

    # Se introducen el usuario y el password
    tn.write(user.encode('ascii') + b"\r")
    tn.write(password.encode('ascii') + b"\r")

    # Ahora se introducen los comandos necesarios para la activación del CLI
    tn.write("system".encode('ascii') + b"\r")
    tn.write("setProvisionEngineMode 1".encode('ascii') + b"\r")
    tn.write("applyProvisionEngineMode".encode('ascii') + b"\r")

    # Se para durante dos segundos el programa ya que el comando anterior
    # tiene un tiempo de ejecución mayor que otros
    time.sleep(2)
```

```
# Con la función interact() se puede interactuar con la conexión Telnet y  
# comprobar, si se quiere, que se ha activado el CLI. Esta línea es opcional.  
tn.interact()
```

except:

```
print("Error")
```

## Script *activar\_TGMS.py*

```
# Este script sirve para activar el TGMS como modo de gestión de la red GPON
# Se ha de importar el módulo telnetlib para la conexión Telnet y el módulo time
import telnetlib
import time

# Se define la dirección IP del host y el puerto de la conexión
host = "172.26.128.38"
port = "4555"

# Se define el usuario y el password
user = "root"
password = "root"

# Ahora se realiza la conexión y se introducen los comandos. Si hubiera un error,
# saltaría la excepción y se mostraría un mensaje de error.
try:
    # Conexión Telnet al host y puerto anteriormente definidos. Si pasaran más
    # de 5 segundos sin establecer conexión, aparecería un mensaje de error.
    tn = telnetlib.Telnet(host,port,5)

    # Se introducen el usuario y el password
    tn.write(user.encode('ascii') + b"\r")
    tn.write(password.encode('ascii') + b"\r")

    # Ahora se introducen los comandos necesarios para la activación del TGMS
    tn.write("system".encode('ascii') + b"\r")
    tn.write("setProvisionEngineMode 0".encode('ascii') + b"\r")
    tn.write("applyProvisionEngineMode".encode('ascii') + b"\r")

    # Se para durante dos segundos el programa ya que el comando anterior
```

```
# tiene un tiempo de ejecución mayor que otros
```

```
time.sleep(2)
```

```
# Con la función interact() se puede interactuar con la conexión Telnet y
```

```
# comprobar, si se quiere, que se ha activado el TGMS. Esta línea es opcional.
```

```
tn.interact()
```

```
except:
```

```
    print("Error")
```

## Script *mostrar\_modos\_funcionamiento.py*

```
# Este script sirve para activar el TGMS como modo de gestión de la red GPON
# Se ha de importar el módulo telnetlib para la conexión Telnet y el módulo time
import telnetlib

# Se define la dirección IP del host y el puerto de la conexión
host = "172.26.128.38"

port = "4555"

# Se define el usuario y el password
user = "root"

password = "root"

# Ahora se realiza la conexión y se introducen los comandos. Si hubiera un error,
# saltaría la excepción y se mostraría un mensaje de error.
try:

    # Conexión Telnet al host y puerto anteriormente definidos. Si pasaran más
    # de 5 segundos sin establecer conexión, aparecería un mensaje de error.
    tn = telnetlib.Telnet(host,port,5)

    # Se introducen el usuario y el password
    tn.write(user.encode('ascii') + b"\r")

    tn.write(password.encode('ascii') + b"\r")

    # Ahora se introducen los comandos necesarios para mostrar el modo de
    # funcionamiento activo en la red
    tn.write("system".encode('ascii') + b"\r")

    tn.write("showProvisionEngineMode".encode('ascii') + b"\r")

    # La función interact() es la que permite interactuar con la conexión Telnet
    # y ver el modo de gestión activo.
    tn.interact()

except:

    print("Error")
```

## **Script *mostrar\_ONUs\_conectadas.py***

```
# Este script sirve para ver las ONUs conectadas a la red y su información principal
# Se ha de importar el módulo telnetlib para la conexión Telnet
import telnetlib

# Se define la dirección IP del host y el puerto de la conexión
host = "172.26.128.38"
port = "4551"

# Se definen las claves de acceso
password1 = "TLNT25"
password2 = "TLNT145"

# Se define el comando enable que permitirá entrar en el menú con privilegios
enable = "enable"

# Se la serie de comandos que permiten ver las ONUs conectadas
comandos = "configure \n olt-device 0 \n olt-channel 0 \n show serial-number allocated"

# Ahora se realiza la conexión y se introducen los comandos. Si hubiera un error,
# saltaría la excepción y se mostraría un mensaje de error.
try:
    # Conexión Telnet al host y puerto anteriormente definidos. Si pasaran más
    # de 5 segundos sin establecer conexión, aparecería un mensaje de error.
    tn = telnetlib.Telnet(host,port,5)
    # Se introducen las claves de acceso y los comandos
    tn.write(password1.encode('ascii') + b"\n")
    tn.write(enable.encode('ascii') + b"\n")
```

```
tn.write(password2.encode('ascii') + b"\n")

tn.write(comandos.encode('ascii') + b"\n")

# La función interact() es la que permite interactuar con la terminal del CLI
# y ver las ONUs conectadas y la información asociada a ellas.

tn.interact()

except:

    print("Error")
```