



UNIVERSIDAD DE VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE  
TELECOMUNICACIÓN

**Desarrollo de algoritmos de asignación de  
recursos en redes de acceso PON y TWDM-PON**

Autor:

**D. David Juárez Estévez**

Tutores:

**Dña. Noemí Merayo Álvarez**

**D. Ramón J. Durán Barroso**

---

**TÍTULO: Desarrollo de algoritmos de asignación de recursos en redes de acceso PON y TWDM-PON**

**AUTOR: D. David Juárez Estévez**

**TUTORES: Dña. Noemí Merayo Álvarez, D. Ramón J. Durán Barroso**

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

---

**TRIBUNAL**

---

**PRESIDENTE: D. Ignacio de Miguel Jiménez**

**SECRETARIO: D. Ramón J. Durán Barroso**

**VOCAL: Dña. Noemí Merayo Álvarez**

**SUPLENTE 1: D. Rubén M. Lorenzo Toledo**

**SUPLENTE 2: D. Juan Carlos Aguado Manzano**

---

---

**FECHA: 19 de Mayo de 2017**

**CALIFICACIÓN:**

---

## **Resumen de TFG**

El estudio de investigación realizado y descrito en este Trabajo Fin de Grado (TFG) se ha centrado en el desarrollo de algoritmos de gestión de recursos en redes PON (*Passive Optical Network*) y TWDM-PON (*Time Wavelength Division Multiple Passive Optical Network*), cuya arquitectura está implementada en un simulador bajo el entorno de simulación OMNET++.

Así pues, inicialmente, se realizó un análisis de las redes de acceso PON, además del entorno de trabajo sobre el cual se iba a realizar este proyecto y la arquitectura de la red PON sobre la cual iban a ir implementados los algoritmos diseñados.

A continuación, se llevó a cabo un estudio del estado del arte relacionado con la implementación de redes neuronales para la gestión del proceso de sintonización de un conjunto de controladores PID previamente implementados en el simulador de redes PON disponible en el grupo de trabajo. Posteriormente, se implementó dicho proceso de sintonización de un controlador PID incorporando el diseño de una red neuronal y las entradas de “*bias*” en las que se centra nuestro estudio, para que de forma automática y periódica se modifiquen en tiempo real los parámetros de sintonización del controlador PID en función del estado actual de la red, del tráfico y de los requisitos de calidad de servicio de los diferentes abonados.

Finalmente, se llevó a cabo un estudio del estado del arte relacionado con el estándar NGPON2 (*Next Generation Passive Optical Network 2*) y se desarrolló el algoritmo TWDM que se basa en él, realizando varias simulaciones para comprobar el impacto del tiempo de tuning que se incluye al trabajar con láseres sintonizables tanto en el receptor como en el transmisor.

## **Palabras clave**

PON (Red Óptica Pasiva); EPON (Red Óptica Pasiva Ethernet); NGPON2 (Red Óptica Pasiva de Próxima Generación); QoS (Calidad de Servicio); PID (Proporcional-Integral-Derivativo); SLA (Acuerdos de nivel de servicio); Red Neuronal

## **Agradecimientos**

Me gustaría agradecer a mis padres, Adoración y José Luis, todo el apoyo que siempre me han dado, tanto a lo largo de la carrera como en la realización de este trabajo, invitándome siempre a seguir hacia adelante y a no rendirme.

Y, por supuesto, me gustaría destacar el agradecimiento a mi tutora de este trabajo, Noemí. Gracias por su paciencia, dedicación y colaboración constante, sin la cual la consecución de este trabajo hubiera sido mucho más complicada.

# ÍNDICE GENERAL

<b>1 Introducción.....</b>	<b>1</b>
1.1 Motivación.....	1
1.2 Objetivos.....	2
1.2.1 Objetivo General.....	2
1.2.2 Objetivos Específicos.....	2
1.3 Fases y Métodos.....	3
1.3.1 Fase de Investigación.....	3
1.3.2 Fase de Diseño.....	3
1.3.3 Fase de Implementación.....	4
1.3.4 Fase de Simulación.....	4
1.3.5 Fase de Realización de los Informes.....	5
1.4 Estructura de la Memoria del TFG.....	5
<b>2 Redes de Acceso Ópticas Pasivas (PON).....</b>	<b>6</b>
2.1 Introducción.....	6
2.2 Definición de Redes de Acceso.....	6
2.3 Redes de Acceso Ópticas Pasivas (PON, <i>Passive Optical Network</i> ).....	7
2.4 Estándares en Redes de Acceso PON.....	7
2.5 Control de Acceso al Medio en Redes PON.....	8
2.6 Conclusiones.....	9
<b>3 Entorno de Trabajo y Arquitectura de la Red Simulada.....</b>	<b>10</b>
3.1 Introducción.....	10
3.2 Simulador de redes OMNeT++.....	10
3.3 Generador de tráfico.....	11
3.4 Arquitectura de la red.....	11
3.5 Conclusiones.....	13

<b>4 Controladores PID basados en Redes Neuronales con auto-aprendizaje.....</b>	<b>14</b>
4.1 Introducción.....	14
4.2 Modelo y Topología de las Redes Neuronales.....	15
4.3 Controladores PID en redes PON.....	17
4.3.1 Introducción a los controladores PID.....	17
4.3.2 Algoritmo de asignación de recursos SPID ( <i>Service Level Agreement PID</i> ) basado en controladores PID.....	18
4.3.3 Diseño de una red neuronal basada en auto-aprendizaje para sintonizar el PID.....	21
4.4 Resultados del estudio de simulación.....	27
4.4.1 Escenario de simulación genérico.....	27
4.4.2 Análisis del impacto de las señales de “ <i>bias</i> ” en la red neuronal.....	28
4.4.3 Análisis del impacto de diferentes factores de normalización.....	33
4.4.4 Análisis de los errores obtenidos en la Red Neuronal.....	37
4.4.5 Análisis del impacto de los valores de inercia en la red neuronal.....	38
4.4.6 Impacto de los parámetros $n$ y $m$ en el comportamiento de la red neuronal.....	40
4.5 Conclusiones.....	46
 <b>5 Implementación, simulación y validación de un algoritmo DBA TWDM.....</b>	 <b>47</b>
5.1 Introducción.....	47
5.2 Estado del arte y arquitecturas del estándar NGPON2.....	47
5.3 Implementación del algoritmo TWDM.....	50
5.3.1 Asignación de Ancho de banda.....	50
5.3.2 Asignación de longitudes de onda en cada ciclo.....	51
5.4 Resultados.....	53
5.4.1 Escenario de simulación genérico.....	53
5.4.2 Impacto del tiempo de <i>tuning</i> para cargas altas.....	53
5.4.3 Impacto del tiempo de <i>tuning</i> para cargas bajas.....	58
5.5 Conclusiones.....	63

<b>6 Conclusiones y Líneas Futuras.....</b>	<b>64</b>
6.1 Conclusiones.....	64
6.2 Líneas Futuras.....	65
 <b>7 Bibliografía.....</b>	 <b>67</b>

## ÍNDICE DE FIGURAS

Figura 1. Arquitectura de una pila TWDM-PON.....	8
Figura 2. Generación de tráfico Self-Similar.....	11
Figura 3. Arquitectura de la red EPON simulada en OMNeT++.....	12
Figura 4. Módulo OLT.....	12
Figura 5. Módulo ONU.....	12
Figura 6. Modelo de neurona artificial.....	15
Figura 7. Topología de la red neuronal.....	16
Figura 8. Diagrama de bloques de un proceso controlado por un PID.....	17
Figura 9. Diagrama de bloques del proceso PID diseñado para controlar los requisitos de ancho de banda de QoS en PONs.....	19
Figura 10. Diagrama de flujo de la Red Neuronal diseñada para controlar los requisitos de ancho de banda de QoS.....	22
Figura 11. Ancho de banda garantizado para el $SLA_0$ .....	28
Figura 12. Ancho de banda garantizado para el $SLA_1$ .....	28
Figura 13. Ancho de banda garantizado para el $SLA_2$ .....	29
Figura 14. Ancho de banda garantizado para el $SLA_0$ .....	30
Figura 15. Ancho de banda garantizado para el $SLA_1$ .....	30
Figura 16. Ancho de banda garantizado para el $SLA_2$ .....	31
Figura 17. Parámetro de sintonización $K_p$ para unos anchos de banda garantizados de 100, 75 y 50 Mbps.....	31
Figura 18. Parámetro de sintonización $T_i$ para unos anchos de banda garantizados de 100, 75 y 50 Mbps.....	32
Figura 19. Parámetro de sintonización $T_d$ para unos anchos de banda garantizados de 100, 75 y 50 Mbps.....	32
Figura 20. Ancho de banda garantizado para distintas normalizaciones en el $SLA_0$ .....	33
Figura 21. Ancho de banda garantizado para distintas normalizaciones en el $SLA_1$ .....	34
Figura 22. Ancho de banda garantizado para distintas normalizaciones en el $SLA_2$ .....	34
Figura 23. Parámetros de sintonización para una normalización de $10^6$ .....	35
Figura 24. Parámetros de sintonización para una normalización de $10^9$ .....	35
Figura 25. Parámetros de sintonización para una normalización de $10^{15}$ .....	36
Figura 26. Error para unas normalizaciones de $10^6$ , $10^9$ y $10^{15}$ .....	37

Figura 27. Ancho de banda garantizado para el $SLA_1$ para diferentes valores de $\alpha$ con $\eta=0.1$ .....	39
Figura 28. Ancho de banda garantizado para el $SLA_1$ para diferentes valores de $\eta$ con $\alpha=0.1$ .....	40
Figura 29. Evolución del parámetro $K_p$ con $m = 8$ y diferentes valores de $n$ (2, 4 y 8).....	41
Figura 30. Evolución del parámetro $T_i$ con $m = 8$ y diferentes valores de $n$ (2, 4 y 8).....	41
Figura 31. Evolución del parámetro $T_d$ con $m = 8$ y diferentes valores de $n$ (2, 4 y 8).....	42
Figura 32. Ancho de banda garantizado para el $SLA_2$ con $m = 8$ y diferentes valores de $n$ (2, 4 y 8).....	42
Figura 33. Evolución del parámetro $K_p$ con $n = 4$ y diferentes valores de $m$ (8, 16 y 32).....	43
Figura 34. Evolución del parámetro $T_i$ con $n = 4$ y diferentes valores de $m$ (8, 16 y 32).....	44
Figura 35. Evolución del parámetro $T_d$ con $n = 4$ y diferentes valores de $m$ (8, 16 y 32).....	44
Figura 36. Ancho de banda garantizado para el $SLA_2$ con $n = 4$ y diferentes valores de $m$ (8, 16 y 32).....	45
Figura 37. Arquitectura TWDM-PON.....	48
Figura 38. Representación del funcionamiento del algoritmo para la asignación de las longitudes de onda.....	51
Figura 39. Representación de una situación de colisión en la asignación de longitudes de onda en cada ciclo entre varias estaciones.....	52
Figura 40. Evolución del ancho de banda total aprovechado para $t_{cycle} = 2\text{ ms}$ y diferentes valores de $t_{tuning}$ ( $t_{tuning} = 0\text{ }\mu\text{s}$ , $10\text{ }\mu\text{s}$ , $50\text{ }\mu\text{s}$ y $300\text{ }\mu\text{s}$ ).....	54
Figura 41. Evolución del ancho de banda total aprovechado para $t_{cycle} = 6.7\text{ ms}$ y diferentes valores de $t_{tuning}$ ( $t_{tuning} = 0\text{ }\mu\text{s}$ , $10\text{ }\mu\text{s}$ , $50\text{ }\mu\text{s}$ y $300\text{ }\mu\text{s}$ ).....	54
Figura 42. Evolución del ancho de banda total aprovechado para $t_{cycle} = 10\text{ ms}$ y diferentes valores de $t_{tuning}$ ( $t_{tuning} = 0\text{ }\mu\text{s}$ , $10\text{ }\mu\text{s}$ , $50\text{ }\mu\text{s}$ y $300\text{ }\mu\text{s}$ ).....	55
Figura 43. Evolución del ancho de banda total aprovechado para $t_{tuning} = 0\text{ }\mu\text{s}$ , $10\text{ }\mu\text{s}$ o $50\text{ }\mu\text{s}$ y diferentes valores de $t_{cycle}$ ( $t_{cycle} = 2\text{ ms}$ , $6.7\text{ ms}$ y $10\text{ ms}$ ).....	56
Figura 44. Evolución del ancho de banda total aprovechado para $t_{tuning} = 300\text{ }\mu\text{s}$ y diferentes valores de $t_{cycle}$ ( $t_{cycle} = 2\text{ ms}$ , $6.7\text{ ms}$ y $10\text{ ms}$ ).....	57
Figura 45. Evolución del ancho de banda aprovechado por ONU para cargas altas cuando $t_{cycle} = 2\text{ ms}$ y $t_{tuning} = 300\text{ }\mu\text{s}$ y $50\text{ }\mu\text{s}$ .....	58
Figura 46. Evolución del ancho de banda total aprovechado para $t_{cycle} = 2\text{ ms}$ y diferentes valores de $t_{tuning}$ ( $t_{tuning} = 0\text{ }\mu\text{s}$ , $10\text{ }\mu\text{s}$ , $50\text{ }\mu\text{s}$ y $300\text{ }\mu\text{s}$ ).....	59
Figura 47. Evolución del ancho de banda total aprovechado para $t_{cycle} = 6.7\text{ ms}$ y diferentes valores de $t_{tuning}$ ( $t_{tuning} = 0\text{ }\mu\text{s}$ , $10\text{ }\mu\text{s}$ , $50\text{ }\mu\text{s}$ y $300\text{ }\mu\text{s}$ ).....	59
Figura 48. Evolución del ancho de banda total aprovechado para $t_{cycle} = 10\text{ ms}$ y diferentes valores de $t_{tuning}$ ( $t_{tuning} = 0\text{ }\mu\text{s}$ , $10\text{ }\mu\text{s}$ , $50\text{ }\mu\text{s}$ y $300\text{ }\mu\text{s}$ ).....	60



Figura 49. Evolución del ancho de banda total aprovechado para $t_{\text{tuning}} = 0 \mu s$ , $10 \mu s$ o $50 \mu s$ y diferentes valores de $t_{\text{cycle}}$ ( $t_{\text{cycle}} = 2 ms$ , $6.7 ms$ y $10 ms$ ).....	61
Figura 50. Evolución del ancho de banda total aprovechado para $t_{\text{tuning}} = 300 \mu s$ y diferentes valores de $t_{\text{cycle}}$ ( $t_{\text{cycle}} = 2 ms$ , $6.7 ms$ y $10 ms$ ).....	61
Figura 51. Evolución del ancho de banda aprovechado por ONU para cargas bajas cuando $t_{\text{cycle}} = 2 ms$ y $t_{\text{tuning}} = 300 \mu s$ y $50 \mu s$ .....	62

## ÍNDICE DE TABLAS

Tabla 1. Parámetros de la simulación considerados en el escenario de simulación para PID.....	27
Tabla 2. Parámetros de la simulación considerados en el escenario de simulación para TWDM.....	53
Tabla 3. Resultados para cargas altas cuando se varía el $t_{\text{tuning}}$ manteniendo constante el $t_{\text{cycle}}$ .....	55
Tabla 4. Resultados para cargas altas cuando se varía el $t_{\text{cycle}}$ manteniendo constante el $t_{\text{tuning}}$ .....	57
Tabla 5. Resultados para cargas bajas cuando se varía el $t_{\text{tuning}}$ manteniendo constante el $t_{\text{cycle}}$ .....	60
Tabla 6. Resultados para cargas bajas cuando se varía el $t_{\text{cycle}}$ manteniendo constante el $t_{\text{tuning}}$ .....	62

## ÍNDICE DE ECUACIONES

Ecuación 1. Función de activación lineal.....	15
Ecuación 2. Función de activación tangente hiperbólica.....	15
Ecuación 3. Ecuación que modela el comportamiento de un controlador PID en el dominio del tiempo.....	17
Ecuación 4. Ecuación que modela el comportamiento de un controlador PID en el dominio discreto.....	18
Ecuación 5. Ecuación para la asignación de los anchos de banda iniciales.....	19
Ecuación 6. Relación que modela el comportamiento de un controlador PID en el tiempo discreto.....	22
Ecuación 7. Relación de las constantes del PID.....	22
Ecuación 8. Relación entre las entradas y las salidas para la capa de entrada.....	24
Ecuación 9. Relación entre las entradas y las salidas para la capa oculta.....	24
Ecuación 10. Salida de la capa oculta utilizando como función de activación la tangente hiperbólica.....	24
Ecuación 11. Relación entre las entradas y las salidas para la capa de salida.....	25
Ecuación 12. Salida de la capa de salida utilizando como función de activación la lineal.....	25
Ecuación 13. Error cuadrático que se utiliza para actualizar los pesos.....	25
Ecuación 14. Cálculo de los incrementos de la matriz de pesos entre las capas oculta y de salida.....	25
Ecuación 15. Ecuación que modela la variación del error cuadrático en función de los pesos entre las capas oculta y de salida.....	25
Ecuación 16. Relación entre $\Delta u(k)$ y las salidas de la Red Neuronal.....	26
Ecuación 17. Simplificación para el cálculo de los incrementos de la matriz de pesos entre las capas oculta y de salida y definición de las deltas obtenidas para la capa de salida.....	26
Ecuación 18. Simplificación para el cálculo de los incrementos de la matriz de pesos entre las capas de entrada y oculta y definición de las deltas obtenidas para la capa oculta.....	26
Ecuación 19. Ancho de banda máximo calculado en el algoritmo TWDM según el SLA asociado a la ONU i.....	51

# 1

## Introducción

### 1.1 Motivación

En este Trabajo Fin de Grado se han llevado a cabo diferentes estrategias y algoritmos para el control de recursos en Redes de Acceso Ópticas Pasivas (PON, *Passive Optical Networks*), partiendo de un simulador ya implementado en el entorno de simulación OMNeT++, basado en el estándar Ethernet (EPON) y desarrollado previamente por el Grupo de Comunicaciones Ópticas (GCO).

La investigación que ha tenido lugar en los últimos años en el campo de las redes PON [1] se ha centrado en el desarrollo de esquemas que gestionen el acceso al medio compartido y que no tengan problemas al distribuir el ancho de banda disponible en el canal de un modo eficiente y equitativo entre los diferentes usuarios que necesiten de este. Al mismo tiempo, la calidad de servicio, también llamada QoS (*Quality of Service*), es otro de los puntos fuertes en los que se centra gran parte de la investigación en nuestros días. La implementación de la red del proyecto, se centra en crear una Red Neuronal basada en el estándar Ethernet (EPON) y una red PON basada en el estándar NGPON2.

Lo que se quiere conseguir con este Trabajo Fin de Grado es mejorar los métodos ya existentes y proponer algunos nuevos para optimizar la gestión de recursos y la calidad de servicio ofrecida en las redes de acceso PON de nuestros días.

Así, el eje principal de este proyecto será el desarrollo de una Red Neuronal en la que mediante auto-aprendizaje se irán obteniendo los parámetros de sintonización de la misma, y además se estudiará el ancho de banda con la finalidad de garantizar unos niveles mínimos del mismo a diferentes perfiles de abonado.

Por otro lado, también se analizará el comportamiento de un algoritmo híbrido TDM-WDM adecuado a redes NGPON2, esto es, TWDM (*Time Wavelength Division Multiple*), donde el proceso de asignación de recursos se realiza mediante técnicas de

multiplexación por división de longitud de onda y multiplexación por división de tiempo. Estas redes NGPON2 proporcionan una velocidad de datos mayor y tienen la capacidad de ahorrar una cantidad de energía significativa, además de acumular 4 XG-PON (10 Gbps PON) utilizando WDM, y aumentar por lo tanto la capacidad del canal de subida y de bajada.

Con estas redes se alcanzan los 40 Gbps, es decir, una velocidad 4 veces mayor de la que se conseguía con las tecnologías anteriores. Cada longitud de onda es capaz de proporcionar a un abonado con acceso óptico hasta una velocidad de 10 Gbit/s, y la longitud de onda de subida y bajada también es capaz de funcionar a la tasa más baja de 2,5 Gbit/s. La distancia típica entre el terminal de línea óptica (OLT) y una unidad de red óptica (ONU) es de 40 km, y el OLT es capaz de soportar hasta 256 unidades ONU [2].

## **1.2 Objetivos**

### **1.2.1 Objetivo general**

El objetivo principal de este proyecto es la gestión de la calidad de servicio en redes PON a través de una Red Neuronal. En concreto, se pretenden implementar nuevas estrategias de sintonización para que de forma automática, transparente y adaptativa se modifiquen de forma dinámica los parámetros de sintonización de la red acorde al estado actual de la misma y al tráfico que hay en ella.

Así mismo, otro objetivo principal de este proyecto se ha centrado en el desarrollo de otras estrategias de control de recursos en redes TWDM-PON basadas en estrategias híbridas TDM-WDM, con el objetivo de aumentar la eficiencia de las redes PON.

Estos objetivos generales se pueden desglosar en otros más específicos que se indican a continuación.

### **1.2.2 Objetivos Específicos**

En la realización de este proyecto, se han cubierto los siguientes objetivos específicos:

1. Análisis e implementación de nuevas estrategias para mejorar el comportamiento de las redes PON mediante el uso de una Red Neuronal. Esta red se encargará de hacer evolucionar dinámicamente mediante auto-aprendizaje los parámetros de sintonización que utilizan los controladores PID, de un modo automático y adaptativo de acuerdo a las condiciones actuales de la red y del tráfico.
2. Implementar en la red PON el algoritmo híbrido TDM-WDM bajo el control de acceso al medio WDMA (*Wavelength Division Multiplexing*

*Access*) para el canal de bajada y TDMA (*Time Division Multiplexing Access*) para el de subida. Mediante WDMA, se realiza la asignación de longitud de onda de manera que la primera que quede libre, será la primera en asignarse a la siguiente ONU. Mientras que TDMA se utiliza para evitar la colisión entre tramas, y se basa en la utilización de una única longitud de onda en el canal de acceso compartido, asignando un cierto intervalo de tiempo a cada usuario a lo largo de un periodo.

## **1.3 Fases y Métodos**

La metodología a seguir para el desarrollo de los objetivos del Trabajo Fin de Grado ha constado fundamentalmente de las siguientes fases:

### **1.3.1 Fase de Investigación**

En esta fase se ha reunido toda la información referente al simulador de Redes de Acceso Ópticas Pasivas (PON) con el que se ha realizado el proyecto.

En primer lugar, se buscó analizar y comprender tanto la arquitectura como el funcionamiento de estas redes PON, además de la versión 4.6 del entorno de simulación OMNeT++ [3] [4], en la que estaba previamente desarrollado dicho simulador. También se realizó una búsqueda sobre la implementación de Redes Neuronales para la gestión del proceso de sintonización de controladores PID.

Por último, se llevó a cabo un análisis de las redes de acceso ópticas PON basadas en el estándar NGPON2.

### **1.3.2 Fase de Diseño**

En esta fase del proyecto se partirá de la arquitectura de red EPON ya implementada en el simulador, y se diseñarán y adoptarán nuevas estrategias para la mejora de los resultados de dichos algoritmos en la gestión de recursos de la red PON.

Por tanto, se diseñará una estrategia basada en Redes Neuronales para llevar a cabo el proceso de sintonización del PID de forma automática y dinámica con la finalidad de controlar diferentes parámetros de las redes PON, tales como el ancho de banda asignado, los parámetros de sintonización del PID, los errores, etc.

Además, se diseñará un algoritmo TWDM híbrido bajo los protocolos de control de acceso al medio WDMA y TDMA.

### 1.3.3 Fase de Implementación

En esta fase se implementarán los dos algoritmos en los que vamos a basar nuestro análisis. En concreto, se llevarán a cabo las siguientes tareas:

- Diseño e implementación de una Red Neuronal para obtener de forma adaptativa y automática los valores de los parámetros de sintonización del controlador PID. Estos parámetros irán evolucionando a lo largo de la simulación adaptando estos valores a las nuevas condiciones de la red y del tráfico y consiguiendo así un mejor comportamiento de los controladores del PID que gestionan los recursos en estas redes.
- Diseño e implementación del algoritmo híbrido TWDM basado en WDMA y TDMA para hacer una asignación de ancho de banda eficiente.

Recogida de los parámetros obtenidos durante la simulación de ambos algoritmos, anchos de banda, errores, etc.

### 1.3.4 Fase de Implementación

En esta fase del Trabajo Fin de Grado se procedió a la realización de las simulaciones de los dos algoritmos implementados con los parámetros definidos teóricamente para el análisis de los resultados obtenidos. En concreto, se han seguido los siguientes pasos:

- Simulación de la estrategia de obtención de los parámetros del controlador PID mediante el uso de una Red Neuronal, variando la carga de la red:
  - Recogida de los resultados de simulación relacionados con el control del ancho de banda de cada perfil de abonado, junto con los parámetros obtenidos de la Red Neuronal diseñada y el controlador PID, tales como la evolución en tiempo real de los parámetros de sintonización y los errores obtenidos en el sistema con los mismos.
  - Realización de las gráficas y estudio de los resultados obtenidos.
- Simulación del algoritmo TWDM para hacer que la asignación de ancho de banda sea eficiente para distintas cargas en la red:
  - Recogida de los resultados de simulación relacionados con el control del ancho de banda de cada perfil de abonado.
  - Realización de las gráficas y estudio de los resultados obtenidos.

### **1.3.5 Fase de Realización de los Informes**

En esta fase se procedió a realizar los informes del Trabajo Fin de Grado:

- Documentación de todos los archivos que forman la red.
- Realización de la memoria del Trabajo Fin de Grado.

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

En el Capítulo 2 se realiza un breve estudio de las Redes de Acceso Ópticas Pasivas, al haber sido ya desarrolladas en Trabajos Fin de Grado previos, y se presenta el estándar NGPON2.

El Capítulo 3 se centra en el entorno de trabajo empleado para la realización e implementación de este proyecto. De este modo, se presenta una breve introducción al entorno de trabajo OMNeT++ y a la arquitectura de la red de acceso PON implementada en el simulador OMNeT++.

En el Capítulo 4 se presenta la estrategia basada en una Red Neuronal para la obtención de los parámetros de los controladores PID en redes PON. Estos parámetros se obtienen de un modo dinámico y adaptativo para las diferentes condiciones de red y tráfico. En este capítulo se explicará detalladamente la definición de la Red Neuronal, cómo funciona el sistema de evolución de dicho proceso de sintonización y, por último, la implementación de dicha Red Neuronal con las novedades que hayamos realizado. Finalmente, se hará el análisis correspondiente de los diferentes parámetros extraídos en el algoritmo y los resultados de QoS de la red PON cuando se integra la Red Neuronal con un controlador PID para garantizar unos mínimos niveles de ancho de banda a diferentes perfiles de abonado.

En el Capítulo 5 se procederá al estudio de la implementación, simulación y validación del algoritmo TWDM desarrollado en este proyecto. También se presentará la descripción detallada de los métodos implementados, así como las simulaciones y resultados obtenidos, comparando de esta manera los resultados logrados con los teóricos.

En el Capítulo 6 se recogen las conclusiones derivadas de todo el trabajo realizado en este Trabajo 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 la presente memoria.



# 2

## Redes de Acceso Ópticas Pasivas (PON)

### 2.1 Introducción

En este capítulo se describirá de forma breve el funcionamiento de las Redes de Acceso Ópticas Pasivas (PON) y se verán las características generales de las tecnologías de redes de acceso, centrándonos en la red con la que se trabajará para la realización del presente trabajo, esto es, la red de acceso óptico pasiva.

El estudio se centrará más en profundidad en realizar una breve reseña de los estándares adoptados, así como de los mecanismos de control de acceso al medio implementados, en concreto, en el acceso múltiple por división en tiempo (TDMA), en el acceso múltiple por división en longitud de onda (WDMA), así como en el acceso al medio con división múltiple en tiempo y longitud de onda (TWDM), aplicado a este tipo de redes.

### 2.2 Definición de Redes de Acceso

La red de acceso se puede definir como el conjunto de elementos que permiten conectar a cada abonado con la central local de la que es dependiente. Esta red se compone de una oficina central (CO, *Central Office*) y en el otro extremo de la red se encuentra el usuario final conectado.

El tipo de enlace que une ambos extremos de la red dependerá de la tecnología desplegada, aunque debido al gran aumento experimentado en el tráfico que circula por las redes existentes, se emplea la fibra óptica.

Por lo tanto, la red de acceso se basa actualmente en la tecnología FTTx (*Fiber to the x*), donde éste es un término con el que designamos cualquier acceso de banda ancha sobre fibra óptica que sustituye parcial o totalmente el cobre del cable de acceso. De todas las

infraestructuras FTTx existentes, nuestro estudio se centra en FTTH o *Fibra Hasta el Hogar*, la cual propone la utilización de fibra óptica hasta el domicilio del usuario y se basa en la utilización de cables de fibra óptica y sistemas de distribución ópticos adaptados a esta tecnología para la distribución de servicios avanzados tales como telefonía, internet de banda ancha y televisión a los hogares y negocios de los abonados.

### **2.3 Redes de Acceso Ópticas Pasivas (PON, *Passive Optical Network*)**

Este tipo de redes está constituido únicamente por elementos pasivos entre la fuente y el destino. En ellas se produce una comunicación bidireccional que tiene lugar entre el OLT (*Optical Line Terminal*) o Unidad de Línea Óptica, localizada dentro de la oficina central, un divisor óptico pasivo (Splitter) y varias ONUs (*Optical Network Units*) o Unidades de Red Óptica, localizadas dentro o cerca de las dependencias del abonado final [5] [6]. El OLT se conecta punto-a-punto con el Splitter, y éste último se conecta punto-a-multipunto con las ONUs, de manera que divide la fibra troncal en varias ramas de distribución que conectan a los usuarios finales con la oficina central.

En cuanto a las topologías, la que predomina es la topología en árbol, en la cual, todos los usuarios comparten el ancho de banda en el canal ascendente, mientras que en el sentido descendente cada usuario tiene el total de la capacidad de su canal individual durante un intervalo de tiempo. Así pues, la red simulada en OMNeT++ para este Trabajo Fin de Grado también sigue una topología en árbol.

### **2.4 Estándares en Redes de Acceso PON**

Respecto a los estándares disponibles para las redes de acceso PON, se ha elegido trabajar con EPON para desarrollar la Red Neuronal. Este estándar está basado en el protocolo Ethernet y se encuentra definido dentro de la especificación IEEE.802.3ah [7], aprobada por la *IEEE Standard Association* (IEEE-SA) en junio de 2004. Fue definido para soportar tasas de transmisión por encima de 1 Gbit/s, siendo esta la empleada en el desarrollo de nuestra red neuronal.

Para realizar la asignación y gestión de ranuras temporales para la transmisión de información en la red, la especificación del estándar EPON define el llamado protocolo de control multipunto MPCP (*Multi-Point Control Protocol*) que actúa dentro de la subcapa de control MAC y arbitra la comunicación entre los canales de subida y de bajada.

En cuanto a la implementación del algoritmo TWDM, se ha elegido trabajar con el estándar NGPON2, ya que es una de las alternativas más prometedoras a nivel de despliegue y en la que se está investigando en la actualidad.

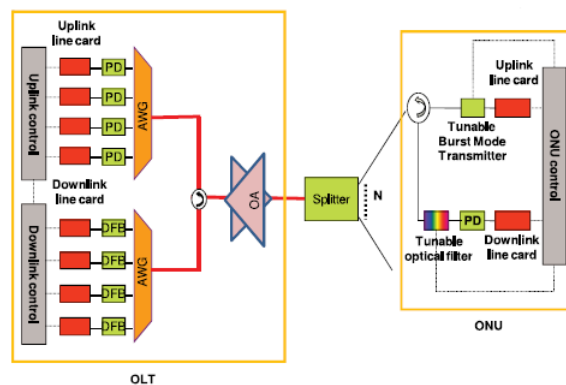
Dicho estándar se definió en 2015 y fue desarrollado por la UIT (*Unión Internacional de Telecomunicaciones*). Se basa en una arquitectura de múltiples longitudes de onda, punto a multipunto y estas redes NGPON2 proporcionan una velocidad de datos mayor y tienen la capacidad de ahorrar una cantidad de energía significativa, además de acumular 4 XG-PON (10 Gbps PON) utilizando WDM, y aumentar la capacidad de subida y de bajada. En concreto, es capaz de proporcionar una capacidad de 40 Gbps, tener un alcance de 40 km y soportar como mínimo 64 unidades ONU por puerto, aunque también puede tener interés para 256 unidades ONU o más.

Las longitudes de onda utilizadas por este estándar incluyen desde 1596 nm a 1603 nm en la dirección de bajada y desde 1524 nm a 1544 nm en la dirección de subida.

## 2.5 Control de Acceso al Medio en Redes PON

Respecto al protocolo de control de acceso al medio empleado en nuestra red EPON para el desarrollo de la Red Neuronal, decir que se utiliza TDMA (*Time Division Multiplexing Access, Acceso Múltiple por División en el Tiempo*). Este protocolo será necesario en el canal ascendente para que diversos usuarios no colisionen al acceder al tramo de red común y se basa en la utilización de varias longitudes de onda en el canal de acceso compartido, asignando un cierto intervalo de tiempo a cada usuario a lo largo de un periodo. Trabajando sobre una sola fibra, se utilizan longitudes de onda diferentes mediante técnicas WDM.

En cuanto a nuestra red NGPON2, se utiliza como protocolo de control de acceso al medio TWDM (*Time Wavelength Division Multiple, División Múltiple en Tiempo y Longitud de Onda*). La arquitectura de un pila TWDM PON se puede ver como ilustra la Figura 1, donde las abreviaturas utilizadas significan: DFB = Láser de retroalimentación distribuida, PDF = fotodiodos, AWG = Grating rayado de guía de ondas, OA = Amplificadores ópticos:



**Figura 1:** Arquitectura de una pila TWDM-PON

WDM se proporciona en la dirección de bajada mediante la combinación de luz de un láser sintonizable en el OLT con multiplexación por longitud de onda. La luz se filtra a continuación en cada ONU con un filtro sintonizable de forma activa que deja pasar

solamente la longitud de onda de bajada deseada a su receptor. En la dirección de subida, el láser sintonizable en cada ONU asigna dinámicamente una longitud de onda. Las fibras de todas las ONU se combinan con un multiplexor o Splitter. TDM se proporciona en la dirección de subida a través de la utilización de láseres de ráfaga en cada ONU.

Finalmente, se utilizan algoritmos de asignación dinámica de ancho de banda (DBA, *Dynamic Bandwidth Allocation*) para adaptar la capacidad de la red a las condiciones de tráfico existentes en todo momento, modificando la distribución del ancho de banda asignado a cada ONU en función de la demanda actual y de los requisitos de Calidad de Servicio (QoS, *Quality of Service*) estipulados por los proveedores de servicio. Con este mecanismo se consigue no desaprovechar la capacidad del canal y será el OLT el encargado de controlar esta asignación del ancho de banda ciclo tras ciclo en función de los requisitos de demanda del ancho de banda actual de las ONUs.

## 2.6 Conclusiones

Como las tecnologías implementadas en el tramo de la red de acceso en los últimos años no cubrían toda la demanda de ancho de banda exigido por los abonados para la transmisión de información, se propuso el despliegue de tecnologías FTTx basadas en fibra óptica. En todas las variantes existentes, la fibra hasta el hogar (FTTH) se perfila como la candidata más idónea para el despliegue de fibra en el acceso.

Para el despliegue de dicha tecnología, la topología más típica para su implementación es la de árbol, debido a que todos los usuarios comparten el ancho de banda. Esta arquitectura presenta una configuración punto-multipunto en el canal de bajada, y punto-a-punto en canal de subida. El problema del canal ascendente radica en que al tener una configuración punto-a-punto, necesita de protocolos de contienda para controlar el acceso al medio en el tramo de la red compartida. En nuestro caso, utilizamos TDMA y WDMA.

Por otro lado, en cuanto a los estándares utilizados, se ha elegido el estándar Ethernet (EPON) para el desarrollo de la Red Neuronal. Este estándar es una tecnología barata, mundialmente aceptada, desplegada en el 90% de las redes mundiales y es idóneo para el transporte de tráfico de información en paquetes Ethernet, al igual que tráfico de voz y de video. Sin embargo, para el desarrollo del algoritmo TWDM, se ha elegido el estándar NGPON2, debido a que es un estándar muy prometedor y que está en auge ahora mismo.

# 3

## Entorno de Trabajo y Arquitectura de la Red Simulada

### 3.1 Introducción

En este capítulo se hará una descripción del entorno de trabajo OMNeT++ utilizado para desarrollar nuestro simulador de Redes de Acceso Ópticas Pasivas y de la arquitectura de la red simulada mediante dicho simulador. Como estos temas ya han sido tratados en profundidad en varios Trabajos Fin de Grado anteriores, se expondrán de una manera más breve.

### 3.2 Simulador de redes OMNeT++

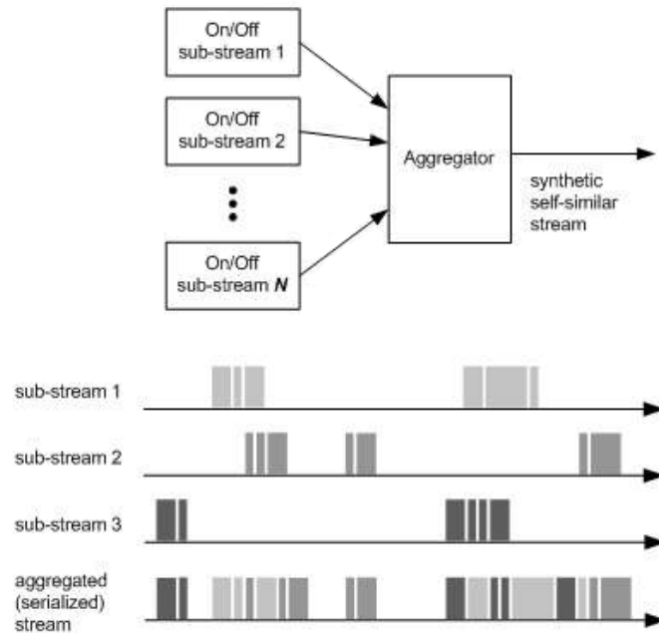
Este entorno de trabajo OMNeT++ es un simulador de redes de eventos discretos, que está basado en el lenguaje de programación C++. Se basa en módulos orientados a objetos, y es de código abierto [8]. Al ser un simulador de eventos discretos, los objetos cambian de estado únicamente en instantes discretos de tiempo, es decir, nada sucede entre dos eventos consecutivos, los cuales tardan un tiempo nulo en ser realizados.

A la hora de realizar una aplicación para simular una red, OMNeT++ permite una jerarquía de modelos, sin un límite de niveles de jerarquía. Para ello la simulación se compone de módulos simples, desarrollados en C++ que pueden ser integrados en módulos complejos, que a su vez componen la red.

La comunicación entre módulos se consigue mediante el intercambio de mensajes o paquetes. Los mensajes pueden representar tramas o paquetes de una red real, pudiendo incluir diferentes estructuras de datos. Asimismo, el intercambio de mensajes puede realizarse de dos formas, bien mediante un camino predefinido que seguirá el mensaje desde su creación en el módulo origen hasta su llegada al módulo destino o bien a través de puertas y conexiones directas entre dos módulos.

### 3.3 Generador de tráfico

En cuanto a la generación del tráfico, se implementarán diferentes patrones del mismo, pero el más significativo es el tráfico Self-Similar [9]. Esta fuente genera de manera rafagosa paquetes de diversos tamaños definidos en la configuración de la red para posteriormente insertarlos en las colas hasta que dichas estaciones transmitan, tal y como se ilustra en la Figura 2:



**Figura 2:** Generación de tráfico Self-Similar

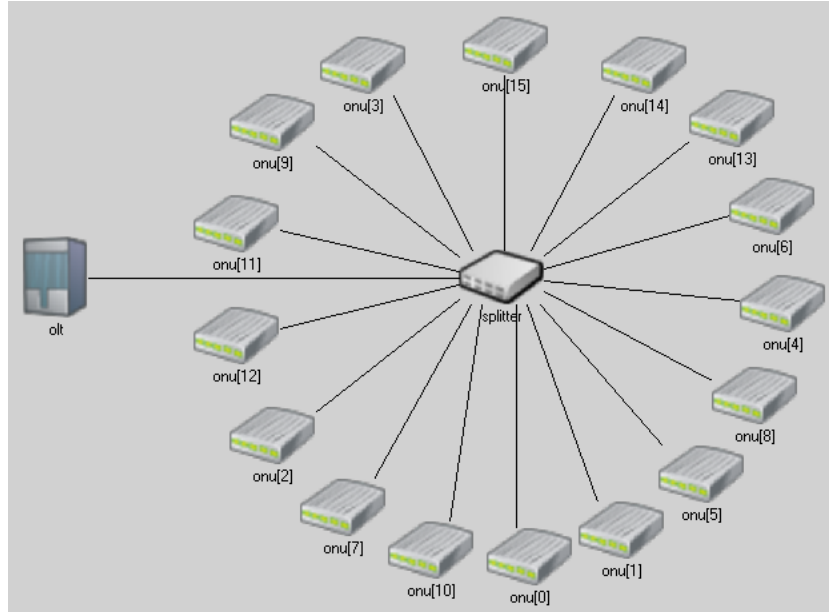
En concreto, en nuestro Trabajo Fin de Grado se trabajará con la segunda versión de las vistas en la Figura 2 (sub-stream 2) [10], con la salvedad de que se generan paquetes de diferentes tamaños, no todos de un tamaño constante como ahí se ilustraba. Así, se han implementado dos fuentes de tráfico Self-Similar, como las indicadas anteriormente, las cuales poseen una carga determinada que iremos variando para las diferentes simulaciones que realicemos. Pero además de estas dos fuentes, poseemos una fuente de tipo CBR (*Constant Bit Rate*), que simulará un tráfico a velocidad constante, que se sumará a las cargas de las fuentes de tráfico Self-Similar, y cuyo valor fijo es de 0,0448. Entonces como fuentes de tráfico tendremos las dos fuentes Self-Similar y esta fuente de carga fija, que sumadas en conjunto darán la carga total de la red.

### 3.4 Arquitectura de la red

La red simulada tiene una topología en árbol y está basada en una infraestructura FTTH (*Fiber To The Home, Fibra hasta el hogar*). El acceso mediante redes PON se centra en una comunicación bidireccional entre una Unidad de Línea Óptica (OLT, *Optical Line Terminal*) localizada dentro de la oficina central y varias Unidades de Red Óptica

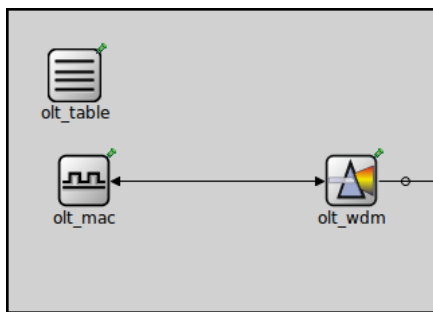
(ONUs, *Optical Network Units*) localizadas dentro o cerca de las dependencias del abonado final.

En concreto, la arquitectura con la que se va a trabajar se ilustra en la Figura 3:

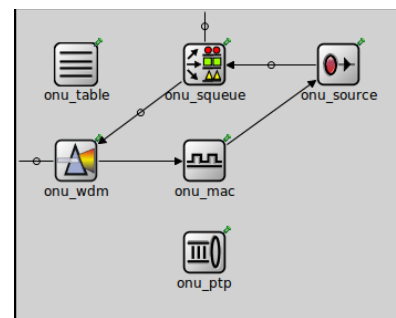


**Figura 3:** Arquitectura de la red EPON simulada en OMNeT++

En cuanto a los dos módulos más importantes de esta arquitectura, es decir, el OLT (*Optical Line Terminal*) y la ONU (*Optical Network Unit*) que se conectan entre sí mediante un divisor óptico o Splitter, se puede ver su arquitectura en la Figuras 4 y 5, respectivamente:



**Figura 4:** Módulo OLT



**Figura 5:** Módulo ONU

Donde el OLT es la parte principal de la Red de Acceso, debido a que lleva a cabo toda la inteligencia y gestión de la misma, mientras que la ONU es el extremo final de cada rama de la topología en árbol de la subred, y se localiza en las dependencias del usuario final.

Por otro lado, el Splitter es el divisor óptico pasivo encargado de enviar todo lo que le llega por sus entradas por todas sus salidas sin encaminar los paquetes. En concreto, los mensajes que le llegan desde el OLT se duplican y se envían hacia todas las ONUs,

mientras que cuando le llegan paquetes desde cada ONU, funciona como un combinador y se envían todos esos paquetes hacia el OLT en el orden que le llegaron.

### **3.5 Conclusiones**

El entorno de trabajo OMNeT++ es un simulador de redes de eventos discretos, basado en módulos orientados a objetos, programados en el lenguaje C++. Se ha elegido este entorno de trabajo debido a que es un entorno multiplataforma de libre distribución para propósitos académicos de investigación, además de que su programación por módulos favorece la simulación de procesos en paralelo y distribuidos.

Por otro lado, en cuanto a la generación de tráfico, se utiliza un patrón de tráfico realista, denominado auto-semejante o Self-Similar, y se observa el comportamiento de los diversos algoritmos implementados en función de diversos parámetros de configuración.

Relacionado con la arquitectura de la red, se han definido el OLT, el Splitter y las ONUs. Además, cada módulo compuesto OLT y ONU se compone de submódulos que conforman la organización interna de cada uno de éstos. Dentro de estos submódulos, las capas MAC del OLT y de las ONUs son las más importantes, las cuales se consideran el cerebro de cada módulo. Estos elementos se disponen siguiendo una topología en árbol.



# 4

## Controladores PID basados en Redes Neuronales con auto- aprendizaje

### 4.1 Introducción

En este capítulo, se va a utilizar uno de los algoritmos que han sido desarrollados en Trabajos Fin de Grado previos. En concreto, un algoritmo de sintonización de un controlador PID (*Proportional-Integral-Derivative*) encargado de realizar la gestión de recursos y garantizar diferentes niveles de ancho de banda a usuarios con distintos perfiles y servicios asociados.

Este método de sintonización estará basado en una Red Neuronal de auto-aprendizaje, por lo que no se realizará una sintonización previa al funcionamiento del controlador. Por contra, los parámetros de sintonización del controlador PID se van modificando en tiempo real para adaptarse a las diferentes condiciones de la red, consiguiendo una sintonización lo más óptima posible del PID en cada momento.

Por tanto, en la primera parte del capítulo se describirá brevemente cuál es el principio de funcionamiento de los controladores PID y cómo se realiza la asignación de ancho de banda en una red EPON utilizando este tipo de controladores. En concreto partimos del algoritmo SPID (*Service level agreement PID*) basado en este tipo de controladores. Posteriormente, se estudiarán técnicas de sintonización de controladores PID basados en redes neuronales, trabajando sobre el diseño de dicha red neuronal, viendo cómo estaba implementada la que ya se tenía en el grupo de trabajo e indicando los cambios que se han ido realizando en ella. Finalmente, se presentarán los resultados obtenidos de las simulaciones realizadas en distintos escenarios para el algoritmo SPID.

## 4.2 Modelo y topología de las Redes Neuronales

La neurona artificial, que es la unidad de proceso de información fundamental, posee un estado interno, llamado nivel de activación, y recibe señales que le permiten cambiar de estado. Para ello, las neuronas artificiales poseen una función, llamada función de activación, que les permite cambiar de nivel a partir de las señales que reciben.

En la Figura 6 se muestra un modelo de neurona artificial, donde las entradas se corresponden con las señales de la sinapsis de una neurona biológica, y se le ha añadido una señal de “bias” ( $X_0 = 1$ ), que va a ser el objeto de nuestro estudio en esta parte del trabajo.

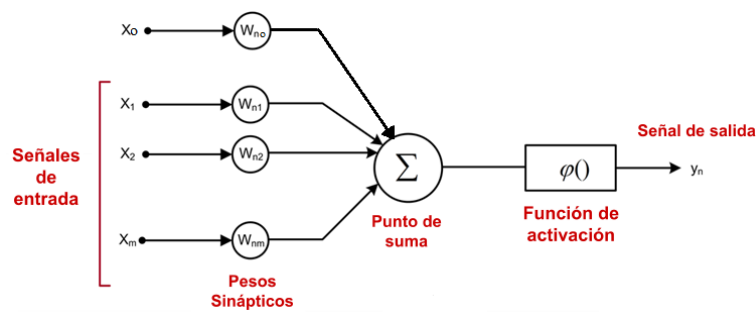


Figura 6: Modelo de neurona artificial

La entrada total a la neurona se calcula como la suma de todas las entradas ponderadas por ciertos valores, denominados pesos, a los que multiplica cada señal antes de ser aplicado el sumatorio.

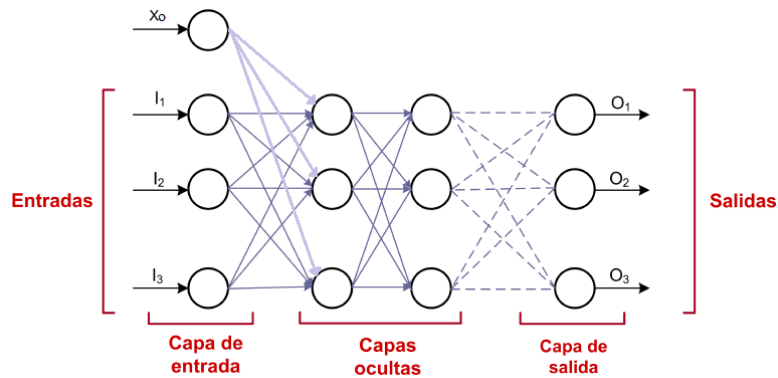
Mediante el ajuste de los pesos sinápticos, la red es capaz de adaptarse a cualquier entorno y realizar una determinada tarea. Después, se suman todas las entradas ponderadas algebraicamente y posteriormente la señal es procesada por una función llamada función de activación, produciendo la salida de la neurona. Una vez calculada la salida de una neurona, ésta se propaga a las neuronas destino con las que esta neurona está conectada [11] [12].

Un parámetro importante que se debe determinar en el proceso de diseño de una red neuronal artificial es la función de activación. Las funciones de activación definen la salida de la neurona en función del potencial de activación, y las más utilizadas son la función lineal, cuya expresión se muestra en la Ecuación 1, y la función tangente hiperbólica, que se define como se observa en la Ecuación 2:

$$g(x) = x \quad (1)$$

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2)$$

En cuanto a la topología, viene dada por el número de capas, la cantidad de neuronas por capa, el grado de conectividad y el tipo de conexión entre neuronas. Una red neuronal artificial está compuesta por una o más capas, las cuales se encuentran interconectadas entre sí. Cuando la red está compuesta por dos o más capas, ésta se denomina multicapa. La estructura básica de interconexión entre neuronas de la red multicapa está representada en la Figura 7, donde también se le ha añadido la señal de “bias” con la que vamos a trabajar.



**Figura 7:** Topología de la red neuronal

Como se puede observar, el primer nivel lo constituyen las neuronas de la capa de entrada, las cuales reciben la información procedente del exterior. A continuación puede haber una o varias capas ocultas. Y finalmente, el último nivel es la capa de salida. La salida de estas unidades sirve como salida de toda la red.

A través de cada interconexión entre neuronas viajan valores numéricos de una neurona a otra, los cuales son evaluados por los pesos de las conexiones. Dichos pesos se ajustan durante el proceso de aprendizaje para producir la red neuronal artificial final.

Las redes neuronales (NN) tradicionales necesitan de buenos ejemplos para ser entrenadas, por lo que el proceso de aprendizaje podría ser bastante largo, debido a que su rendimiento depende en gran medida de la cantidad de datos con los que son entrenados [13]. Por el contrario, nuestra propuesta se centra en el desarrollo de un auto-aprendizaje y la adaptación de redes neuronales que permiten un ajuste en línea y en tiempo real de los parámetros de ajuste del PID de acuerdo con el reciente estado de funcionamiento de la red. De hecho, hay muchos algoritmos en la literatura que implementan redes neuronales auto-adaptativas para diseñar controladores PID inteligentes [14-17]. En particular, proponemos desarrollar un controlador PID experto e inteligente basado en una red neuronal con el fin de proporcionar automáticamente niveles mínimos de ancho de banda a diferentes perfiles.

El funcionamiento de una red neuronal se basa en que cada vector de entrada es introducido en la red copiando cada valor de dicho vector en la neurona de entrada correspondiente. Cada neurona de la red, una vez recibida la totalidad de sus entradas,

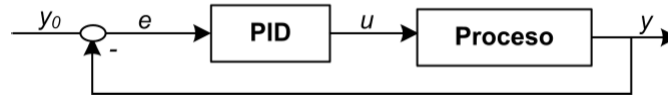
las procesa y genera una salida que es propagada a través de las conexiones, llegando como entrada a la neurona destino. Una vez que la entrada ha sido completamente propagada por toda la red, se producirá un vector de salida, cuyos componentes son cada uno de los valores de salida de las neuronas de salida.

En cuanto a la señal de “bias”, comentar que, como ya se sabe, una neurona se activa o dispara si su entrada total supera un cierto umbral. Ahora bien, muchas veces es interesante modificar este umbral, haciendo más difícil que la neurona se dispare (subiendo el umbral) o más fácil (bajando el mismo). Para hacer esto, se añade lo que se llama “neurona de inclinación” o señal de “bias”,  $X_0$ , a la que se le asigna un valor fijo de 1, y un peso sináptico asociado.

### 4.3 Controladores PID en redes PON

#### 4.3.1 Introducción a los controladores PID

Un controlador PID (*Proporcional-Integral-Derivativo*, *Proportional-Integral-Derivative*) es un controlador simple basado en un lazo de realimentación, que tiene el objetivo de mantener la variable que se controla lo más cerca posible del valor deseado [18]. El diagrama de bloques típico para un PID se muestra en la Figura 8, donde se puede observar que consta de dos bloques principales, el proceso bajo control y el controlador PID.



**Figura 8:** Diagrama de bloques de un proceso controlado por un PID

El proceso tiene una entrada, que es la variable de control denotada por  $u$  en la Figura 8, y una salida ( $y$ ) que es la variable que se quiere mantener lo más cerca posible del valor deseado ( $y_0$ ). Por lo tanto, el PID compara el valor de salida del proceso  $y$  con el valor de referencia  $y_0$ , y la diferencia entre estos dos valores, es el error cometido y se denota por  $e$ . Entonces, el PID ajusta la variable de control  $u$  en función del error cometido para mantener el valor de la variable de salida ( $y$ ) lo más cerca posible del valor de referencia ( $y_0$ ), para minimizar el error cometido.

La ecuación que modela el comportamiento de un controlador PID en el dominio del tiempo es la que se muestra en la Ecuación 3, donde  $u(t)$  es la variable de control en el dominio del tiempo y  $e(t)$  es el error, definido como  $e(t) = y_0 - y(t)$ :

$$u(t) = K_p \cdot e(t) + K_p \cdot \frac{1}{T_i} \int_0^t e(\tau) \cdot d\tau + K_p \cdot T_d \cdot \frac{de(t)}{dt} \quad (3)$$

En esta ecuación se observa que la señal de control  $u(t)$  es la suma de tres términos: el término P (proporcional al error), el término I (proporcional a la integral del error) y el término D (proporcional a la derivada del error).

Sin embargo, para aplicar el control de un PID en una simulación de eventos discretos, como es el caso del sistema que nos atañe, es necesario aplicar un controlador PID discreto. Este tipo de PID lee el error y lo calcula en un determinado intervalo de tiempo, llamado tiempo de muestreo. Así, una aproximación de la Ecuación 3 en el dominio discreto se muestra en la Ecuación 4 [19] [20], donde T se refiere a dicho tiempo de muestreo:

$$u[n] = K_p \cdot e[n] + K_p \cdot \frac{T}{T_i} \sum_{m=0}^n e[m] + K_p \cdot \frac{T_d}{T} \cdot (e[n] - e[n-1]) \quad (4)$$

En esta ecuación en el dominio discreto la componente proporcional (P) se ajusta al error cometido y se encarga del error presente. La componente integral (I) es proporcional a la integral del error y tiene como función principal asegurarse de que la salida del proceso se mantenga igual al valor de referencia en un estado estacionario, y está relacionada con la acumulación de errores pasados. Mientras que la componente derivada (D) es proporcional a la derivada del error, proporciona estabilidad al sistema y será la encargada de realizar una predicción de los errores futuros. Además, la variable  $K_p$  es la ganancia proporcional,  $T_i$  es la ganancia integral y  $T_d$  es la ganancia derivada

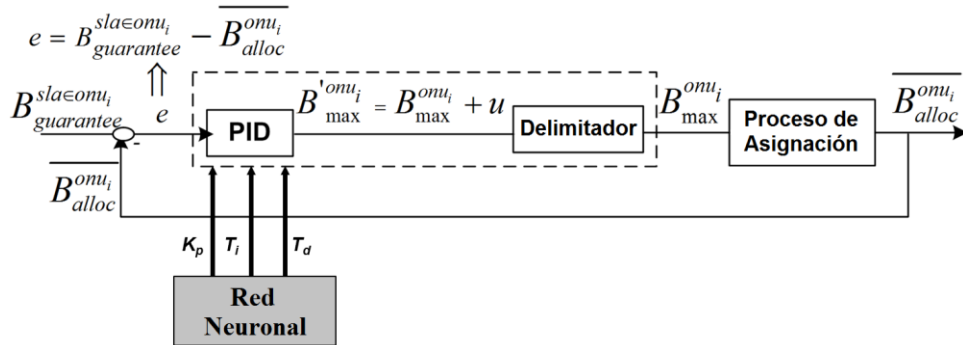
### 4.3.2 Algoritmo de asignación de recursos SPID (*Service Level Agreement PID*) basado en controladores PID

Este algoritmo SPID realiza un control automático de la asignación de ancho de banda mediante el uso de un controlador PID, con el fin de mantener el ancho de banda asignado a cada usuario lo más cerca posible de su ancho de banda garantizado, de modo que los requisitos de calidad de su SLA contratado puedan cumplirse.

Para asignar el ancho de banda a cada  $ONU_i$  en cada ciclo ( $B_{alloc}^{onu_i}$ ), el algoritmo implementa una política de *polling* con un esquema limitado, que es la forma más común y eficiente de asignar el ancho de banda en las redes PON [21], y con la ventaja de que su diseño puede modificarse fácilmente para controlar otros parámetros esenciales en la red.

Este esquema limitado y simple se define como  $B_{alloc}^{onu_i} = \text{Mínimo} \{B_{required}^{onu_i}, B_{max}^{onu_i}\}$ , donde  $B_{required}^{onu_i}$  es el ancho de banda requerido por la  $ONU_i$  en un ciclo y  $B_{max}^{onu_i}$  es el ancho de banda máximo permitido a cada ONU en un ciclo (para evitar la monopolización del canal de algunas ONUs). Con el fin de integrar el controlador en el proceso de asignación de ancho de banda ( $B_{alloc}^{onu_i}$ ) el término  $B_{max}^{onu_i}$  se actualiza constantemente por medio de la señal de control ( $u[n]$ ) así que  $B_{max}^{onu_i}$  puede evolucionar dinámicamente para

asegurar eficientemente los niveles mínimos de ancho de banda a cada perfil. El diagrama de bloques del algoritmo se muestra en la Figura 9. Como el algoritmo PID controla el ancho de banda garantizado asociado a cada ONU, el valor de referencia es el ancho de banda mínimo garantizado que debe ser asegurado por el proveedor de servicio dependiendo del SLA contratado ( $B_{\text{guaranteed}}^{\text{sla} \in \text{onu}_i}$ ). El término bajo control es el ancho de banda medio asignado asociado con cada ONU<sub>i</sub> ( $\overline{B_{\text{alloc}}^{\text{onu}_i}}[n]$ ), y para calcular el error instantáneo ( $e[n]$ ) cometido en cada ONU, es necesario sustraer el ancho de banda medio asignado de su nivel de ancho de banda garantizado ( $e[n] = B_{\text{guaranteed}}^{\text{sla} \in \text{onu}_i} - \overline{B_{\text{alloc}}^{\text{onu}_i}}$ ). Por lo tanto, para actualizar el ancho de banda máximo permitido ( $B_{\text{max}}^{\text{onu}_i}$ ), la señal de control  $u[n]$  se añade al ancho de banda máximo permitido anterior ( $B_{\text{alloc}}^{\text{onu}_i} = B_{\text{max}}^{\text{onu}_i} + u[n]$ ), como se puede observar en la Figura 9. De esta manera, si por ejemplo el ancho de banda medio asignado a la ONU<sub>i</sub> es menor que su nivel de ancho de banda garantizado, el error se vuelve positivo y la señal de control también, por lo que el algoritmo incrementa su ancho de banda máximo permitido para consentir a la ONU<sub>i</sub> cumplir con las restricciones de QoS para el ancho de banda. Finalmente, se incluye un delimitador en el sistema para adaptar los máximos proporcionalmente a los calculados por el controlador, para ajustar el tiempo máximo de ciclo del estándar EPON (2 ms) (Figura 9).



**Figura 9:** Diagrama de bloques del proceso PID diseñado para controlar los requisitos de ancho de banda de QoS en PONs

También hay que apuntar que aunque el PID adapta constantemente los valores del ancho de banda máximo, éstos se inician con un peso asociado al SLA contratado ( $W^{\text{sla}_j}$ ), por lo que las condiciones iniciales de ancho de banda se calculan en base a la Ecuación 5, donde  $B_{\text{cycle available}}$  es el ancho de banda disponible en cada ciclo máximo,  $N_{\text{onus}}^{\text{sla}_j}$  representa el número de ONUs asociadas al SLA  $j$  en la red y el término  $W^{\text{sla}_k / \text{onu}_i \in \text{sla}_k}$  se refiere al peso del SLA  $k$  contratado por la ONU  $i$ .

$$B_{\text{max}}^{\text{onu}_i} = \frac{B_{\text{cycle available}} \cdot W^{\text{sla}_k / \text{onu}_i \in \text{sla}_k}}{\sum_j W^{\text{sla}_j} \cdot N_{\text{onus}}^{\text{sla}_j}} \quad (5)$$

Por último, hay que indicar que para un correcto funcionamiento de un PID, es necesario determinar los parámetros que ajustan la señal de control, esto es, la ganancia

proporcional  $K_p$ , el tiempo integral  $T_i$  y el tiempo derivativo  $T_d$ . A este proceso se le denomina sintonización del PID y es una etapa muy importante, ya que es necesario determinar un valor óptimo para estos parámetros en función de la aplicación, con el fin de que el sistema sea estable y se alcancen los objetivos para los cuales ha sido diseñado el controlador.

Para llevar a cabo este proceso, se utilizan métodos tales como el de respuesta de frecuencia propuesto por Ziegler-Nichols [18] [19]. Sin embargo, ésta es una técnica manual que podía ser muy lenta y laboriosa si los valores seleccionados están lejos de los adecuados, aunque su principal problema es que una vez hallados los parámetros del controlador, son constantes, por lo que no se adaptan a posibles cambios en el estado del proceso. Así pues, su aplicación en redes de acceso PON podría llegar a no ser adecuada ya que el perfil del tráfico demandado por un usuario cambia a lo largo del día o a lo largo de ciertas épocas del año.

Otro tipo de métodos utilizados se basan en la integración de sistemas más automatizados, como por ejemplo la incorporación de algoritmos genéticos para automatizar el proceso de ajuste. Los algoritmos genéticos, al resolver problemas de optimización de forma adecuada, han sido utilizados en diversas ocasiones para la sintonización de controladores PID. En concreto, en la literatura es posible encontrar numerosos ejemplos de sintonización mediante algoritmos genéticos de controladores PID utilizados en diversos campos y aplicaciones. Por ejemplo, en [22–25] los autores aplican satisfactoriamente algoritmos genéticos en la sintonización de PIDs en plantas de distinto orden, obteniendo en todos los casos mejores respuestas que con el método de sintonización de Ziegler-Nichols. En [26] se utiliza un algoritmo genético para sintonizar un controlador PID aplicado a un proceso no lineal y se presenta un ejemplo de su utilización en un proceso de neutralización de la acidez en disoluciones o pH. De forma similar, los autores en [27] implementan un algoritmo genético para el control de los niveles de agua en tanques. Por último, en el campo de las telecomunicaciones, también se ha utilizado la sintonización de PIDs mediante algoritmos genéticos en el control de la congestión mediante el manejo activo de colas (AQM) en redes TCP/IP con el objetivo de reducir la pérdida de paquetes y mejorar la utilización de la red [28]. Aunque automatiza de manera eficiente el proceso de sintonización y estos parámetros de sintonización ofrecen un buen rendimiento en el proceso de asignación de ancho de banda, esta técnica no permite una adaptación en tiempo real de los parámetros de sintonización en caso de que la red o las condiciones del tráfico cambien repentinamente. Por lo tanto, los algoritmos genéticos se consideran una técnica de sintonización sin conexión, por lo que habría que lanzarlo de nuevo para encontrar los parámetros de sintonización más acordes a la situación de red actual.

Otro método que se utiliza, son las redes neuronales tradicionales, que aprenden de ejemplos en un proceso que tiene claramente dos fases: la fase de aprendizaje y la fase de ejecución. Durante la primera fase, la fase de aprendizaje, la red es entrenada para realizar un determinado tipo de procesamiento. Una vez alcanzado un nivel de



entrenamiento adecuado, se pasa a la fase de ejecución, donde la red es utilizada para llevar a cabo la tarea para la cual fue entrenada. El aprendizaje es la parte más importante de una red de neuronas artificiales y consiste en la determinación de los valores precisos de los pesos para todas sus conexiones, de tal forma que la red neuronal sea capaz de resolver un problema de manera eficiente. Una vez finalizada la fase de aprendizaje, la red puede ser utilizada para realizar la tarea para la que fue entrenada. Ejemplos de este tipo de métodos se pueden ver en la bibliografía consultada en [29-31]. Una de las principales ventajas que presenta es que la red aprende la relación existente entre los datos, adquiriendo la capacidad de generalizar conceptos. Se trata de que, tras un periodo de entrenamiento determinado, la red esté lista para funcionar correctamente y calcular la salida para cualquier caso que se presente.

Finalmente, por todo lo comentado anteriormente, es bastante deseable diseñar un método que actualice los parámetros de sintonización ante cualquier cambio dentro de la red, es decir, una técnica de ajuste en línea o en tiempo real. De esta forma, se propone la integración de una red neuronal auto-adaptativa dentro del controlador PID que aprende automáticamente de los últimos errores cometidos dependiendo de las condiciones de la red en tiempo real, gracias a lo que se modifican los pesos de la matriz de la red neuronal, y por lo tanto, también los valores de los parámetros del PID, a diferencia de lo que ocurría con los métodos anteriores que aprendían de ejemplos y donde no se modificaban los pesos de la matriz. Modelos de este método basado en el auto-aprendizaje se pueden consultar en la bibliografía en [32-35].

### **4.3.3 Diseño de una red neuronal basada en auto-aprendizaje para sintonizar el PID**

En esta parte se presenta el diseño e implementación de una red neuronal basada en auto-aprendizaje para sintonizar el controlador PID. Este tipo de técnica, no necesita una sintonización previa al funcionamiento del controlador, esto es, calcular los parámetros de sintonización antes de que el PID funcione. Por el contrario, estos parámetros de sintonización se van actualizando en tiempo real para adaptarse a las diferentes condiciones de la red, consiguiendo una sintonización óptima del controlador PID en cada momento.

En este sentido, en la literatura existen diversos ejemplos de controladores PID sintonizados mediante redes neuronales en diferentes campos de aplicación. En todos ellos, el PID con una sintonización adaptativa basada en redes neuronales ofrece un mejor comportamiento que un PID tradicional, debido a que los sistemas que controlan son no lineales o variantes en el tiempo, o simplemente porque ofrecen una mayor robustez a cambios en el entorno. Estos sistemas se basan en el auto-aprendizaje en base al estado actual del sistema, gracias a lo que se modifican los pesos de la matriz de la red neuronal, y por lo tanto, también los valores de los parámetros del PID. Esto se puede ver por ejemplo en los trabajos descritos en [32-35].



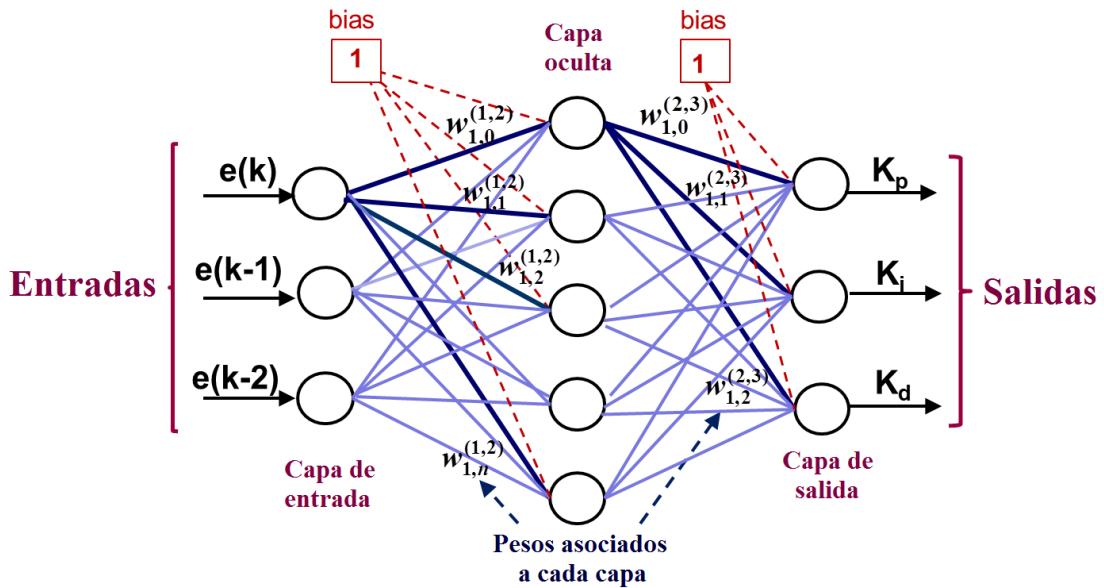
Para el diseño de la red neuronal que va a sintonizar al PID, se va a utilizar una representación alternativa de la ecuación del PID en tiempo discreto presentada en la Ecuación 4. La nueva expresión viene dada por la Ecuación 6, donde  $K_p$  es la constante del término proporcional,  $K_i$  es la constante del término integral y  $K_d$  es la constante del término derivativo.

$$\Delta u(k) = \Delta u[k-1] + K_p \cdot (e[k] - e[k-1]) + K_i \cdot e[k] + K_d \cdot (e[k] - 2 \cdot e[k-1] + e[k-2]) \quad (6)$$

En consecuencia, los parámetros de sintonización del PID utilizando redes neuronales son los términos  $K_p$ ,  $K_i$  y  $K_d$ . La relación entre las constantes  $K_i$  y  $K_d$  de la Ecuación 6 con los términos  $T_i$  y  $T_d$  de la Ecuación 4 viene dada por la Ecuación 7, donde  $T$  representa el tiempo de muestreo.

$$K_i = K_p \cdot \frac{T}{T_i} \quad K_d = K_p \cdot \frac{T_d}{T} \quad (7)$$

Por lo tanto, el esquema general del proceso controlado usando una Red Neuronal se muestra en la Figura 10, donde como se puede observar, los parámetros de sintonización  $K_p$ ,  $K_i$  y  $K_d$  son proporcionados por la salida de la red neuronal. Además, nuestro método propuesto necesita conocer el error en tiempo real del controlador PID (Ecuación 6) para actualizar eficientemente los parámetros de sintonización. Este error viene dado por el valor absoluto de la diferencia entre el ancho de banda garantizado de cada ONU ( $B_{\text{guaranteed}}^{\text{sla} \in \text{onu}_i}$ ) y su ancho de banda asignado medio en tiempo real ( $\overline{B_{\text{alloc}}^{\text{onu}_i}}[n]$ ), es decir,  $e[k] = |B_{\text{guaranteed}}^{\text{sla} \in \text{onu}_i} - \overline{B_{\text{alloc}}^{\text{onu}_i}}[n]|$ .



**Figura 10:** Diagrama de flujo de la Red Neuronal diseñada para controlar los requisitos de ancho de banda de QoS

Por otro lado, para diseñar la arquitectura de la red neuronal es necesario seleccionar algunos parámetros, tales como el número de capas, las neuronas, las funciones de activación y los métodos de entrenamiento. En cuanto al número de capas, la literatura revela que muchos algoritmos implementan una capa oculta (aparte de la capa de entrada y de la capa de salida), ya que proporciona una estructura simple y buen rendimiento [36-38]. El número de neuronas en la capa de salida coincide con los parámetros de sintonización del PID, por lo que se establece en tres ( $K_p$ ,  $K_i$  y  $K_d$ ) y el número de neuronas en la capa de entrada es tres porque los parámetros de sintonización se calculan sobre la base de los tres últimos errores,  $e[k]$ ,  $e[k-1]$  y  $e[k-2]$  (siguiendo la Ecuación 6). Por último, el número de neuronas en la capa oculta depende de la exactitud que se quiera lograr, pero la literatura consultada señala que un número entre una y cuatro veces el número de entradas es suficiente, por lo que seleccionamos cinco neuronas inicialmente [36-38].

Además, se añadió una señal de “*bias*” en las capas intermedia y de salida para buscar una respuesta más estable del sistema. Entonces, la red neuronal artificial de avance directo (*feedforward*) es un perceptrón de múltiples capas que asigna conjuntos de datos de entrada a un conjunto de salidas apropiadas, es decir, parámetros de sintonización adecuados según los errores (Figura 10). Por otro lado, la elección de la función de activación se basa en el rango de valores deseado para los resultados. De esta manera, se elige la función tangente hiperbólica para la capa intermedia, ya que es una de las más clásicas y extendidas. Para la capa de salida, seleccionamos la función lineal (en valores positivos), debido a que los parámetros de sintonización ( $K_p$ ,  $K_i$  y  $K_d$ ) pueden tomar cualquier valor positivo, por lo que no es conveniente que los valores estén limitados por un umbral máximo y mínimo.

Las redes neuronales tradicionales utilizan técnicas de aprendizaje en las que se proporciona a la red un conjunto de buenos ejemplos para aprender y trabajar eficientemente manteniendo el peso entre las neuronas con valores fijos. Por el contrario, nuestra propuesta constantemente aprende de los últimos errores (aplicando una técnica de aprendizaje incremental) por lo que cambian continuamente los pesos entre las neuronas y así la adaptación a las condiciones de red en tiempo real son más óptimas. El estudio matemático utilizado para incrementar la matriz de pesos se basa en varios estudios publicados en [14-17]. De esta manera, la Red Neuronal de retro propagación (*backpropagation*) (BP) es una de las más implementadas en muchos sistemas. Utiliza el algoritmo de aprendizaje de retro propagación aplicado en dos etapas: avance directo (*feedforward*) de los datos y retro propagación (*backpropagation*) de los errores. En la fase de avance directo (*feedforward*) de los datos, el error del controlador PID ( $e[k]$ ,  $e[k-1]$  y  $e[k-2]$ ) se inserta en la capa de entrada y se propaga a las siguientes capas (capas ocultas y de salida), y el algoritmo obtiene los nuevos parámetros de sintonización para los ciclos siguientes. Por otro lado, en la etapa de retro propagación (*backpropagation*), con los últimos valores de salida (parámetros de sintonización), se calcula el error cometido (típicamente usando el de rendimiento cuadrático) y se propagan hacia atrás (de salida a entrada). Por lo tanto, los pesos entre

las neuronas se ajustan usando algún método, típicamente el algoritmo del gradiente descendiente, basado en estos errores. Estas dos fases se repiten continuamente a lo largo del tiempo para asegurar un buen desempeño del proceso de asignación de ancho de banda. En particular, en nuestra propuesta en cada número de iteraciones del PID, llamado  $n$ , se actualizan los parámetros de sintonización (algoritmo *feedforward*). Además, a cada número de iteraciones del PID (llamado  $m$ ), también se actualiza la matriz de pesos (algoritmo de *retro propagación*). El valor de  $m$  tiene que ser mayor que el de  $n$  debido a que los parámetros de sintonización deben ser actualizados teniendo en cuenta la nueva matriz de pesos fija durante algún tiempo. Si no es así, pueden existir periodos con cambios en la matriz de pesos que no se reflejen en el cálculo de los nuevos parámetros de sintonización.

En particular, el algoritmo de propagación de *feedforward* calcula los valores de entrada y salida de cada capa de acuerdo con los pesos actuales y las últimas entradas de la Red Neuronal (los últimos tres errores y la señal de “*bias*”), logrando los nuevos parámetros de sintonización en la salida de la misma. Entonces, la capa de entrada sigue la Ecuación 8, en la que las entradas  $x(j)$  corresponden a  $e[k]$ ,  $e[k-1]$ ,  $e[k-2]$  y la entrada de “*bias*” mientras que  $O_j^{(1)}(k)$  es la salida de la capa. Los superíndices en las diferentes ecuaciones (1, 2, 3) están asociados con la capa, es decir, la entrada, la capa oculta y la capa de salida. Además, el exponente de los pesos se refiere a pesos entre dos capas, por ejemplo  $w_{ij}^{(1,2)}$  se corresponde a los pesos entre la capa de entrada y la capa oculta.

$$O_j^{(1)}(k) = x(j) \quad (j = 0, 1 \dots) \quad (8)$$

Para la capa oculta (subíndice 2), las entradas  $(I_i^{(2)}(k))$  y salidas  $(O_i^{(2)}(k))$  siguen las Ecuaciones 9 y 10. En estas ecuaciones,  $w_{ij}^{(1,2)}$  son los pesos entre la capa de entrada y la capa oculta. Como se dijo antes, la función tangente hiperbólica se elige como la función de activación para esta capa  $(f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}})$ .

$$I_i^{(2)}(k) = \sum_j w_{ij}^{(1,2)} \cdot O_j^{(1)}(k) \quad (9)$$

$$O_i^{(2)}(k) = f(I_i^{(2)}(k)) \quad (k = 0, 1 \dots) \quad (10)$$

Finalmente, las entradas y salidas de la capa de salida siguen las Ecuaciones 11 y 12. La salida de esta capa se corresponde con los tres parámetros de sintonización ( $K_p$ ,  $K_i$  y  $K_d$ ) y la función de activación para esta capa es la función lineal ( $g(x) = x$ ). Sin embargo, la función lineal se trunca para evitar valores negativos ya que los parámetros de sintonización tienen que ser positivos. Los términos  $w_{li}^{(2,3)}$  son las ponderaciones entre la capa oculta y la capa de salida.

$$I_l^{(3)}(k) = \sum_i w_{li}^{(2,3)} \cdot O_i^{(2)}(k)$$

$$O_i^{(3)}(k) = g \left( I_l^{(3)}(k) \right) \quad (l = 0, 1 \dots)$$
(11)

Así que:

$$\begin{aligned} O_1^{(3)}(k) &= K_p \\ O_2^{(3)}(k) &= K_i \\ O_3^{(3)}(k) &= K_d \end{aligned}$$
(12)

Por otro lado, el algoritmo de retro propagación modifica los factores ponderados en la Red Neuronal, comenzando desde la capa de salida hasta la capa de entrada. Para actualizar los pesos en la red neuronal, se utiliza el error cuadrático (Ecuación 13), basado en el parámetro bajo control (niveles garantizados de ancho de banda). Además, los pesos se modifican utilizando el método del gradiente descendiente. Entonces, los incrementos se ajustan de acuerdo con la dirección del gradiente negativo a los coeficientes de ponderación (Ecuación 14), así que un término de inercia ( $\alpha$ ) y un coeficiente de tasa de aprendizaje ( $\eta$ ) impactan directamente sobre la evolución y la velocidad de convergencia de la Red Neuronal. Aunque ambos parámetros se fijan inicialmente a un valor de 0.1, en la sección de resultados en base a lo consultado en la bibliografía en [14-17], analizaremos su impacto en la Red Neuronal. En dicha ecuación,  $\Delta w_{li}^{(2,3)}(k)$  denotan los últimos incrementos logrados en la última modificación de pesos entre las capas oculta y de salida.

$$J_1 = \left( \frac{1}{2} \right) \cdot e^2 [k] = \left( \frac{1}{2} \right) \cdot \left( B_{\text{guaranteed}}^{\text{slaEonui}} - B_{\text{alloc}}^{\text{onui}} \right)^2$$
(13)

$$\Delta w_{li}^{(2,3)}(k) = -\eta \cdot \frac{\partial J_1}{\partial w_{li}^{(2,3)}} + \alpha \cdot \Delta w_{li}^{(2,3)}(k-1)$$
(14)

Para calcular  $\frac{\partial J_1}{\partial w_{li}^{(2,3)}}$ , seguimos la Ecuación 15. En ella, el término  $\frac{\partial y(k)}{\partial \Delta u(k)}$  se desconoce, por lo que lo sustituimos por la aproximación  $\text{sgn} \left( \frac{\partial y(k)}{\partial \Delta u(k)} \right)$  (la falta de precisión puede ser corregida con  $\eta$ ), como se considera en varios estudios [14-17].

$$\frac{\partial J_1}{\partial w_{li}^{(2,3)}} = \frac{\partial J_1}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial \Delta u(k)} \cdot \frac{\partial \Delta u(k)}{\partial O_l^{(3)}(k)} \cdot \frac{\partial O_l^{(3)}(k)}{\partial I_l^{(3)}(k)} \cdot \frac{\partial I_l^{(3)}(k)}{\partial w_{li}^{(2,3)}}$$
(15)

Tomando en cuenta la Ecuación 6 y 12, obtenemos las expresiones para  $\frac{\partial \Delta u(k)}{\partial O_l^{(3)}(k)}$  (Ecuación 16):

$$\begin{aligned}\frac{\partial \Delta u(k)}{\partial O_1^{(3)}(k)} &= \frac{\partial \Delta u(k)}{\partial K_p} = e(k) - e(k-1) \\ \frac{\partial \Delta u(k)}{\partial O_2^{(3)}(k)} &= \frac{\partial \Delta u(k)}{\partial K_i} = e(k) \\ \frac{\partial \Delta u(k)}{\partial O_3^{(3)}(k)} &= \frac{\partial \Delta u(k)}{\partial K_d} = e(k) - 2 \cdot e(k-1) + e(k-2)\end{aligned}\tag{16}$$

Finalmente, con estas expresiones podemos simplificar la fórmula para los factores ponderados  $\Delta w_{ii}^{(2,3)}(k)$  de la capa de salida, como se muestra en la Ecuación 17:

$$\begin{aligned}\Delta w_{ii}^{(2,3)}(k) &= \eta \cdot \delta_i^{(3)} \cdot O_i^{(2)}(k) + \alpha \cdot \Delta w_{ii}^{(2,3)}(k-1) \\ \text{Dónde: } \delta_i^{(3)} &= e(k) \cdot \text{sgn}\left(\frac{\partial y(k)}{\partial \Delta u(k)}\right) \cdot \frac{\partial \Delta u(k)}{\partial O_1^{(3)}(k)} \cdot g' [I_i^{(3)}(k)]\end{aligned}\tag{17}$$

De la misma manera que la capa de salida, modificamos los factores ponderados en la capa oculta, siguiendo las expresiones de la Ecuación 18:

$$\begin{aligned}\Delta w_{ij}^{(1,2)}(k) &= \eta \cdot \delta_i^{(2)} \cdot O_j^{(1)}(k) + \alpha \cdot \Delta w_{ij}^{(1,2)}(k-1) \\ \text{Dónde: } \delta_i^{(2)} &= f' [I_i^{(2)}(k)] \cdot \sum_{l=1}^3 \delta_l^{(3)} \cdot w_{li}^{(2,3)}(k)\end{aligned}\tag{18}$$

Como ya se ha visto en la Figura 10, nuestro punto de partida se basa en añadir dos entradas de “bias” ( $X_0=1$ ), una que se conecte con todas las neuronas de la capa oculta, y otra que lo haga con todas la de la capa de salida, pero sin modificar el número de salidas de la red neuronal, que sigue siendo de tres.

En cuanto al código programado, sólo hubo que modificar las siguientes líneas con respecto a lo que ya tenían implementado en el grupo de trabajo, para añadir las señales comentadas anteriormente:

```
int HIDDEN_NN=6;
int IN_NN=4;
double V_IN_NN[4];
double W1 [6][4];
double Incremento_W1[6][4];
double IncrementoAnt_W1[6][4];
double W2[3][6];
double Incremento_W2[3][6];
double IncrementoAnt_W2[3][6];
double out_hidden[6];
double deltaHidden[6];
double in_hidden[6];
```

```
double back_hidden[6];
V_IN_NN[3]=1.0;
in_hidden[5]=1.0;
```

## 4.4 Resultados del estudio de simulación

### 4.4.1 Escenario de simulación genérico

A continuación, en la Tabla 1, se especifican los principales parámetros de simulación de las redes neuronales para realizar las diversas pruebas del controlador PID para gestionar los recursos en las redes PON.

Parámetros de la simulación	Valores
Número de ONUs	16
Número de longitudes de onda	1
Capacidad de la red	1 Gbps
Tamaño del paquete	Variable entre 64 y 1500 bytes
Tamaño del buffer	6 MBytes
Tamaño de los paquetes	Trimodal
Método de inserción de paquetes	Método de inserción de prioridad de colas
Método de extracción de paquetes	Método de extracción de colas de prioridad
Número de Streams	32
Número de clases de servicio	3
Número de SLAs	3
Service Level Agreement (SLA) Perfiles de abonado	SLA <sub>0</sub> : 1 ONU Ancho de banda garantizado: 100 Mbps El “0” indica que es el de mayor prioridad
	SLA <sub>1</sub> : 5 ONUs Ancho de banda garantizado: 75 Mbps El “1” denota que es el de prioridad media
	SLA <sub>2</sub> : 10 ONUs Ancho de banda garantizado: 50 Mbps El “2” indica que es la prioridad más baja
Carga de las fuentes de tráfico óptico	0.4
Carga total en la red	0.85
Tiempo del PID	3 segundos
Tiempo de muestreo	3 segundos
Número de iteraciones del PID por entrenamiento (n)	4
Número de iteraciones del PID por auto-aprendizaje (m)	8
Normalización	10 <sup>9</sup>
Coefficientes de inercia	$\alpha = 0.1$ y $\eta = 0.1$

**Tabla 1:** Parámetros de la simulación considerados en el escenario de simulación para PID

#### 4.4.2 Análisis del impacto de las señales de “bias” en la red neuronal

En primer lugar, en las Figuras 11, 12 y 13, y para los valores del escenario de simulación genérico indicado en la Tabla 1, se muestra cómo evoluciona en tiempo real el ancho de banda asignado (en Mbps), viendo las diferencias que hay entre que estén activadas o no las señales de “bias” en las capas de la red neuronal, para los tres SLAs ( $SLA_0$ ,  $SLA_1$ ,  $SLA_2$ ) de los que se dispone.

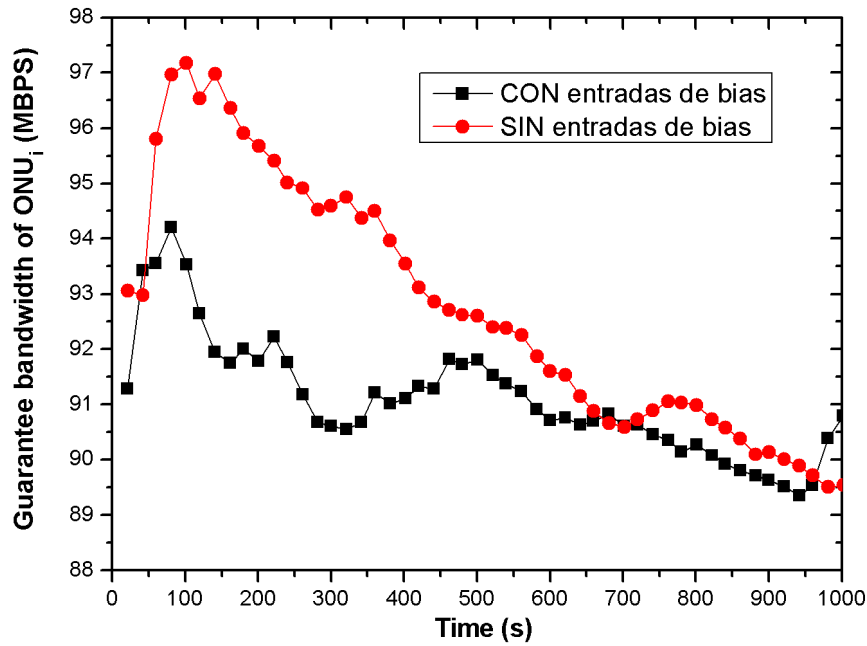


Figura 11: Ancho de banda garantizado para el  $SLA_0$

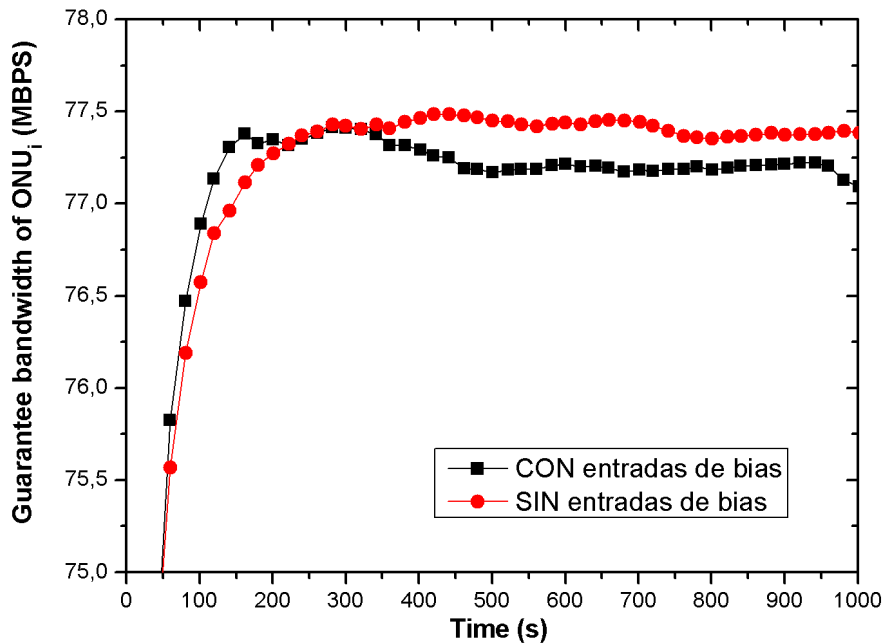
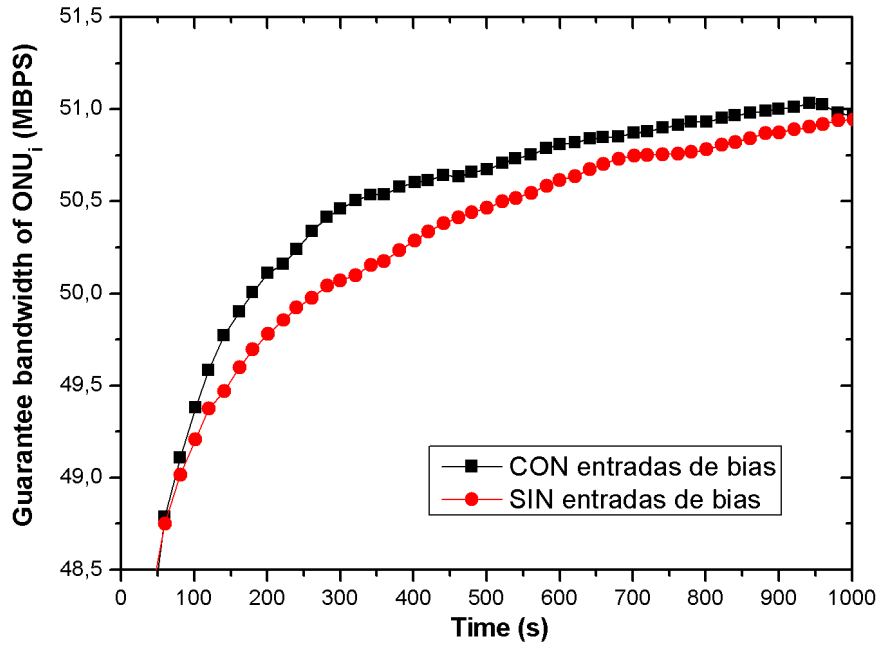


Figura 12: Ancho de banda garantizado para el  $SLA_1$



**Figura 13:** Ancho de banda garantizado para el SLA<sub>2</sub>

Se observa que para todos los SLAs, pero sobre todo inicialmente para el SLA<sub>0</sub>, el ancho de banda varía de manera inestable, ya que la red tiene un periodo de adaptación, hasta que se comienzan a obtener unos anchos de banda garantizados bastante estables.

También se puede ver, que con la inclusión de las señales de “*bias*”, se obtiene una convergencia ligeramente más rápida hasta que se alcanzan los valores establecidos para los anchos de banda garantizados.

Se repite la misma prueba que la indicada en el escenario de simulación genérico, pero para unos anchos de banda garantizados de 80 Mbps para el SLA<sub>0</sub>, 60 Mbps para el SLA<sub>1</sub> y 60 Mbps para el SLA<sub>2</sub>, observando en las Figuras 14, 15 y 16 cómo evoluciona en tiempo real el ancho de banda asignado (Mbps), para comprobar si se produce el mismo comportamiento que antes, o sucede algo distinto.



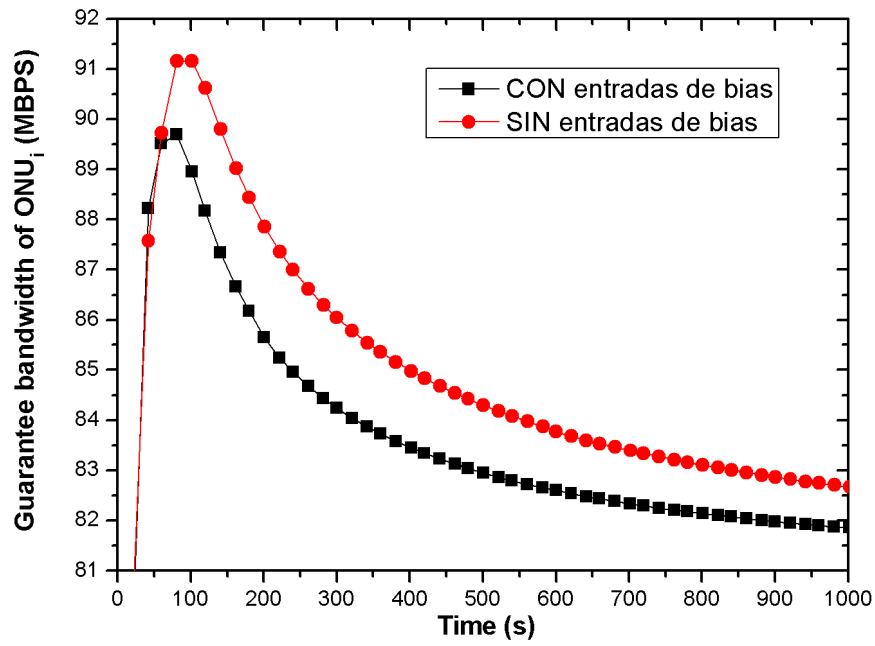


Figura 14: Ancho de banda garantizado para el SLA<sub>0</sub>

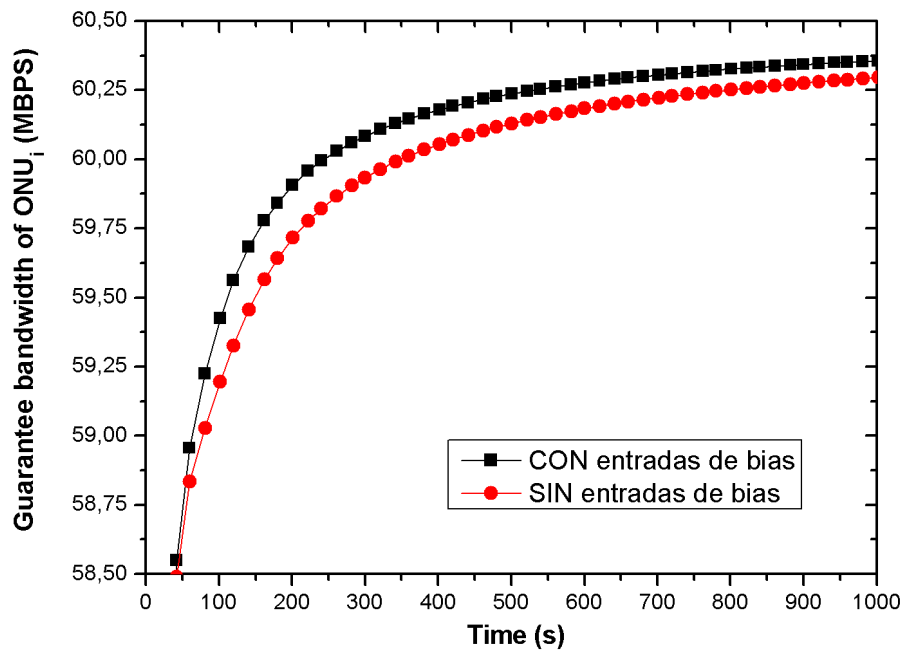


Figura 15: Ancho de banda garantizado para el SLA<sub>1</sub>

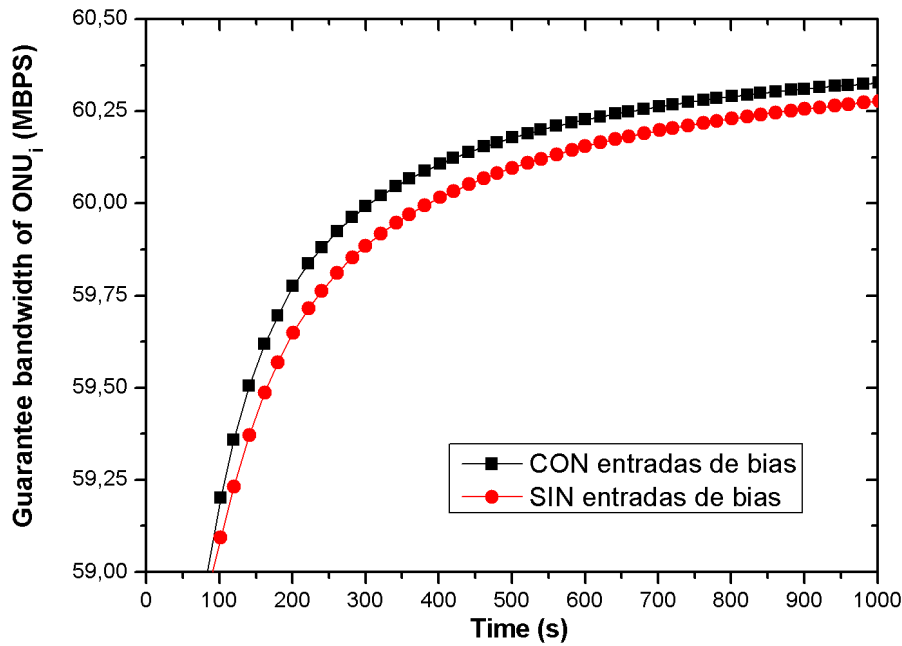


Figura 16: Ancho de banda garantizado para el SLA<sub>2</sub>

Se observa el mismo comportamiento que tenía lugar anteriormente, ya que se comprueba que se alcanzan de manera ligeramente más rápida unos valores estables para el ancho de banda garantizado.

A continuación, se va a estudiar cómo varían los parámetros de sintonización del PID ( $K_p$ ,  $T_i$  y  $T_d$ ) tanto si las señales de “bias” están activadas como si no. Basándonos en los parámetros definidos en el escenario genérico de simulación (Tabla 1), se ilustra este comportamiento en las Figuras 17, 18 y 19, para  $K_p$ ,  $T_i$  y  $T_d$  respectivamente.

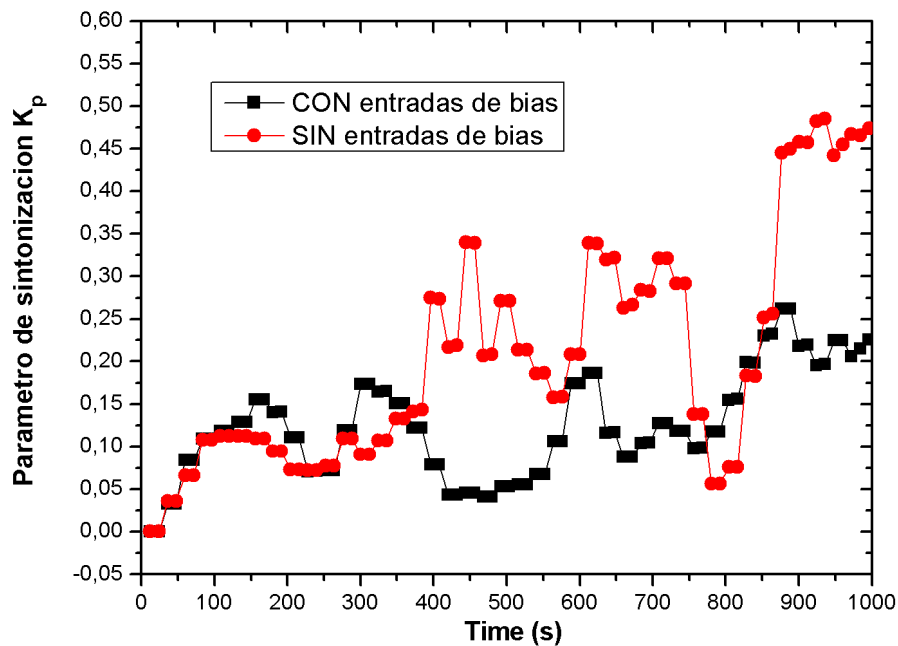
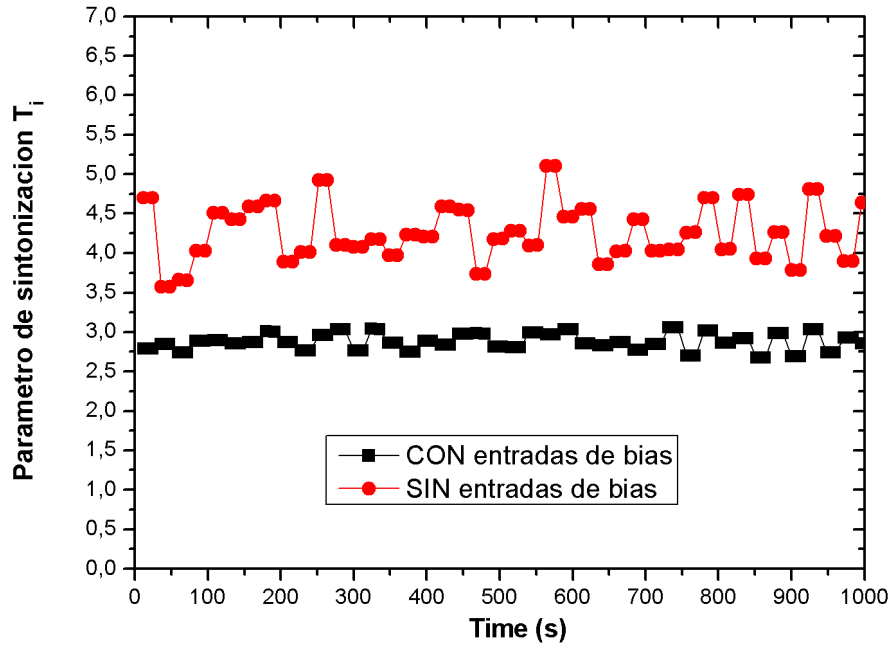
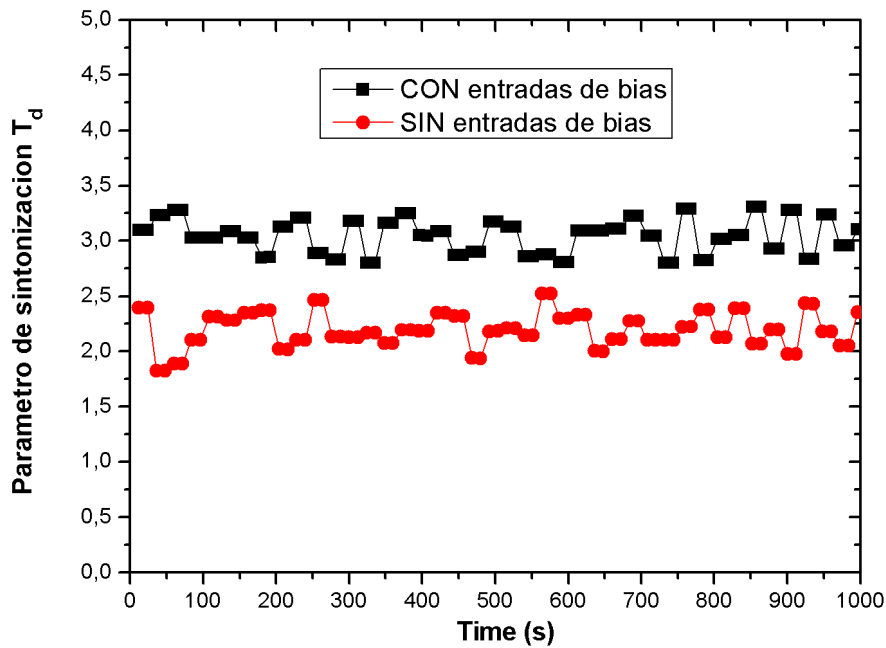


Figura 17: Parámetro de sintonización  $K_p$  para unos anchos de banda garantizados de 100, 75 y 50 Mbps



**Figura 18:** Parámetro de sintonización  $T_i$  para unos anchos de banda garantizados de 100, 75 y 50 Mbps



**Figura 19:** Parámetro de sintonización  $T_d$  para unos anchos de banda garantizados de 100, 75 y 50 Mbps

Se observa en las figuras anteriores, que con las señales de “bias” activadas, estos parámetros de sintonización tienen unos valores más estables que sin ellas. Esto se observa principalmente para el parámetro  $K_p$  (Figura 17), donde se ve que con las señales de “bias” activadas se obtienen unos valores más bajos que sin ellas, y para  $T_i$  (Figura 18) y  $T_d$  (Figura 19), donde se obtienen unos valores que oscilan mucho menos, al contrario que sucedía sin la señal de “bias”, donde había mayores picos y variaciones.

### 4.4.3 Análisis del impacto de diferentes factores de normalización

En primer lugar, indicar que este parámetro de normalización se utiliza a la hora de actualizar las deltas que van a servir para calcular los incrementos (Ecuación 16). En esta ecuación, se utilizaban 3 errores cuadráticos medios de tres ciclos diferentes (el error presente y los errores del anterior ciclo y el dos veces anterior), los cuales deben ir normalizados, debido a que el error cuadrático medio (Ecuación 13) podría llegar a ser muy grande al estar elevado al cuadrado en bits, y los pesos de la red neuronal se encuentran en órdenes mucho más bajos. De ahí la importancia de este parámetro, ya que si no es el correcto, la evolución de los pesos y, en consecuencia de los parámetros de sintonización ( $K_p$ ,  $T_i$  y  $T_d$ ), no sería la adecuada y se obtendrían valores incoherentes.

Por tanto, ya con las señales de “bias” activadas, y para los anchos de banda garantizados indicados en el escenario de simulación genérico (Tabla 1), se comprueba en las Figuras 20, 21 y 22 la evolución en tiempo real del ancho de banda garantizado para distintos valores de normalización ( $10^6$ ,  $10^9$ ,  $10^{15}$ ) y así se justifica cuál es la más óptima.

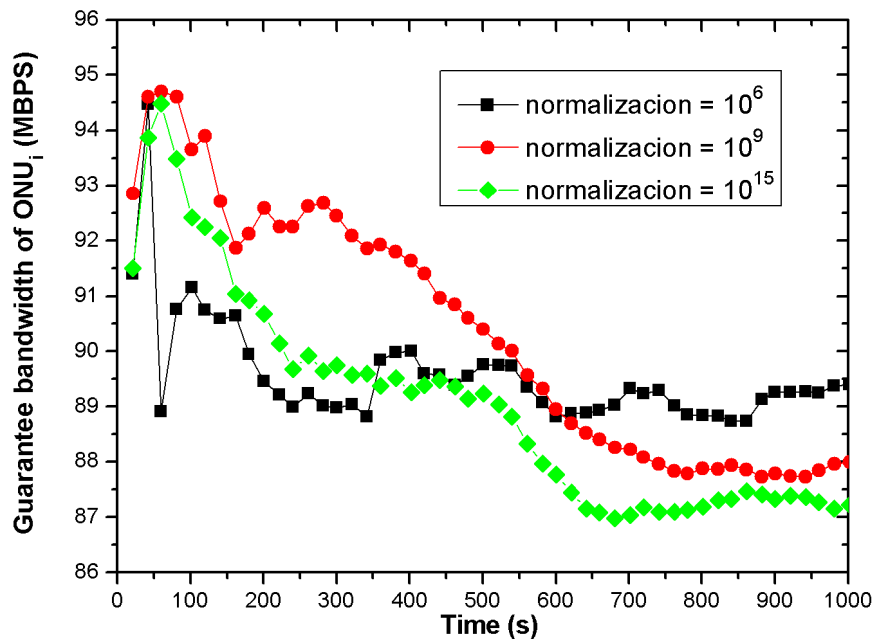
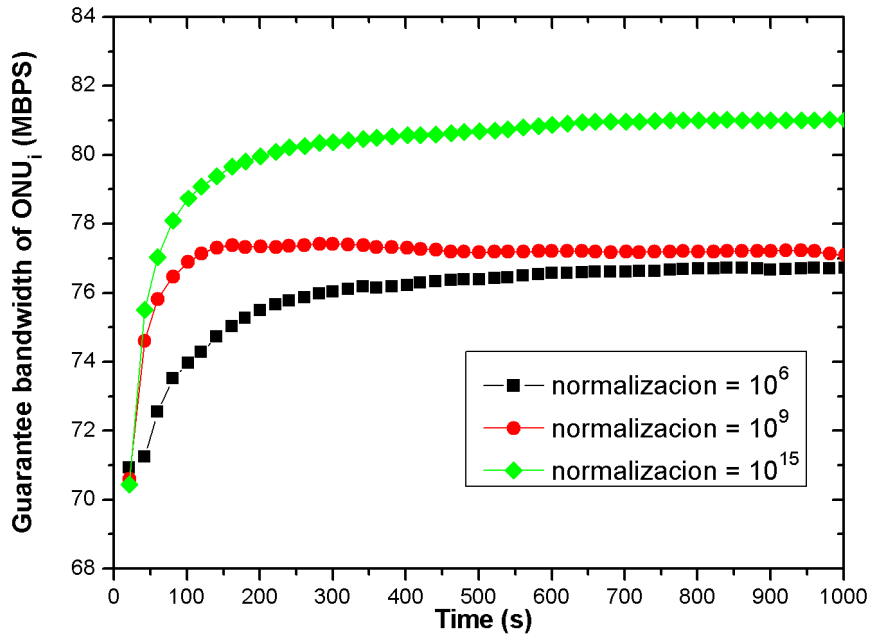
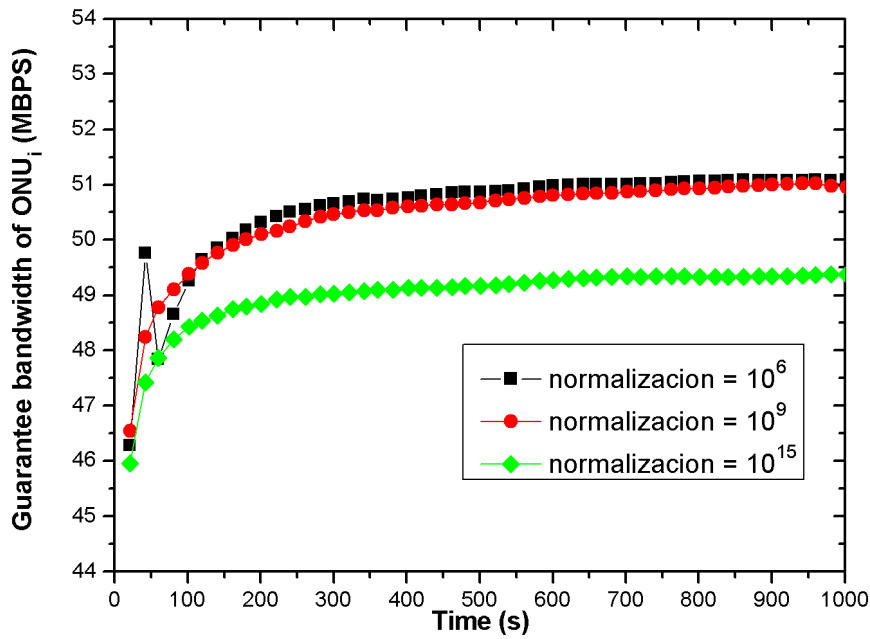


Figura 20: Ancho de banda garantizado para distintas normalizaciones en el SLA<sub>0</sub>

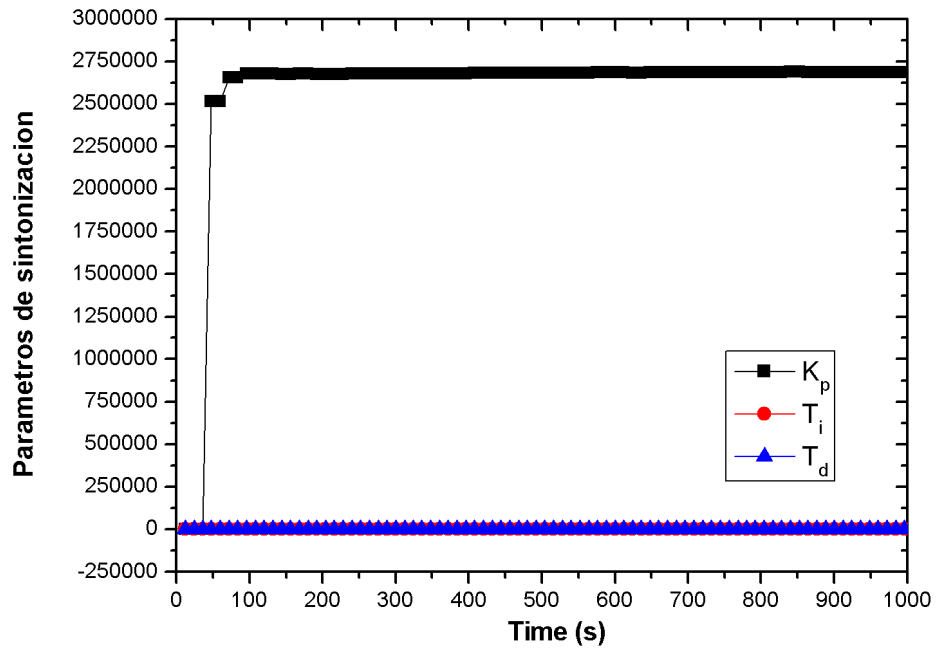


**Figura 21:** Ancho de banda garantizado para distintas normalizaciones en el SLA<sub>1</sub>

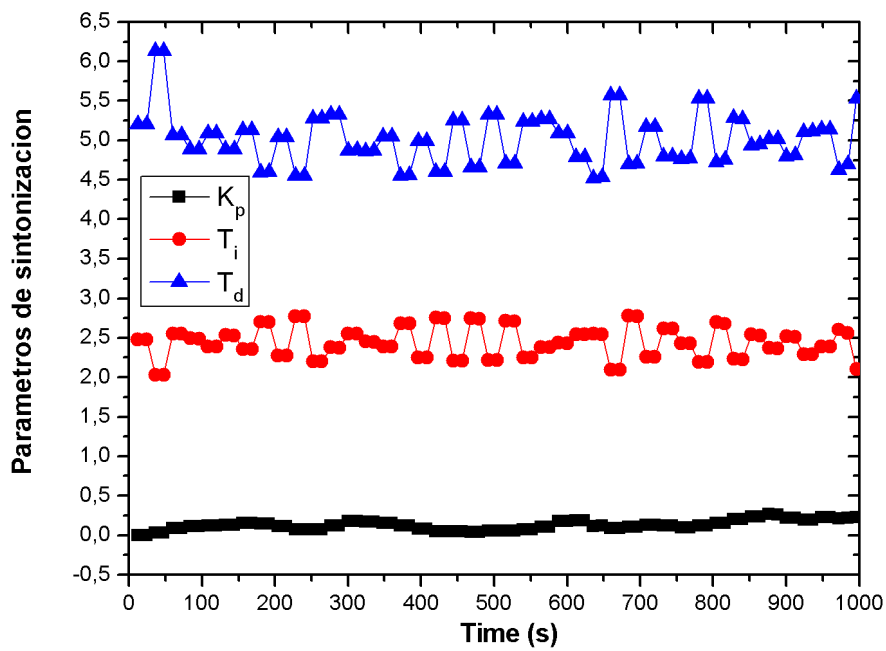


**Figura 22:** Ancho de banda garantizado para distintas normalizaciones en el SLA<sub>2</sub>

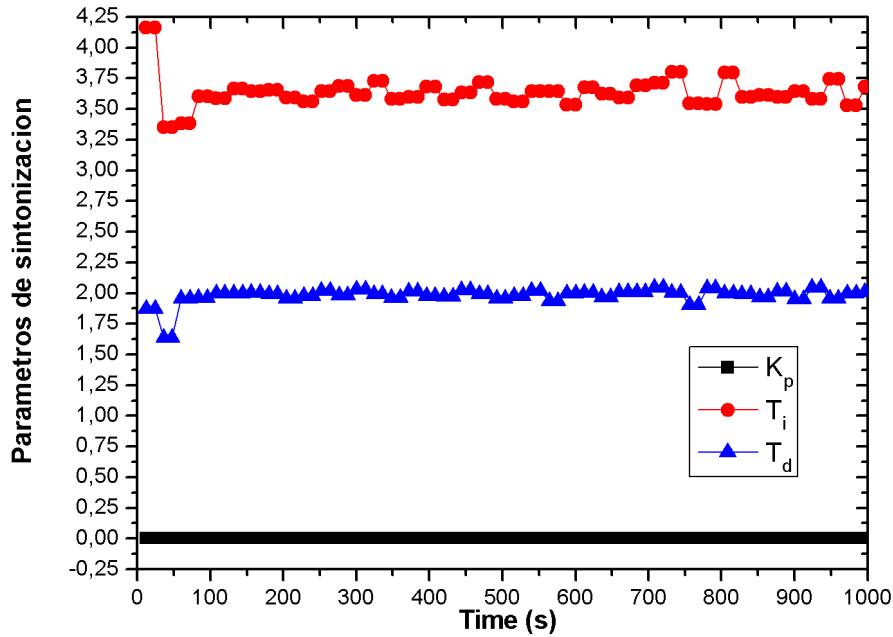
Junto con el análisis del impacto del ancho de banda, también se analiza la evolución en tiempo real de los parámetros de sintonización del PID ( $K_p$ ,  $T_i$  y  $T_d$ ) para cada una de las normalizaciones (Figuras 23, 24 y 25), pues existe una correlación entre ambos.



**Figura 23:** Parámetros de sintonización para una normalización de  $10^6$



**Figura 24:** Parámetros de sintonización para una normalización de  $10^9$



**Figura 25:** Parámetros de sintonización para una normalización de  $10^{15}$

A la vista de las gráficas anteriores, tanto de los anchos de banda garantizados (Figuras 20, 21 y 22) como de los parámetros de sintonización en las Figuras 23, 24 y 25, podemos indicar que la normalización más correcta es la de  $10^9$  (Figura 24), ya que es la que menos tarda en estabilizar el sistema, con una asignación adecuada de los niveles de ancho de banda garantizados, y con unos valores adecuados para los parámetros de sintonización. Si nos fijamos especialmente en el parámetro  $K_p$ , y la normalización es muy baja ( $10^6$ ) (Figura 23), adquiere unos valores muy elevados (del orden de  $3,5 \cdot 10^6$ ), mientras que si es muy alta ( $10^{15}$ ), los valores para este parámetro son demasiado bajos, esto es, del orden de  $1,8 \cdot 10^{-7}$  (Figura 25).

Por lo tanto, se puede concluir que para normalizaciones de valor bajo, como por ejemplo  $10^6$ , se obtendrían unos valores de los parámetros de sintonización muy grandes e incoherentes (del orden de  $3,5 \cdot 10^6$  para  $K_p$  como ya se indicó anteriormente), debido a que la normalización del error es demasiado pequeña, lo que provoca que los errores sean mucho más grandes y se desestabilice la función de control del PID, tal y como se observa en la evolución de los parámetros  $K_p$ ,  $T_i$  y  $T_d$  (Figura 23).

En el otro extremo, con un valor de normalización muy alto, por ejemplo  $10^{15}$ , se obtendrían unos valores demasiado bajos, sobre todo para  $K_p$  (del orden de  $1,8 \cdot 10^{-7}$  como se vio anteriormente), por lo que los incrementos de la matriz oculta de pesos serían muy pequeños, y la matriz evolucionaría muy lentamente. Este hecho provoca que la Red Neuronal tarda mucho más en alcanzar los valores de ancho de banda garantizados para los diferentes perfiles de abonado y los resultados no serán tan exactos a los estipulados, tal y como se observa en las Figuras 21 y 22 para el ancho de banda garantizado en el  $SLA_1$  y en el  $SLA_2$ , respectivamente.

#### 4.4.4 Análisis de los errores obtenidos en la Red Neuronal

En este punto, se va a comprobar cómo evoluciona en tiempo real el error para las diferentes normalizaciones consideradas en los apartados anteriores, para los mismos anchos de banda garantizados que se indicaban en el escenario de simulación genérico (Tabla 1). En concreto, en la Figura 26, se mostrará el error para unas normalizaciones de  $10^6$ ,  $10^9$  y  $10^{15}$ :

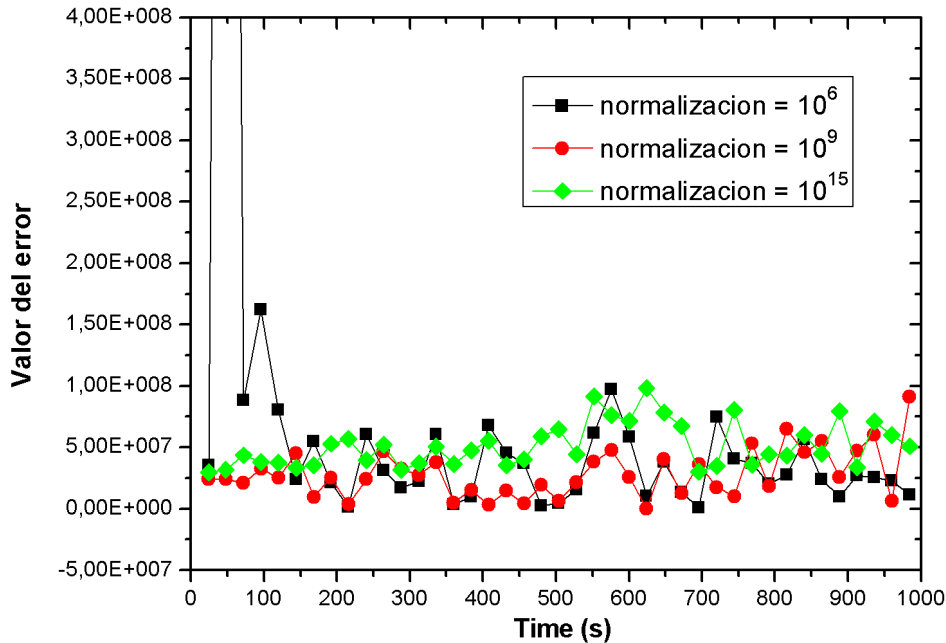


Figura 26: Error para unas normalizaciones de  $10^6$ ,  $10^9$  y  $10^{15}$

Se observa que para una normalización de  $10^6$ , el error obtenido en función del tiempo es bastante grande al comienzo de la simulación, en concreto, toma valores del orden de  $2,25 \cdot 10^9$  bits (dichos valores no se observan en esta figura para poder comparar mejor el resto de normalizaciones). Como consecuencia de esto y de lo comentado anteriormente en referencia a que los parámetros de sintonización que se obtienen son muy elevados, esta normalización no va a ser la correcta.

Mientras, para una normalización de  $10^9$ , se ve que se obtiene un error mucho menor, con un pico máximo de  $1 \cdot 10^8$  bits, y como también vimos antes, este nivel de normalización da lugar a unos parámetros de sintonización bastante adecuados.

Por último, para la normalización de  $10^{15}$ , se observa que el error ronda los mismos valores que para la normalización anterior, pero tal y como se comentó anteriormente, se obtienen unos valores muy bajos para el parámetro de sintonización  $K_p$ . Esto hace que el sistema evolucione muy lentamente y necesite más tiempo para estabilizarse, es decir, para conseguir los valores de los parámetros del PID correctos y garantizar los anchos de banda adecuados.

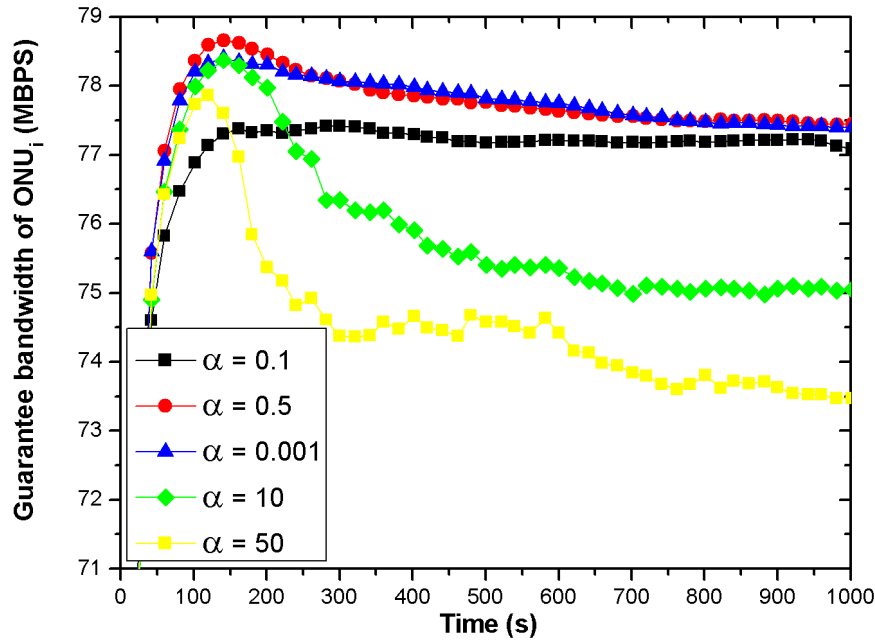


#### 4.4.5 Análisis del impacto de los valores de inercia en la red neuronal

En este apartado se va a analizar la evolución en tiempo real del ancho de banda garantizado para cada uno de los perfiles de abonado (SLAs), estudiando cómo influye la variación de los dos coeficientes de inercia utilizados en la Ecuación 14 ( $\alpha$  y  $\eta$ ), que marcarán la magnitud de desplazamiento de los pesos, lo que repercutirá en la velocidad de convergencia del algoritmo.

Estos coeficientes, como se puede observar en la Ecuación 14 ya mencionada, van multiplicando a diferentes términos. Por un lado,  $\alpha$  multiplica a la acumulación de los incrementos  $\left(\Delta w_{li}^{(2,3)}(k-1)\right)$ , mientras que  $\eta$  multiplica a las deltas  $\left(\frac{\partial J_1}{\partial w_{li}^{(2,3)}}\right)$ , que son un incremento parcial. El coeficiente que más impacto tiene a la hora de aumentar la velocidad de convergencia es por tanto  $\alpha$ , debido a que este incremento es acumulado, mientras que  $\eta$  va a ser un incremento parcial, por lo que el término que multiplica  $\alpha$  va a ser más grande que el de  $\eta$ .

En nuestro caso, se van a fijar inicialmente ambos a 0.1, tal y como se vio en la literatura consultada [14-17], ya que permite que los incrementos de la matriz y la velocidad de convergencia no sean excesivamente grandes ni pequeños. Sin embargo, en este apartado se va a analizar su impacto representando la evolución en tiempo real del ancho de banda garantizado para el SLA<sub>1</sub> con  $\eta$  constante y de valor 0.1, mientras se varía el valor de  $\alpha$ , tal y como se ilustra en la Figura 27. Estas pruebas se han realizado con la normalización que hemos comentado anteriormente que era la más óptima, es decir, la de  $10^9$ . El hecho de representar solamente el SLA<sub>1</sub> se debe a que en este perfil de abonado se observa mejor el impacto de los diferentes coeficientes y parámetros a lo largo de la simulación, aunque el mismo comportamiento se obtiene para los otros dos perfiles de abonado (SLA<sub>0</sub> y SLA<sub>2</sub>).

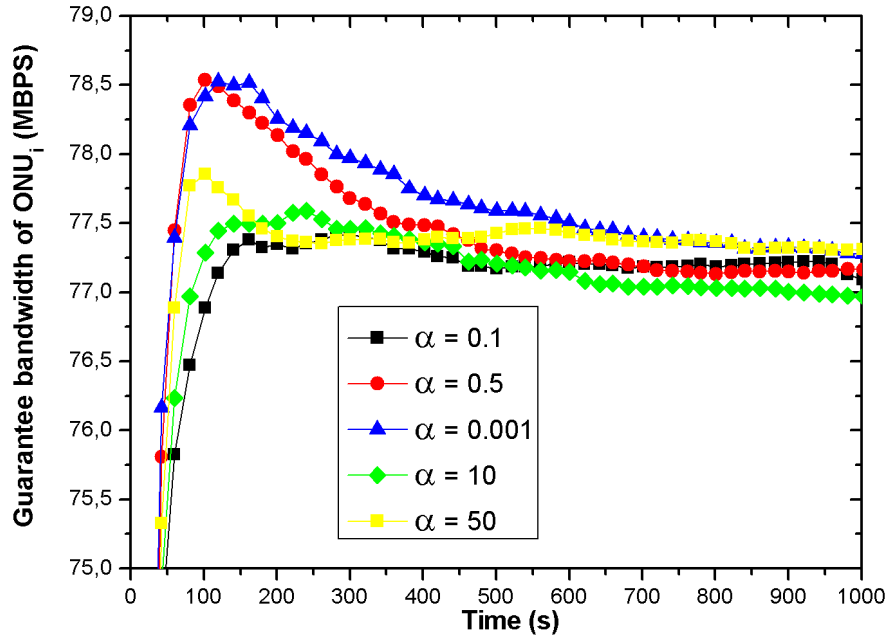


**Figura 27:** Ancho de banda garantizado para el  $SLA_1$  para diferentes valores de  $\alpha$  con  $\eta=0.1$

Se observa en primer lugar, que si ponemos unos valores demasiado extremos para  $\alpha$ , saldrían cosas muy inestables. Por ejemplo, para un valor de  $\alpha$  grande, como puede ser el de  $\alpha=50$ , el ancho de banda garantizado oscila más e incluso podría llegar a obtener valores incorrectos, ya que la velocidad es muy grande y la evolución necesita de velocidades más pequeñas. En el lado opuesto, para valores pequeños, como puede ser  $\alpha=0.001$ , el sistema tardaría mucho más en llegar a los valores correctos de ancho de banda garantizado para los diferentes SLA, debido a que la velocidad de convergencia sería más lenta, tal y como se ve en la Figura 27.

Por lo tanto, un nivel intermedio para el parámetro  $\alpha$ , tal como  $\alpha=0.1$  consigue una velocidad de evolución bastante óptima, y además se observa que tarda menos en converger, por lo tanto, esta será con la que nos quedemos para el resto de pruebas, al ser la más correcta.

Ahora se muestra justo el caso opuesto, es decir, se representa en la Figura 28 la evolución en tiempo real del ancho de banda garantizado para  $\alpha$  constante y de valor 0.1, mientras variamos el valor de  $\eta$ .



**Figura 28:** Ancho de banda garantizado para el  $SLA_1$  para diferentes valores de  $\eta$  con  $\alpha=0.1$

En este caso, se ve que las simulaciones para distintos valores de  $\eta$  tienen un comportamiento más similar, sobre todo a medida que pasa el tiempo. Esto es debido a que este término no va a condicionar tanto el comportamiento del algoritmo. Sin embargo, el valor de  $\eta=0.1$  se observa que proporciona un nivel de estabilidad ligeramente mejor que el resto de valores, sobre todo al comienzo de la simulación.

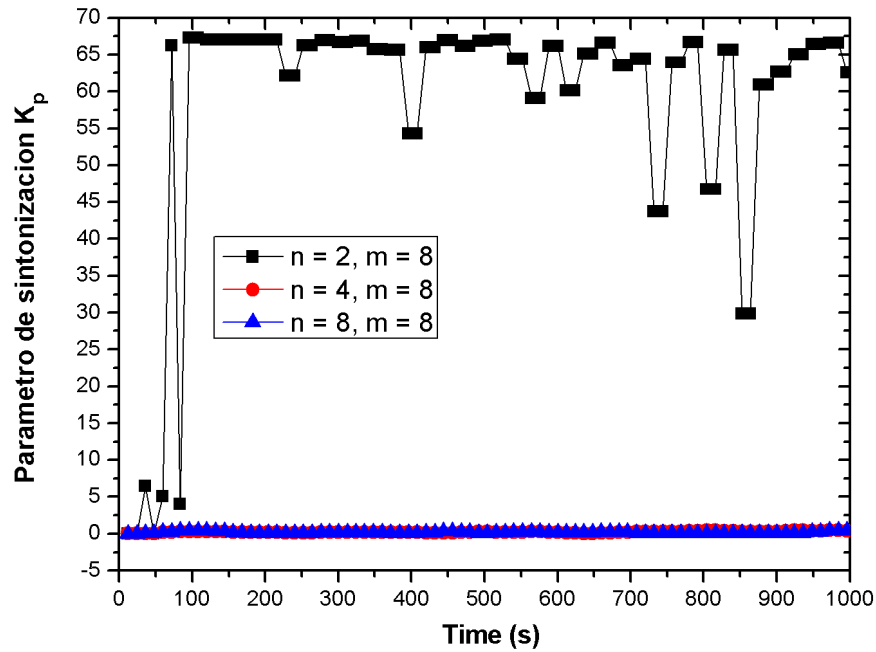
#### 4.4.6 Impacto de los parámetros $n$ y $m$ en el comportamiento de la red neuronal

En este último punto, se va a representar la evolución de los parámetros de sintonización del PID ( $K_p$ ,  $T_i$  y  $T_d$ ) y del ancho de banda garantizado para los distintos perfiles de abonado. Para ello, se va a analizar el impacto de los parámetros  $n$  y  $m$  en la red neuronal, es decir, la velocidad con la que conmutan los valores de los parámetros de sintonización y la matriz de pesos en la red neuronal.

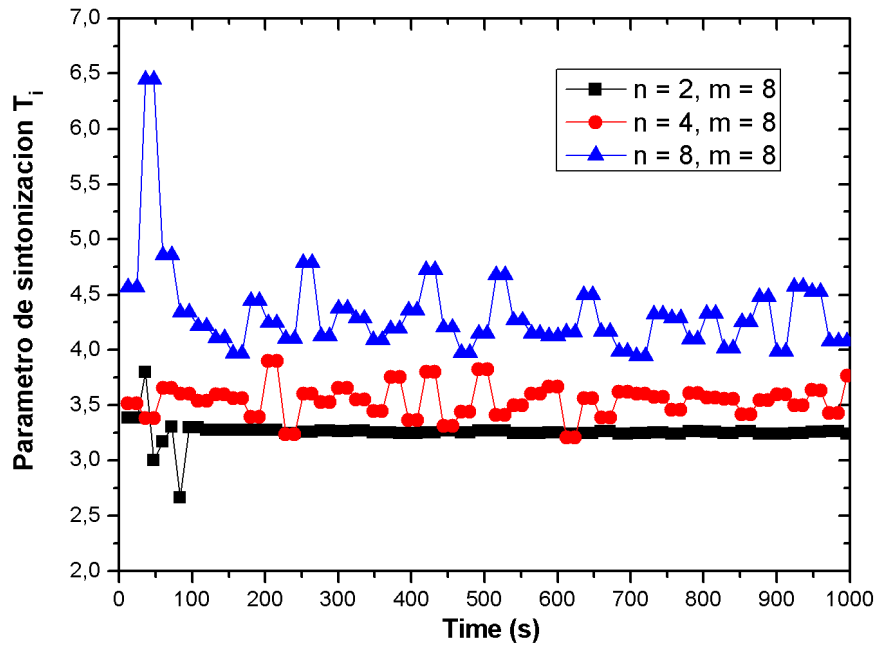
Como ya se comentó anteriormente, el término  $n$  siempre debe ser menor que  $m$ , debido a que se tienen que actualizar más veces los parámetros de sintonización del controlador PID ( $n$ ) que el incremento de la matriz de pesos ( $m$ ), para evaluar cómo evolucionan los parámetros de sintonización manteniendo la matriz de pesos fija durante un tiempo, aunque también se ilustrará un ejemplo en el que  $n$  y  $m$  valgan lo mismo, observando cómo se comporta la red en ese caso.

Por tanto, estas pruebas se han hecho para una normalización de  $10^9$ , y para unos valores de  $\alpha$  y  $\eta$  igual a 0.1, que como se ha visto en todos los apartados anteriores, son bastante adecuados. En concreto, las pruebas que se han hecho, muestran la evolución en tiempo real de los parámetros de sintonización del controlador PID para un  $m$  fijo de

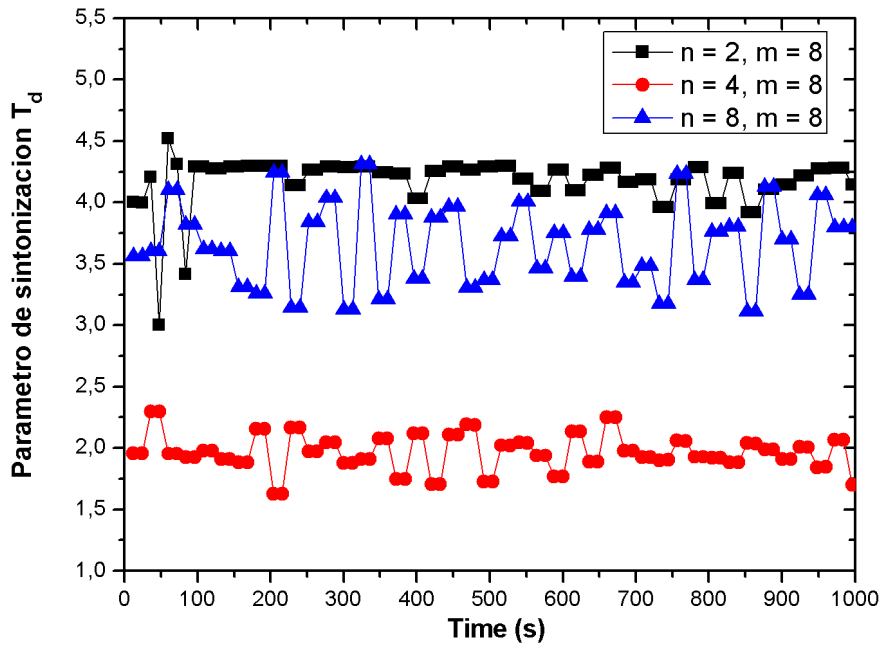
valor 8, y variando  $n$  entre 2, 4 y 8, tal y como se ilustra en las Figuras 29, 30 y 31 para los parámetros  $K_p$ ,  $T_i$  y  $T_d$ , respectivamente.



**Figura 29:** Evolución del parámetro  $K_p$  con  $m = 8$  y diferentes valores de  $n$  (2, 4 y 8)

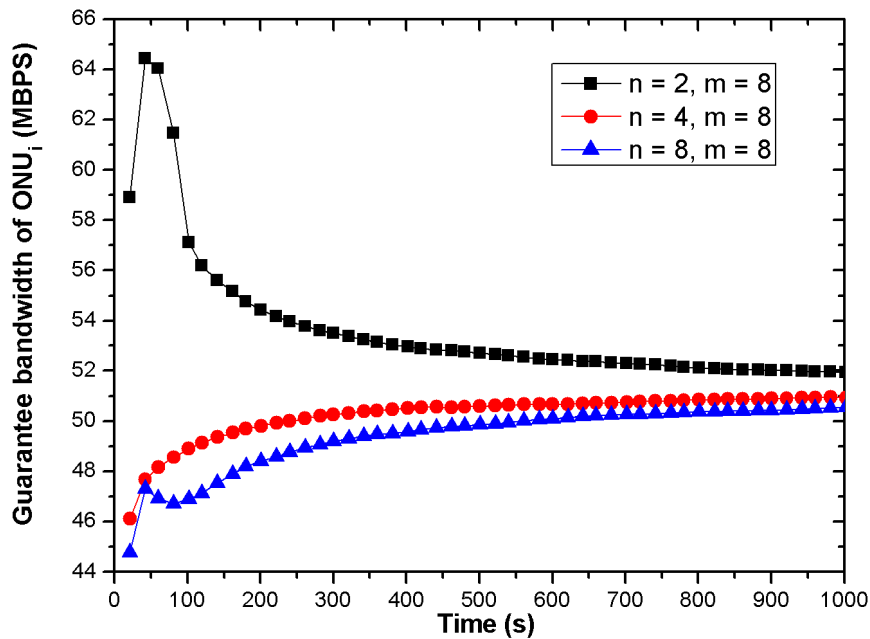


**Figura 30:** Evolución del parámetro  $T_i$  con  $m = 8$  y diferentes valores de  $n$  (2, 4 y 8)



**Figura 31:** Evolución del parámetro  $T_d$  con  $m = 8$  y diferentes valores de  $n$  (2, 4 y 8)

En cuanto a la evolución en tiempo real del ancho de banda garantizado, se muestra en la Figura 32 para el  $SLA_2$  para ver cómo afecta la variación de estos parámetros.

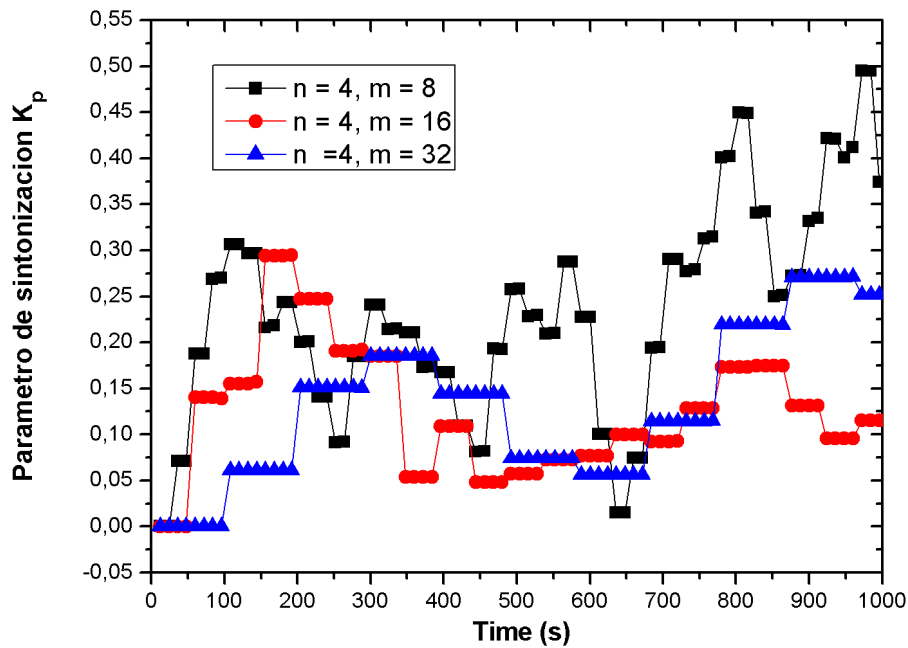


**Figura 32:** Ancho de banda garantizado para el  $SLA_2$  con  $m = 8$  y diferentes valores de  $n$  (2, 4 y 8)

A vista de la Figura 29, se puede comentar que se obtienen unos mejores parámetros de sintonización del controlador PID para los valores de  $n=4$  y  $8$  y  $m=8$ , ya que se observa que para el caso de  $n=2$  y  $m=8$ , el parámetro de sintonización  $K_p$  es muy elevado. Esto se debe a que si tenemos una  $n$  pequeña, se actualizan muy rápidamente los valores de los parámetros de sintonización del PID, lo cual no es conveniente, ya que la red se tiene que adaptar a estos nuevos valores.

Mientras, en la Figura 32, se ve perfectamente que se alcanzan unos niveles adecuados del ancho de banda garantizado cuando se varía  $n$  entre 4 y 8, manteniendo fijo  $m$  a 8, pero no así para el caso en que  $n=2$  y  $m=8$ , donde se obtienen unos anchos de banda garantizados un poco por encima de los valores deseados. Por tanto, podemos concluir que, descartada la situación con  $n=2$  y  $m=8$  por todo lo comentado anteriormente tanto en relación a los parámetros de sintonización como del ancho de banda garantizado, y al ver que la convergencia para el caso  $n=8$  y  $m=8$  es más lenta, la situación en la que los parámetros de sintonización del PID valen  $n=4$  y  $m=8$ , es la que obtiene una adecuación al ancho de banda más coherente y estable.

Por otro lado, en cuanto al valor del parámetro  $m$ , que se refiere a los incrementos de la matriz de pesos, también tiene que ser lo suficientemente grande para que el sistema tenga tiempo de adaptarse, pero tampoco puede alcanzar un valor extremo. Por tanto, para mostrar esta discusión, se va a representar la evolución en tiempo real de los parámetros de sintonización del PID para un valor fijo de  $n$  ( $n = 4$ ) y diferentes valores de  $m$  crecientes (8, 16 y 32), tal y como se ilustra en las Figuras 33, 34 y 35.



**Figura 33:** Evolución del parámetro  $K_p$  con  $n = 4$  y diferentes valores de  $m$  (8, 16 y 32)

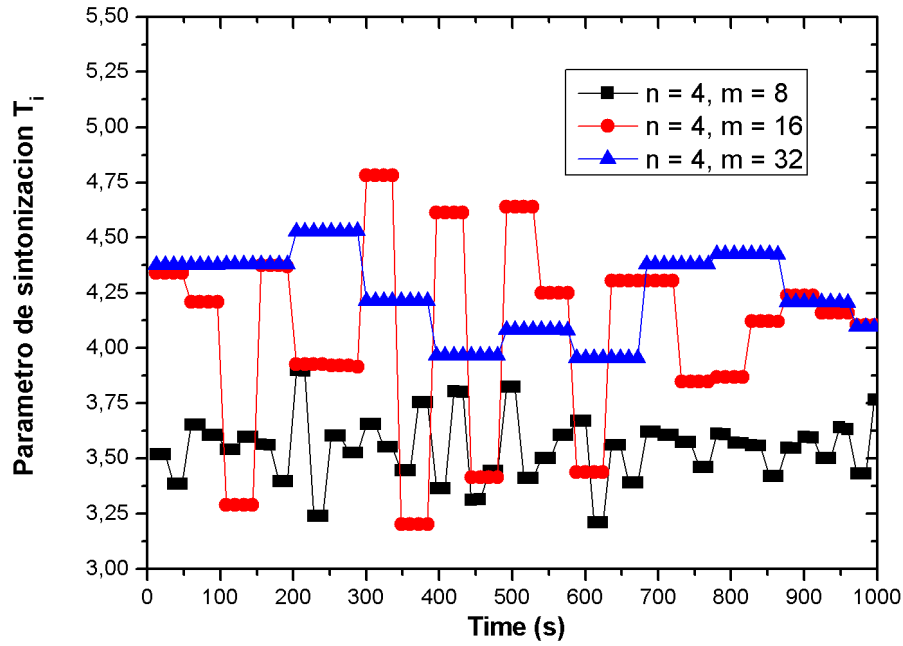


Figura 34: Evolución del parámetro  $T_i$  con  $n = 4$  y diferentes valores de  $m$  (8, 16 y 32)

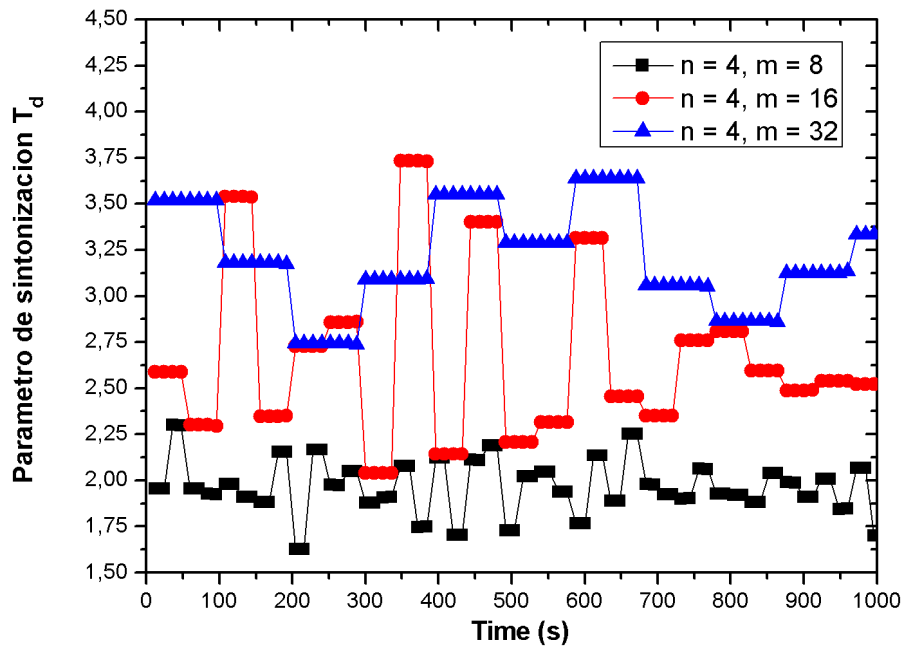
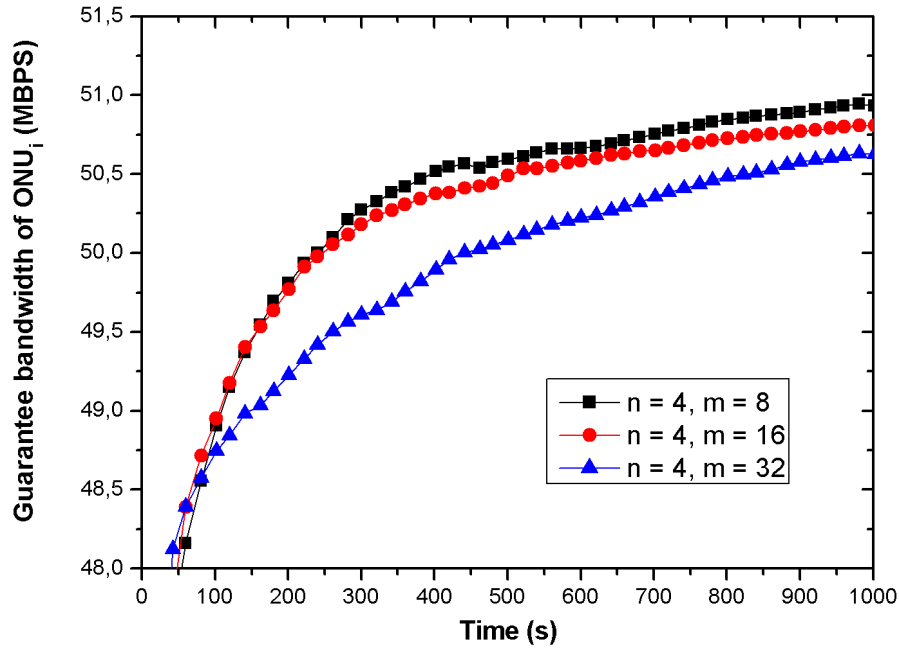


Figura 35: Evolución del parámetro  $T_d$  con  $n = 4$  y diferentes valores de  $m$  (8, 16 y 32)

Para analizar de forma paralela la evolución en tiempo real del ancho de banda garantizado, viendo ahora cómo influye la variación del parámetro  $m$  cuando está fijo  $n$ , se ilustra en la Figura 36 el  $SLA_2$ , como ejemplo.



**Figura 36:** Ancho de banda garantizado para el SLA<sub>2</sub> con  $n = 4$  y diferentes valores de  $m$  (8, 16 y 32)

Este parámetro  $m$  tiene que ser lo suficientemente grande para que el sistema tenga tiempo de adaptar sus parámetros de sintonización manteniendo una matriz de pesos fija durante un tiempo. Por lo tanto, en principio cuanto mayor sea este término, el sistema será más estable, pero sin descuidar que tenga un valor muy extremo, ya que al ir aumentando el valor de  $m$ , los parámetros de sintonización del PID ( $K_p$ ,  $T_i$  y  $T_d$ ) que se mostraban en las Figuras 33, 34 y 35, tienen un carácter más oscilante, debido a que al incrementar la matriz de pesos, se mantienen los valores un mayor tiempo y pueden llegar a actualizarse los parámetros muy despacio.

En cuanto a la evolución en tiempo real del ancho de banda garantizado, que se mostraba en la Figura 36 para el SLA<sub>2</sub>, se ve perfectamente que se alcanzan unos niveles de forma más estable y adecuada para los valores de  $m$  más bajos.

Por lo que en conclusión, viendo las gráficas anteriores, se elige un valor para el parámetro  $m=8$ , ya que como se ha comentado antes, si se elige un valor demasiado alto, los parámetros de sintonización del PID tienden a tener valores más fluctuantes (Figuras 33, 34 y 35). Así pues, unos valores adecuados de estos dos parámetros para diseñar la red neuronal serían  $n=4$  y  $m=8$ .



## 4.5 Conclusiones

En este capítulo, se han realizado varias modificaciones en el algoritmo PID previamente desarrollado, el cual consta de una Red Neuronal que, a partir de los errores cometidos en la asignación de ancho de banda a lo largo del tiempo, ofrece unas variables de sintonización óptimas para cada situación de la red en tiempo real.

En la primera parte se describieron las características de cómo estaba implementada la Red Neuronal y las modificaciones necesarias que debíamos realizar para añadirle las señales de “*bias*”. Para ello fue necesario buscar bastantes referencias bibliográficas, y después plasmar estos cambios en el código del que ya disponíamos.

Por último, se fue analizando el impacto de ciertos parámetros de la Red Neuronal, tales como la normalización del error, los factores  $n$  y  $m$  y los valores de inercia  $\alpha$  y  $\eta$ , para comprobar el funcionamiento de dicha red.

Como conclusión, se podría indicar que la Red Neuronal diseñada, con las señales de “*bias*” añadidas, da unos buenos resultados y es una solución correcta. Además se comprobó que el algoritmo modificado es capaz de ir adaptando los parámetros de sintonización para conseguir una asignación de ancho de banda óptima en función de la carga de la red.

# 5

## Implementación, simulación y validación de un algoritmo DBA TWDM

### 5.1 Introducción

Este capítulo se centra en la implementación de un algoritmo de asignación dinámica de ancho de banda en una red híbrida TWDM-PON (*Time Wavelength Division Multiple Passive Optical Networks*) en el simulador de redes del que ya se disponía en el Grupo de Comunicaciones Ópticas (GCO). Se va a indicar paso a paso la descripción, implementación y validación de este algoritmo en una red PON con múltiples longitudes de onda.

A lo largo del capítulo se expondrán las distintas simulaciones realizadas y se presentarán los diversos escenarios para cada simulación efectuada, ya que cada uno se configura individualmente y está relacionado con diversos parámetros de la red.

En concreto, se analizarán con especial atención los parámetros de ancho de banda total aprovechado (*throughput*) y ancho de banda aprovechado por ONU para cada uno de los distintos niveles de abonado de los que se dispone.

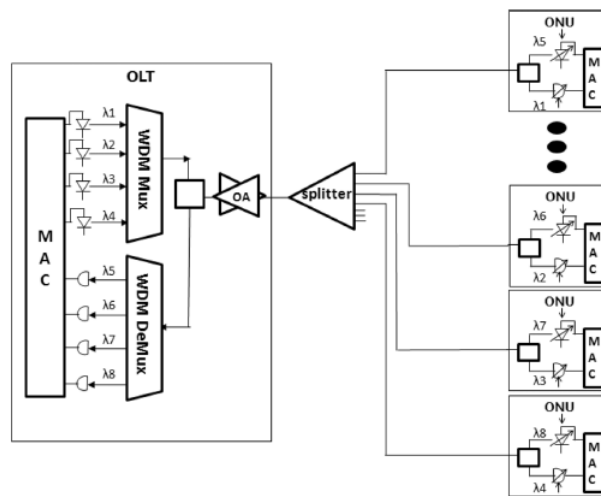
### 5.2 Estado del arte y arquitecturas del estándar NGPON2

En este apartado del capítulo se va a realizar una exposición del estado del arte sobre la próxima generación de redes PON basadas en el estándar NGPON2 (*Next Generation Passive Optical Network 2*). Sobre esta arquitectura de red se implementará un algoritmo de asignación de ancho de banda híbrido TWDM, esto es, usando multiplexación simultánea en tiempo y longitud de onda.

Este estándar se definió en 2015 [39] y fue desarrollado por la UIT (*Unión Internacional de Telecomunicaciones*). Se basa en una arquitectura de múltiples longitudes de onda, punto a multipunto, donde esta nueva tecnología proporciona una velocidad de datos mayor y tiene la capacidad de ahorrar una cantidad de energía significativa, además de acumular 4 arquitecturas XG-PON (10 Gbps PON) utilizando WDM (4 longitudes de onda según el estándar), y aumentar la capacidad de subida y de bajada. Las longitudes de onda utilizadas por este estándar incluyen desde 1596 nm a 1603 nm en la dirección de bajada y desde 1524 nm a 1544 nm en la dirección de subida [40].

Con estas redes se alcanza una tasa agregada de 40 Gbps, es decir, una velocidad 4 veces mayor de la que se conseguía con las tecnologías anteriores. Cada longitud de onda de subida/bajada es capaz de proporcionar un ancho de banda simétrico de hasta 10 Gbit/s a cada abonado si el canal no está multiplexado en el tiempo entre varias ONUs.

Una arquitectura típica de una red de acceso TWDM-PON basada en el estándar NGPON2, se muestra en la Figura 37. Cada ONU utiliza dinámicamente las cuatro longitudes de onda de subida y las cuatro de bajada, mediante un transmisor sintonizable y un receptor también sintonizable situados en el OLT, tal y como sucede en toda la bibliografía consultada a este respecto en [41-49], donde se hace uso de este tipo de receptores y transmisores. En la dirección de bajada, se utiliza el receptor sintonizable para seleccionar el canal de longitud de onda adecuado, mientras que en la dirección de subida, el transmisor se sintoniza para emitir en el canal de longitud de onda deseado [40]. La distancia típica entre el terminal de línea óptica (OLT) y una unidad de red óptica (ONU) es de 40 km, y una OLT es capaz de soportar hasta 256 ONUs [50].



**Figura 37:** Arquitectura típica TWDM-PON

La arquitectura requiere la multiplexación por división en el tiempo y longitud de onda (TWDM) en las direcciones de subida y de bajada y el acceso múltiple por división en el tiempo (TDMA). Cada ONU filtra las tramas en el sentido descendente y procesa sólo las que pertenecen a ella misma. En la dirección de subida, el OLT realiza una asignación dinámica de longitud de onda y de ancho de banda (DWBA, *Dynamic Wavelength and Bandwidth Allocation*) para asignar dinámicamente una longitud de onda de subida y un intervalo de transmisión a cada ONU para que no se solapen [51].

Debido a la potencialidad que presenta este estándar a nivel de despliegue futuro, numerosos estudios de investigación se están centrando en el diseño y desarrollo de algoritmos que contemplen una asignación eficiente del ancho de banda disponible, esto es, teniendo en cuenta la asignación dinámica del ancho de banda y de las longitudes de onda. Así pues, entre estos trabajos nos podemos encontrar con algunos tales como [52], que se concentra en mejorar el rendimiento de una red WDM/TDM PON de largo alcance (LR, *Long Reach*) mediante la introducción de algoritmos MAC multihilo, conocidos como EFT-VF (*Earliest Finish Time with Void Filling*) y EFT-parcial-VF (*Earliest Finish Time partial with Void Filling*), para lograr una asignación dinámica de ancho de banda y longitud de onda eficientes. Más recientemente, en [53] se propone un algoritmo de asignación dinámica de ancho de banda y longitud de onda, pero para arquitecturas TWDM-PON compatibles con NGPON2. Para ahorrar en consumo de energía y costes de implementación, [54] propone un TWDM, multi-PON, proyectado para reprogramar la relocalización de longitud de onda contando en longitudes de onda compartidas por todas las ramas TWDM-PON conectadas. Alternativamente en [55], se ha desarrollado un modelo de optimización de la energía basado en la adaptación del número de longitudes de onda activas en la red y haciendo una comparación en términos de ahorro de energía entre reconfiguración y equilibrio de carga frente reconfiguración. Recientemente, el trabajo presentado en [56] propone un nuevo algoritmo DBA en TWDM-PON que reduce la frecuencia de ajuste de la longitud de onda mejorando la utilización del canal y reduciendo el retraso medio del paquete. Además, los autores de [57] desarrollaron un algoritmo de asignación de longitud de onda y de ancho de banda basado en la predicción lineal (LP, *Linear Polarization*) para asignar dinámicamente recursos en TWDM-PON para reducir el retraso medio extremo a extremo. El algoritmo propuesto en [58] explora la sintonización y las capacidades de optimización de una unidad de red óptica de 10 Gbps de superficie que emite superficies (10G-VCSEL-ONU) para mejorar los ahorros de energía en el OLT y las ONUs, respectivamente. Mientras, en [59] se propone un marco de ancho de banda dinámico y de asignación de longitud de onda dinámico que utilice el número óptimo de longitudes de onda y el modo suspensión para TWDM-PON con limitación de retardo con ONUs sintonizables de 10Gbps-SC-VCSEL. En el trabajo de [60] se propone una política de umbral adaptativa que reduce el impacto de la reconfiguración en el marco del retardo promedio con penalización sobre la eficiencia energética. Finalmente, en [61] se propone un esquema de asignación de longitud de onda dinámica y ranuras de tiempo para un TWDM-PON basado en la clasificación de prioridad inter-ONU.

El hecho de que este tipo de redes TWDM-PON trabajen típicamente con láseres sintonizables tanto en el receptor como en el transmisor [41-49], implica que cuando un transmisor cambia su longitud de onda, se requiere un tiempo de conmutación (tiempo de *tuning*) en el que no puede transmitirse ningún paquete, por lo que este tiempo de conmutación disminuye la utilización del ancho de banda y aumenta la sobrecarga de transmisión. Para saber de qué valores estamos hablando para este tiempo de *tuning*, realizamos una profunda búsqueda bibliográfica de la que vamos a ir extrayendo posibles valores para realizar nuestras pruebas, según lo que se utilice en dichos artículos consultados. En el trabajo presentado en [62] se trabaja con tiempos de conmutación alrededor de los 100  $\mu$ s. Así mismo, en el algoritmo propuesto en [63] se hacen las pruebas con tiempos de sintonización que tienen un valor en torno a los 10  $\mu$ s, y en el trabajo presentado en [64] se realizan las simulaciones para unos tiempos de *tuning* que varían entre 10  $\mu$ s, 50  $\mu$ s y 100  $\mu$ s. Finalmente, el trabajo presentado en [65] utiliza valores que oscilan entre 10  $\mu$ s, 100  $\mu$ s y 1000  $\mu$ s en el esquema de asignación dinámica de ancho de banda desarrollado para una arquitectura TWDM-PON. En base a toda esta información recabada, se ve que los tiempos de *tuning* toman valores del orden de los microsegundos y, en concreto, nosotros realizaremos el estudio de simulación para tiempos de sintonización de 10  $\mu$ s, 50  $\mu$ s y 300  $\mu$ s. De ese modo, se verá cómo influye la variación de este parámetro tanto en el ancho de banda total aprovechado (*throughput*), como en el ancho de banda aprovechado por ONU para cada uno de los distintos perfiles de abonado de la red.

### 5.3 Implementación del algoritmo TWDM

En este apartado se va a mostrar cómo se ha realizado la implementación del algoritmo TWDM en nuestro simulador, prestando especial atención a cómo se realiza la asignación del ancho de banda y de las longitudes de onda.

#### 5.3.1 Asignación de Ancho de banda

La asignación del ancho de banda a cada ONU<sub>i</sub> ( $B_{\text{alloc}}^{\text{onu}_i}$ ) se realiza de forma dinámica en la capa MAC del OLT cuando llega un mensaje *Report* procedente de una ONU, mediante una política de *polling* con un esquema limitado, que se define como  $B_{\text{alloc}}^{\text{onu}_i} = \text{Mínimo} \{B_{\text{required}}^{\text{onu}_i}, B_{\text{max}}^{\text{onu}_i}\}$ , donde  $B_{\text{required}}^{\text{onu}_i}$  es el ancho de banda requerido por la ONU<sub>i</sub> y  $B_{\text{max}}^{\text{onu}_i}$  es el ancho de banda máximo permitido para cada ONU.

En cuanto al ancho de banda máximo que se puede asignar a cada ONU dependiendo del SLA que tenga asociado, indicar que a diferencia de lo que ocurría en la parte descrita para la Red Neuronal en el capítulo anterior, aquí no varía en cada ciclo, y se define según lo que indica la Ecuación 19. En dicha ecuación,  $B_{\text{cycle\_available}}$  es el ancho de banda disponible en cada ciclo máximo,  $N_{\text{onus}}^{\text{sla}_j}$  representa el número de ONUs asociadas

al  $SLA_j$  en la red y el término  $W^{sla_k / onu_i \in sla_k}$  se refiere al peso asociado a cada  $SLA_k$  contratado por la  $ONU_i$ .

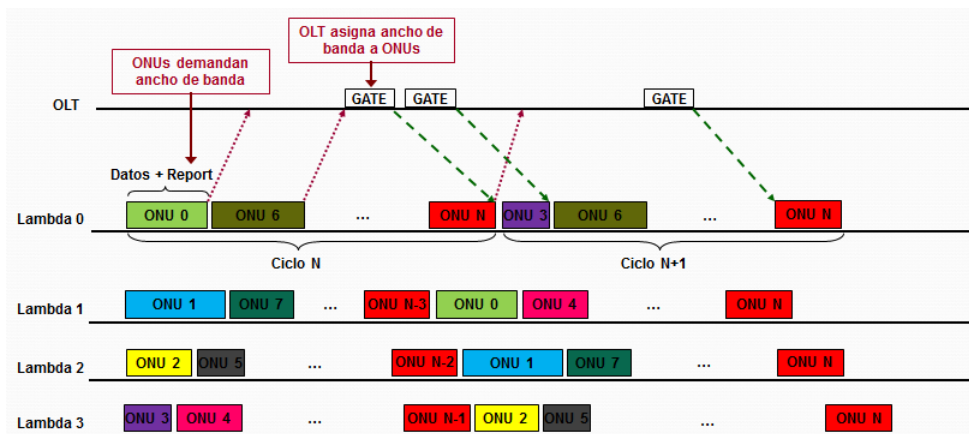
$$B_{\max}^{onu_i} = \frac{B_{\text{cycle available}} \cdot W^{sla_k / onu_i \in sla_k}}{\sum_j W^{sla_j} \cdot N_{\text{onus}}^{sla_j}} \quad (19)$$

### 5.3.2 Asignación de longitudes de onda en cada ciclo

Otro aspecto importante a tener en cuenta es cómo asignar las longitudes de onda que se van a utilizar en cada ciclo. Para ello, indicar que en nuestro algoritmo TWDM, se transmitirá por la longitud de onda que primero quede libre (de las cuatro disponibles), para aprovechar al máximo la capacidad de los canales.

Este algoritmo está basado en el principio de que la primera longitud de onda que queda libre será la asignada a la siguiente ONU que transmite, de modo que se pueda aprovechar al máximo el ancho de banda disponible en todas las longitudes de onda. Para ello, se tendrán que comparar los distintos instantes de tiempo en los que todas las longitudes de onda quedan libres, y quedarnos con el instante menor de todos, que determinará la longitud de onda por la que transmitirá la siguiente ONU, denominando  $t_{tx}$  al tiempo de transmisión de la longitud de onda que primero ha quedado libre.

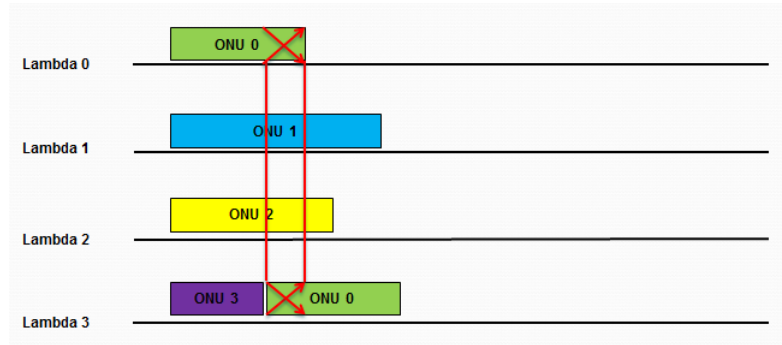
Para explicar cómo se realiza esta asignación, se utiliza la Figura 38, donde en primer lugar, como ya se indicó anteriormente, cada ONU va a transmitir por la primera longitud de onda que esté libre, por lo que inicialmente, la  $ONU_0$  va a transmitir por la longitud de onda  $\lambda_0$ , la  $ONU_1$  lo va a hacer por la longitud de onda  $\lambda_1$ , la  $ONU_2$  lo va a hacer por la longitud de onda  $\lambda_2$  y, por último, la  $ONU_3$  lo hará por la longitud de onda  $\lambda_3$ .



**Figura 38:** Representación del funcionamiento del algoritmo para la asignación de las longitudes de onda

Por otro lado, también hay que tener en cuenta que una ONU no puede transmitir por dos longitudes de onda al mismo tiempo en dos ciclos distintos, ya que estamos usando láseres sintonizables. Para ver más claramente esta situación, se observa la Figura 39,

donde se hace la descripción para un caso sencillo, utilizando la hipótesis de sólo disponer de 4 longitudes de onda y 4 ONUs para ver más claramente dicha explicación, donde se ve que la ONU<sub>0</sub>, después de haberse producido el primer ciclo con normalidad, debería transmitir por la longitud de onda  $\lambda_3$  en el siguiente ciclo, ya que es la primera que ha quedado libre en el anterior. Sin embargo, vemos que se produce una colisión, ya que la ONU<sub>0</sub> está transmitiendo a la vez por dos longitudes de onda distintas, en este caso concreto por  $\lambda_0$  y por  $\lambda_3$ , produciendo una situación de colisión que no se puede permitir que ocurra.



**Figura 39:** Representación de una situación de colisión en la asignación de longitudes de onda en cada ciclo entre varias estaciones

Por último, cabe reseñar que como ya se comentó anteriormente, según lo consultado en la bibliografía en [41-49], al utilizar equipos sintonizables tanto para el transmisor como para el receptor, esto conlleva que hay un tiempo de conmutación o tiempo de *tuning*, cuyos valores también se analizaron en [62-65]. Estos tiempos de conmutación hay que tenerlos en cuenta siempre que estos láseres pasen de transmitir en una longitud de onda a otra distinta, ya que degradan el comportamiento de la red, no siendo así cuando se transmite por la misma longitud de onda que se hacía anteriormente.

## 5.4 Resultados

En este apartado del capítulo, se expone el escenario de simulación genérico para la ejecución del algoritmo TWDM, junto con los resultados gráficos obtenidos al realizar las simulaciones.

### 5.4.1 Escenario de simulación genérico

A continuación, en la Tabla 2, se presenta el escenario de simulación genérico con los parámetros que se configuran de manera común para todas las simulaciones.

Parámetros de la simulación	Valores
Número de ONUs	128
Número de longitudes de onda	4
Longitud de la red	40 Kilómetros
Capacidad de la red	10 Gbps
Tasa de transmisión ONU-abonado	1 Gbit/s
Tiempo de guarda	1 microsegundo
Tamaño del paquete	Variable entre 64 y 1500 bytes
Tamaño del buffer	10 MBytes
Tamaño de los paquetes	Trimodal
Método de inserción de paquetes	Método de inserción de prioridad de colas
Método de extracción de paquetes	Método de extracción de colas de prioridad
Número de Streams	32
Número de clases de servicio	3
Número de SLAs	3
Service Level Agreement (SLA) Perfiles de abonado	SLA <sub>0</sub> : 8 ONUs ( $N_{\text{onus}}^{\text{sla}_0} = 8$ ) Ancho de banda garantizado: 500 Mbps ( $B_{\text{alloc}}^{\text{onu}_i \in \text{sla}_0} = 500$ ) Peso: 3 ( $W^{\text{sla}_0} = 3$ )
	SLA <sub>1</sub> : 40 ONUs ( $N_{\text{onus}}^{\text{sla}_1} = 40$ ) Ancho de banda garantizado: 350 Mbps ( $B_{\text{alloc}}^{\text{onu}_i \in \text{sla}_1} = 350$ ) Peso: 2 ( $W^{\text{sla}_1} = 2$ )
	SLA <sub>2</sub> : 80 ONUs ( $N_{\text{onus}}^{\text{sla}_2} = 80$ ) Ancho de banda garantizado: 200 Mbps ( $B_{\text{alloc}}^{\text{onu}_i \in \text{sla}_2} = 200$ ) Peso: 1 ( $W^{\text{sla}_2} = 1$ )

**Tabla 2:** Parámetros de la simulación considerados en el escenario de simulación para TWDM

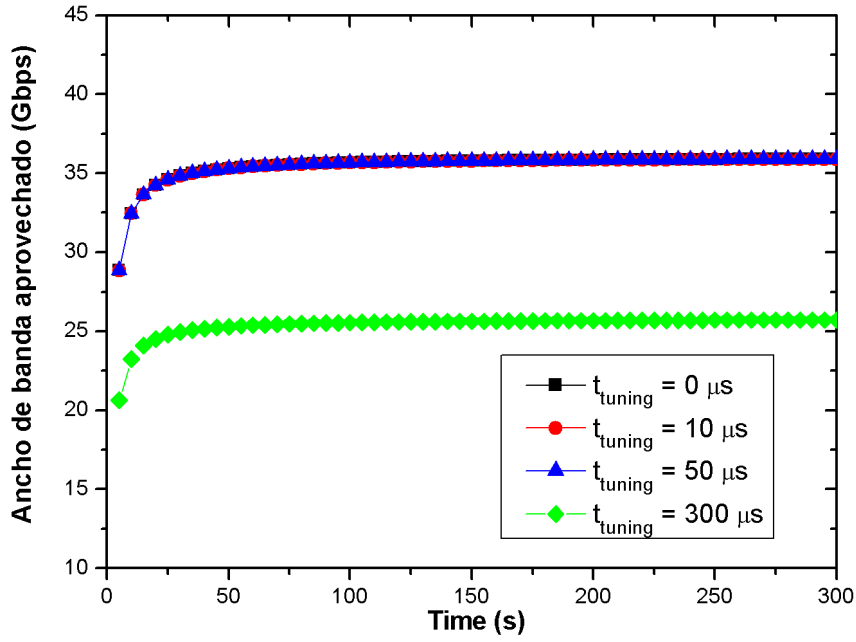
### 5.4.2 Impacto del tiempo de *tuning* para cargas altas

En este caso, se hacen las pruebas para una carga en cada ONU de 0.5 (500 Mbps), ya que se considera que la tasa de transmisión ONU-abonado es de 1 Gbps. Por tanto, el ancho de banda demandado por las 128 ONUs en la red será de  $0.5 \text{ Gbps} \cdot 128 \text{ ONUs} = 64 \text{ Gbps}$ . Sin embargo, en este tipo de redes tenemos 10 Gbps para cada longitud de onda, lo que hace un total de ancho de banda agregado de 40

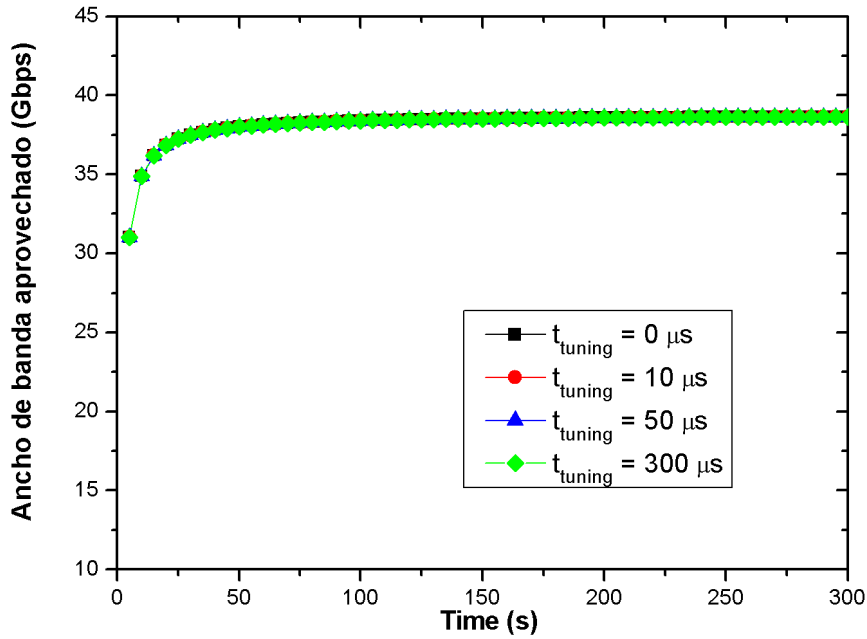


Gbps (al disponer de cuatro longitudes de onda), por lo que no se le va a poder dar todo lo que piden las ONUs.

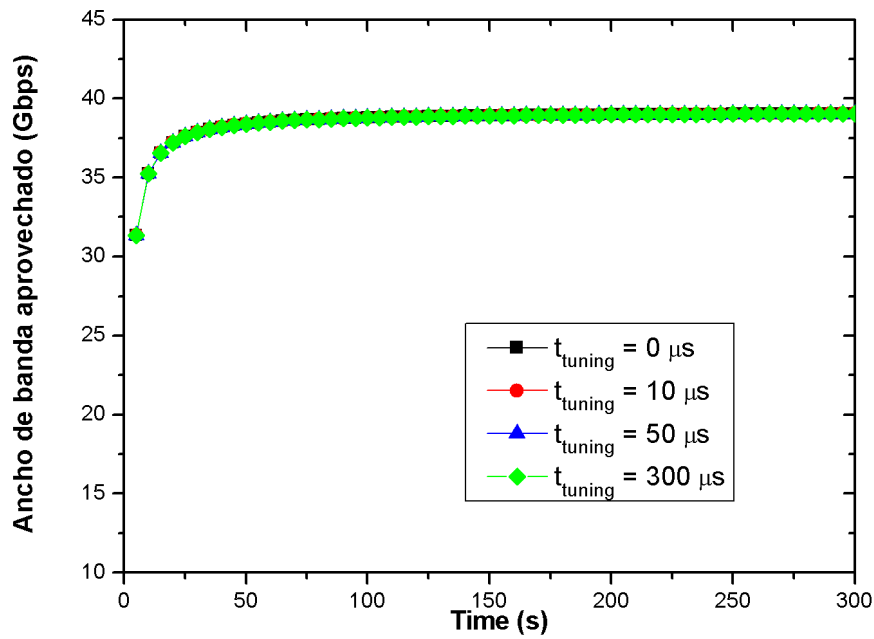
En cuanto a las pruebas realizadas, en primer lugar, se muestra cómo influye la variación del tiempo de *tuning* ( $t_{\text{tuning}} = 0 \mu\text{s}$ ,  $10 \mu\text{s}$ ,  $50 \mu\text{s}$  y  $300 \mu\text{s}$ ) para diferentes tiempos de ciclo máximos, en concreto  $t_{\text{cycle}} = 2 \text{ ms}$  (Figura 40),  $t_{\text{cycle}} = 6.7 \text{ ms}$  (Figura 41) y  $t_{\text{cycle}} = 10 \text{ ms}$  (Figura 42).



**Figura 40:** Evolución del ancho de banda total aprovechado para  $t_{\text{cycle}} = 2 \text{ ms}$  y diferentes valores de  $t_{\text{tuning}}$  ( $t_{\text{tuning}} = 0 \mu\text{s}$ ,  $10 \mu\text{s}$ ,  $50 \mu\text{s}$  y  $300 \mu\text{s}$ )



**Figura 41:** Evolución del ancho de banda total aprovechado para  $t_{\text{cycle}} = 6.7 \text{ ms}$  y diferentes valores de  $t_{\text{tuning}}$  ( $t_{\text{tuning}} = 0 \mu\text{s}$ ,  $10 \mu\text{s}$ ,  $50 \mu\text{s}$  y  $300 \mu\text{s}$ )



**Figura 42:** Evolución del ancho de banda total aprovechado para  $t_{\text{cycle}} = 10 \text{ ms}$  y diferentes valores de  $t_{\text{tuning}}$  ( $t_{\text{tuning}} = 0 \mu\text{s}$ ,  $10 \mu\text{s}$ ,  $50 \mu\text{s}$  y  $300 \mu\text{s}$ )

En la Figura 40, para un tiempo de ciclo máximo de 2 ms, se ve que afecta el tiempo de *tuning* al ancho de banda total aprovechado (40 Gbps), pero solamente cuando este aumenta hasta los 300  $\mu\text{s}$ , donde dicho ancho de banda total aprovechado cae drásticamente. Por el contrario, si se observan tanto la Figura 41, para un tiempo de ciclo máximo de 6.7 ms, como la Figura 42, para un tiempo de ciclo de 10 ms, se ve que no afecta en el ancho de banda total aprovechado si aumentamos el tiempo de *tuning* a cualquiera de los valores con los que estamos trabajando. Esto es así en estos casos, porque el tiempo de ida y vuelta (400  $\mu\text{s}$ ) más el tiempo de *tuning*, siempre será inferior al tiempo de ciclo, luego no impactará en el ancho de banda aprovechado total. Finalmente y a modo de conclusión, a medida que aumenta el tiempo de ciclo, el impacto del tiempo de *tuning* en el aprovechamiento del ancho de banda total disponible es mucho menor, tal y como se puede comprobar a la vista de los resultados que se ilustran en la Tabla 3.

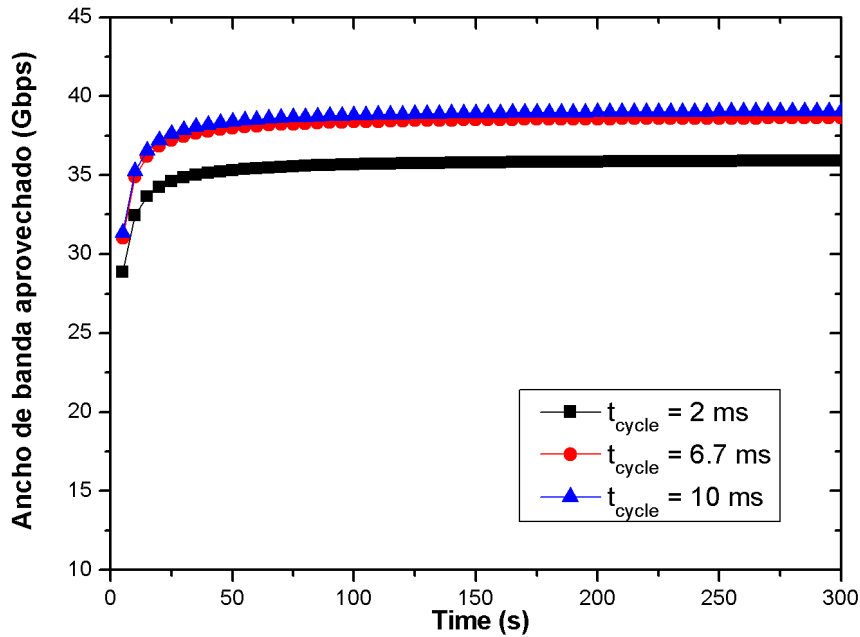
$t_{\text{cycle}}$	$t_{\text{tuning}}$			
	0 $\mu\text{s}$	10 $\mu\text{s}$	50 $\mu\text{s}$	300 $\mu\text{s}$
<b>2 ms</b>	35,9082 Gbps	35,902 Gbps	35,9097 Gbps	25,7066 Gbps
<b>6.7 ms</b>	38,6467 Gbps	38,6461 Gbps	38,6464 Gbps	38,6468 Gbps
<b>10 ms</b>	39,0458 Gbps	39,0455 Gbps	39,0456 Gbps	39,0451 Gbps

**Tabla 3:** Resultados para cargas altas cuando se varía el  $t_{\text{tuning}}$  manteniendo constante el  $t_{\text{cycle}}$

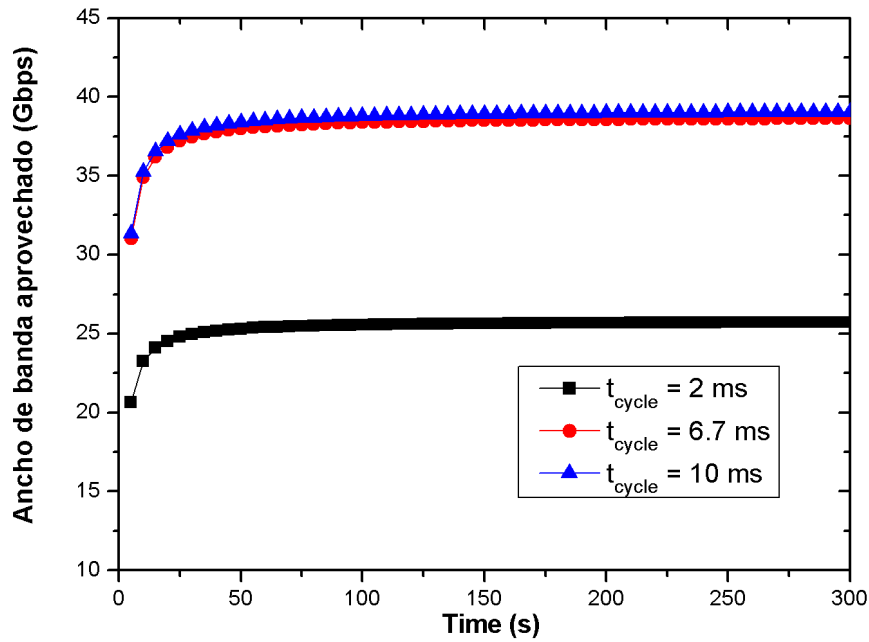
En base a estos valores, y para el tiempo de *tuning* nulo, se observa claramente que la capacidad real de la red alcanza en torno a los 36 Gbps para el tiempo de ciclo de 2 ms, sobre los 38.6 Gbps para el tiempo de ciclo de 6.7 ms, y sobre los 39 Gbps para el tiempo de ciclo de 10 ms, es decir, acercándose prácticamente a la capacidad máxima de

la red (40 Gbps). En este punto, se hicieron varias pruebas para explicar la diferencia de ancho de banda máximo aprovechado total (sobre un máximo de 40 Gbps) entre los obtenidos para los diferentes tiempos de ciclo, que es lo que se pierde en el *overhead*, siendo mayor para el tiempo de ciclo de 2 ms que para los otros dos casos. En concreto, para el tiempo de ciclo de 2 ms, se obtiene un *overhead* de unos 2.77182 Gbps, para el tiempo de ciclo de 6.7 ms es de unos 0.844448 Gbps y para el tiempo de ciclo de 10 ms, se obtienen unos 0.571486 Gbps. Estas diferencias para los distintos tiempos de ciclo, se deben a que para el tiempo de ciclo de 2 ms con 128 ONUs hay 128 tiempos de guarda y 128 mensajes de control *Report* repartidos en dicho tiempo (más el desalineamiento debido a los paquetes de datos Ethernet). Lo mismo sucede para los tiempos de ciclo de 6.7 ms y 10 ms, pero en esos casos, esos tiempos de *overhead* (*guarda* y *Report*) están repartidos en un tiempo de ciclo mayor. Por eso, el *overhead* para el tiempo de ciclo de 10 ms es aproximadamente 5 veces más pequeño que para el tiempo de ciclo de 2 ms, unas 1,5 veces menor que el tiempo de ciclo de 6.7 ms, y este tiempo de ciclo de 6.7 ms tiene un *overhead* aproximadamente 3,35 veces menor que para el tiempo de ciclo de 2 ms, tal y como ocurre a la vista de los valores indicados anteriormente.

El estudio ahora se basa en ir modificando los tiempos de ciclo, mientras se deja constante el tiempo de *tuning*, tal y como se ilustra en las Figuras 43 y 44.



**Figura 43:** Evolución del ancho de banda total aprovechado para  $t_{\text{tuning}} = 0 \mu\text{s}$ ,  $10 \mu\text{s}$  o  $50 \mu\text{s}$  y diferentes valores de  $t_{\text{cycle}}$  ( $t_{\text{cycle}} = 2 \text{ ms}$ ,  $6.7 \text{ ms}$  y  $10 \text{ ms}$ )



**Figura 44:** Evolución del ancho de banda total aprovechado para  $t_{tuning} = 300 \mu s$  y diferentes valores de  $t_{cycle}$  ( $t_{cycle} = 2 \text{ ms}$ ,  $6.7 \text{ ms}$  y  $10 \text{ ms}$ )

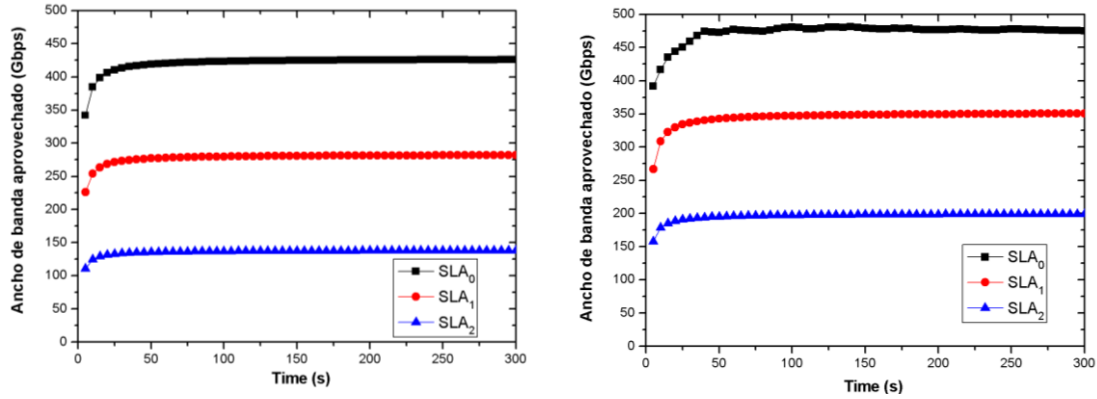
En primer lugar, indicar que para no recargar demasiado la memoria, en la Figura 43 solamente se muestra el ancho de banda total aprovechado para un tiempo de  $t_{tuning}$  constante de  $0 \mu s$  mientras se varía el tiempo de ciclo, ya que para unos tiempos de  $t_{tuning}$  constantes de  $10 \mu s$  y  $50 \mu s$  sucedería exactamente lo mismo. Por otro lado, cuando el tiempo de  $t_{tuning}$  aumenta a  $300 \mu s$  (Figura 44) se ve que afecta más cuanto menor es el tiempo de ciclo, es decir, afecta bastante para un tiempo de ciclo de  $2 \text{ ms}$ , pero luego ya no lo hace ni para el de  $6.7 \text{ ms}$  ni para el de  $10 \text{ ms}$ , tal y como se comprueba con los valores obtenidos que se muestran en la Tabla 4.

$t_{tuning}$	$t_{cycle}$		
	2 ms	6.7 ms	10 ms
0 $\mu s$	35,9082 Gbps	38,6467 Gbps	39,0458 Gbps
10 $\mu s$	35,902 Gbps	38,6461 Gbps	39,0455 Gbps
50 $\mu s$	35,9097 Gbps	38,6464 Gbps	39,0456 Gbps
300 $\mu s$	25,7066 Gbps	38,6468 Gbps	39,0451 Gbps

**Tabla 4:** Resultados para cargas altas cuando se varía el  $t_{cycle}$  manteniendo constante el  $t_{tuning}$

A continuación, se muestra el ancho de banda aprovechado por cada ONU, para constatar que lo asigna correctamente en base a los anchos de banda garantizados para cada SLA y definidos en el escenario de simulación genérico (Tabla 2). En este caso, como se vio que no se iba a poder dar todo lo que se pedía en función de las cuentas teóricas realizadas al comienzo de este apartado, sí que van a tener influencia los distintos pesos para cada uno de los 3 SLAs, tal y como se observa en la Figura 45 (Imagen de la izquierda), donde se muestra el caso más restrictivo, es decir, para el tiempo de ciclo de  $2 \text{ ms}$  y con tiempo de  $t_{tuning}$  igual a  $300 \mu s$ , por lo que se van a obtener unos valores ligeramente por debajo de los esperados, que por tanto no cumplen

los límites definidos para cada SLA. En cambio, ya para el tiempo de *tuning* de 50  $\mu$ s, según se ve en la Figura 45 (Imagen de la derecha), vemos que sí se cumplen los anchos de banda garantizados establecidos para cada SLA, ya que, el SLA<sub>0</sub> obtiene un ancho de aproximadamente 500 Mbps, el SLA<sub>1</sub> sobre los 350 Mbps y el SLA<sub>2</sub> de 200 Mbps, que coincide con el ancho de banda mínimo que el algoritmo debe garantizar a cada uno de los SLAs de la red de acceso.

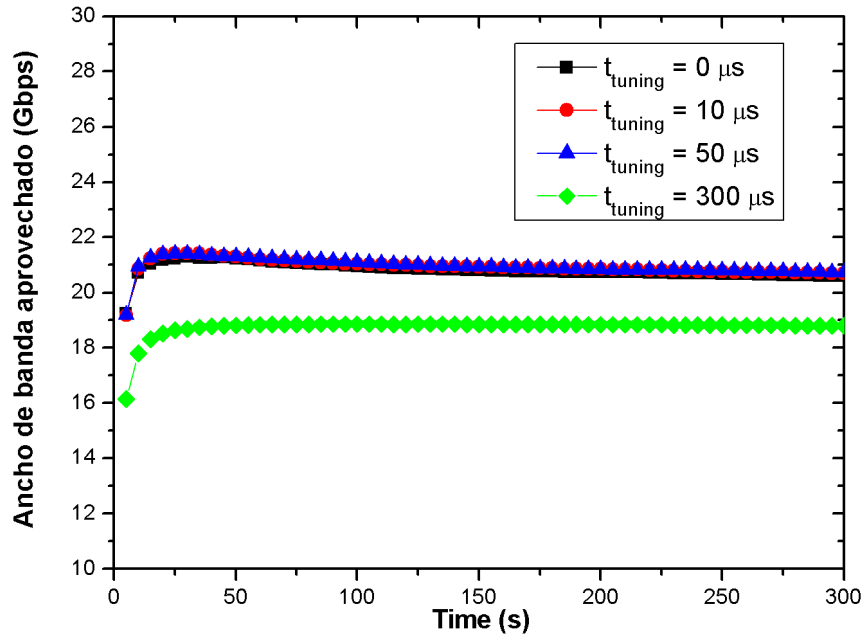


**Figura 45:** Evolución del ancho de banda aprovechado por ONU para cargas altas cuando  $t_{cycle} = 2$  ms y  $t_{tuning} = 300$   $\mu$ s y 50  $\mu$ s

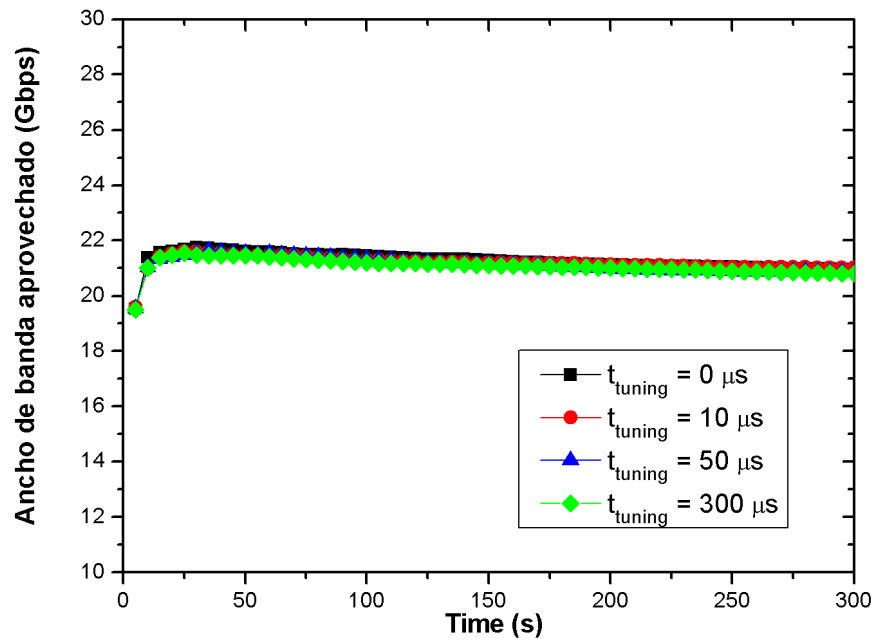
### 5.4.3 Impacto del tiempo de *tuning* para cargas bajas

En este apartado, se va a analizar el impacto de los tiempos de conmutación cuando la carga de red total es relativamente baja. Para ello, se han hecho pruebas para una carga de ONU de 0.2 (200 Mbps), por lo que la demanda total de todas las ONUs en la red será de  $0.2 \text{ Gbps} \cdot 128 \text{ ONUs} = 25.6 \text{ Gbps}$ , relativamente menor a la capacidad total de 40 Gbps de la red. Por lo tanto, en este caso, el algoritmo sí que va a poder asignar todo lo que se pidan las ONUs (e incluso más).

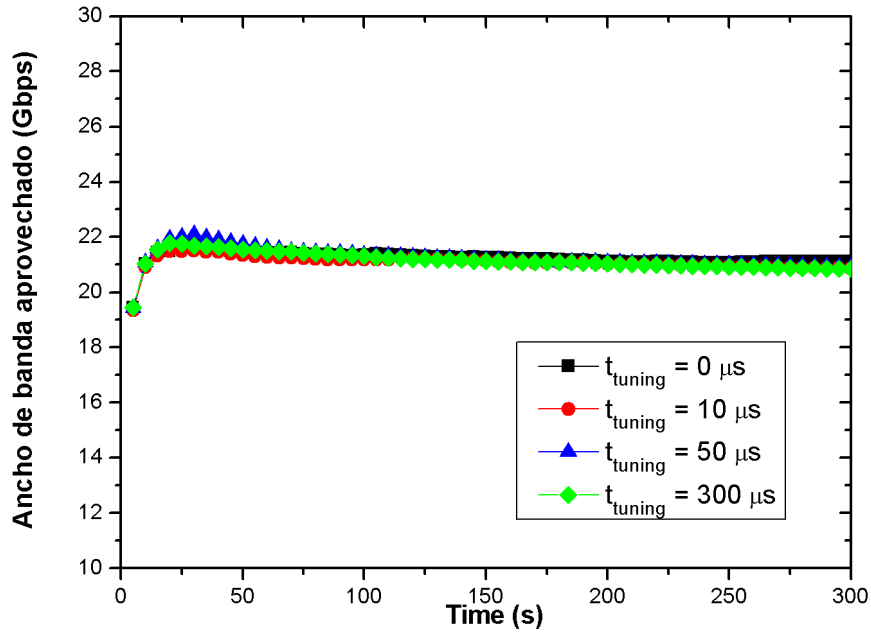
Al igual que en el análisis anterior, en primer lugar, se muestra en las Figuras 46, 47 y 48 cómo influye la variación del tiempo de *tuning* para diferentes tiempos de ciclo máximos, sobre el ancho de banda total aprovechado.



**Figura 46:** Evolución del ancho de banda total aprovechado para  $t_{cycle} = 2 \text{ ms}$  y diferentes valores de  $t_{tuning}$  ( $t_{tuning} = 0 \text{ μs}$ ,  $10 \text{ μs}$ ,  $50 \text{ μs}$  y  $300 \text{ μs}$ )



**Figura 47:** Evolución del ancho de banda total aprovechado para  $t_{cycle} = 6.7 \text{ ms}$  y diferentes valores de  $t_{tuning}$  ( $t_{tuning} = 0 \text{ μs}$ ,  $10 \text{ μs}$ ,  $50 \text{ μs}$  y  $300 \text{ μs}$ )



**Figura 48:** Evolución del ancho de banda total aprovechado para  $t_{\text{cycle}} = 10 \text{ ms}$  y diferentes valores de  $t_{\text{tuning}}$  ( $t_{\text{tuning}} = 0 \mu\text{s}$ ,  $10 \mu\text{s}$ ,  $50 \mu\text{s}$  y  $300 \mu\text{s}$ )

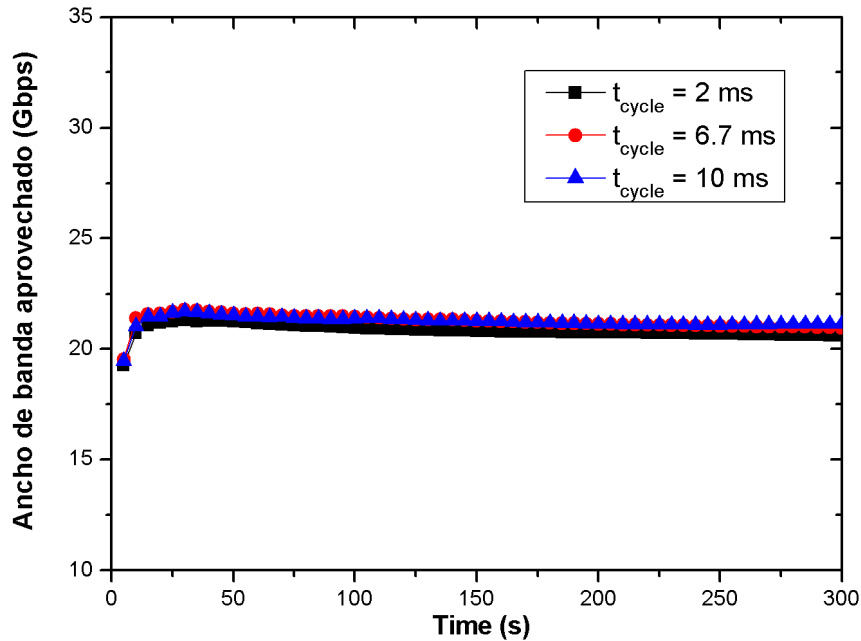
Se observa que para cargas bajas, el impacto del tiempo de *tuning* sobre el ancho de banda desperdiciado es mucho menor. En concreto, en la Figura 46, para un tiempo de ciclo constante de 2 ms, se ve que afecta solamente el tiempo de *tuning* de 300  $\mu\text{s}$ , y en menor medida que lo hacía para cargas altas. Mientras, en la Figura 47, para un tiempo de ciclo máximo de 6.7 ms, y en la Figura 48, para un tiempo de ciclo de 10 ms, se observa que ya no afecta ninguno de los tiempos de conmutación, tal y como sucedía también para cargas altas. Para ver con números concretos cómo afecta la variación de este tiempo de *tuning* cuando se mantiene constante el tiempo de ciclo, se muestra en la Tabla 5 un resumen con todos los valores obtenidos:

$t_{\text{cycle}}$	$t_{\text{tuning}}$			
	0 $\mu\text{s}$	10 $\mu\text{s}$	50 $\mu\text{s}$	300 $\mu\text{s}$
2 ms	20,5948 Gbps	20,6961 Gbps	20,7388 Gbps	18,7955 Gbps
6.7 ms	20,9607 Gbps	21,0175 Gbps	20,8538 Gbps	20,8015 Gbps
10 ms	21,1315 Gbps	20,8705 Gbps	20,9179 Gbps	20,8436 Gbps

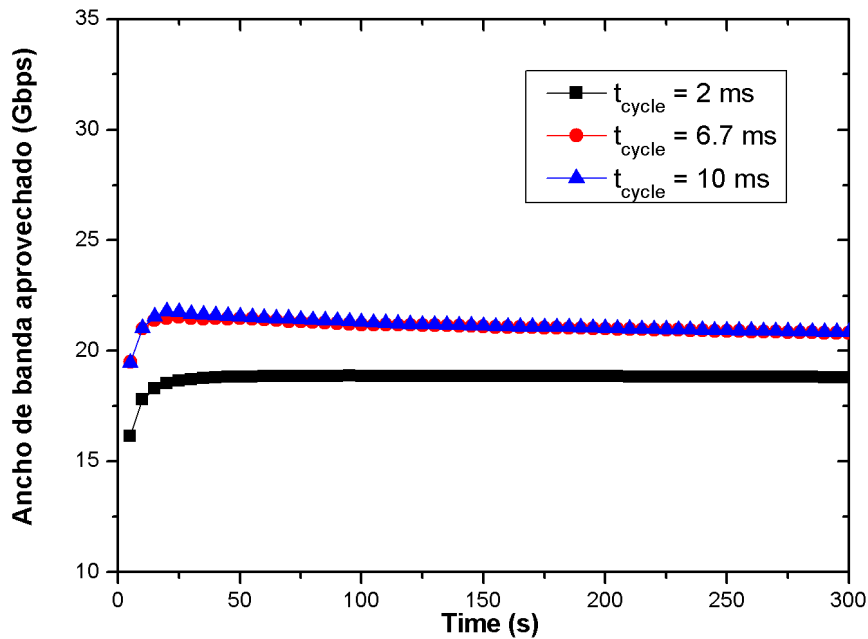
**Tabla 5:** Resultados para cargas bajas cuando se varía el  $t_{\text{tuning}}$  manteniendo constante el  $t_{\text{cycle}}$

En función de estos valores, y para el tiempo de *tuning* nulo, se hicieron también varias pruebas para explicar la diferencia de ancho de banda máximo aprovechado entre los obtenidos para los diferentes tiempos de ciclo. En este caso, se obtiene un comportamiento similar para los distintos tiempos de ciclo, esto es, se da todo el ancho de banda que piden, salvo una pequeña cantidad que se pierde en el *overhead*. Se comprueba que, en el análisis para cargas bajas, influye menos este *overhead*, ya que es prácticamente igual para los tres tiempos de ciclo distintos, y vale sobre los 3.36293 Gbps.

Ahora se hace el estudio en base a ir modificando los tiempos de ciclo, mientras se deja constante el tiempo de *tuning*, tal y como se ilustra en las Figuras 49 y 50.



**Figura 49:** Evolución del ancho de banda total aprovechado para  $t_{\text{tuning}} = 0 \mu\text{s}$ ,  $10 \mu\text{s}$  o  $50 \mu\text{s}$  y diferentes valores de  $t_{\text{cycle}}$  ( $t_{\text{cycle}} = 2 \text{ ms}$ ,  $6.7 \text{ ms}$  y  $10 \text{ ms}$ )



**Figura 50:** Evolución del ancho de banda total aprovechado para  $t_{\text{tuning}} = 300 \mu\text{s}$  y diferentes valores de  $t_{\text{cycle}}$  ( $t_{\text{cycle}} = 2 \text{ ms}$ ,  $6.7 \text{ ms}$  y  $10 \text{ ms}$ )

En la Figura 49 se observa que cuando el tiempo de tuning varía entre 0, 10 o 50  $\mu\text{s}$ , el comportamiento es el mismo y se da a todas las ONUs el ancho de banda demandado. En cambio, en la Figura 50, cuando el tiempo de *tuning* es de 300  $\mu\text{s}$ , se ve que afecta ligeramente para el tiempo de ciclo más pequeño, es decir, el de 2 ms, mientras que para

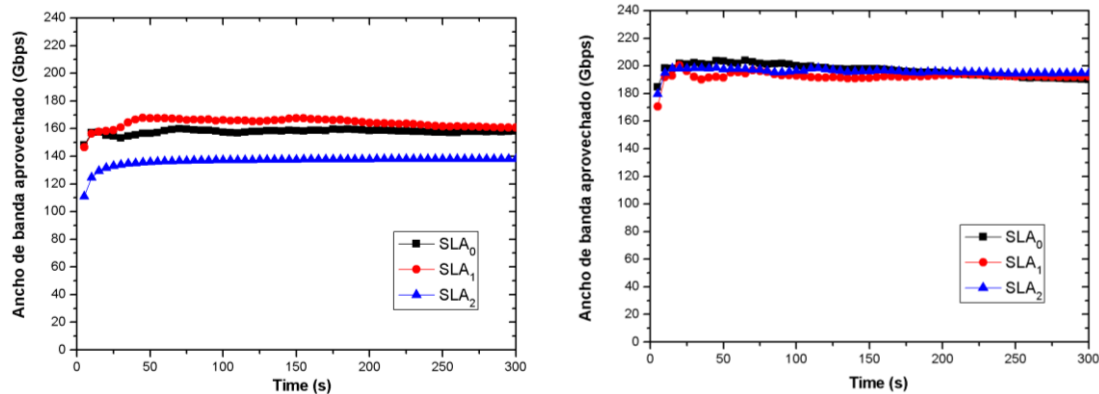


los otros dos, el comportamiento es el mismo y el algoritmo da todo lo que piden las ONUs. Un resumen con estos valores que se obtienen se puede observar en la Tabla 6:

$t_{\text{tuning}}$	$t_{\text{cycle}}$		
	2 ms	6.7 ms	10 ms
0 $\mu\text{s}$	20,5948 Gbps	20,9607 Gbps	21,1315 Gbps
10 $\mu\text{s}$	20,6961 Gbps	21,0175 Gbps	20,8705 Gbps
50 $\mu\text{s}$	20,7388 Gbps	20,8538 Gbps	20,9179 Gbps
300 $\mu\text{s}$	18,7955 Gbps	20,8015 Gbps	20,8436 Gbps

**Tabla 6:** Resultados para cargas bajas cuando se varía el  $t_{\text{cycle}}$  manteniendo constante el  $t_{\text{tuning}}$

Finalmente, se muestra el ancho de banda ofrecido a cada ONU para constatar que lo asigna correctamente en base a las características del escenario de simulación genérico que se describían en la Tabla 2. Se muestra también en la Figura 51 (Imagen de la izquierda) el caso más restrictivo, es decir, para el tiempo de ciclo de 2 ms y con tiempo de *tuning* igual a 300  $\mu\text{s}$ , pero en este caso, como la capacidad de la red es mayor que lo demandado, a las ONUs de los 3 SLAs se les dará casi todo el ancho de banda que piden, cercano a los 200 Mbps (carga de 0.2), aunque inferior en este caso, debido a que afecta este tiempo de *tuning* alto. Por otro lado, para el tiempo de *tuning* de 50  $\mu\text{s}$ , según se observa en la Figura 51 (Imagen de la derecha), se comprueba que sí se les dará casi estos 200 Mbps a las ONUs de los 3 SLAs, ya que este tiempo de *tuning* no afecta como lo hacía el utilizado anteriormente.



**Figura 51:** Evolución del ancho de banda aprovechado por ONU para cargas bajas cuando  $t_{\text{cycle}} = 2 \text{ ms}$  y  $t_{\text{tuning}} = 300 \mu\text{s}$  y  $50 \mu\text{s}$

## 5.5 Conclusiones

En este capítulo se ha descrito la implementación de un algoritmo TWDM en el simulador de redes desarrollado en OMNeT++ del que ya se disponía en el grupo de trabajo, y se han analizado los resultados obtenidos al evaluar el impacto del tiempo de conmutación o *tuning* cuando se consideran diferentes tiempos de ciclo.

Este algoritmo se ejecuta cada vez que llega un mensaje *Report* de una ONU al OLT. De esta manera, no se desperdicia tiempo en esperar la llegada de todos esos mensajes de control en un ciclo, es decir, utiliza una política de *polling*. Una vez llega el mensaje *Report* al OLT, se procede a asignar el ancho de banda y el instante de tiempo de inicio de transmisión para dicha ONU, tal y como se expuso en la sección correspondiente en la primera longitud de onda que quede libre.

Para la ejecución de las simulaciones con este algoritmo, se ha presentado el escenario de simulación considerando varias clases de servicio para cada abonado y realizando las mismas para distintas cargas, comprobando así el impacto del tiempo de *tuning* para cada una de ellas. Los resultados obtenidos en el análisis de estos tiempos de conmutación, muestran que tanto para cargas altas como para cargas bajas, aunque afectando en mayor medida a las cargas altas, según aumenta el tiempo de ciclo, el impacto del tiempo de *tuning* en el aprovechamiento del ancho de banda total disponible se va reduciendo. Por otro lado, a medida que aumenta el tiempo de *tuning*, éste impactará más en el ancho de banda aprovechado cuanto menor sea el tiempo de ciclo. En concreto, para nuestro caso en concreto, el ancho de banda total aprovechado se ve afectado cuando el tiempo de ciclo es el menor de los analizados, el de 2 ms, y cuando el tiempo de *tuning* es el mayor de los estudiados, el de 300  $\mu$ s. Sin embargo, en ninguna otra combinación de los tiempos de ciclo y *tuning* utilizados en el estudio se ve afectado dicho ancho de banda total aprovechado.

# 6

## Conclusiones y Líneas Futuras

### 6.1 Conclusiones

En este Trabajo Fin de Grado (TFG) se ha llevado a cabo un análisis de algunas de las estrategias para la asignación dinámica de ancho de banda en redes PON, partiendo de una red de acceso óptica que había sido previamente desarrollada e implementada por el Grupo de Comunicaciones Ópticas (GCO) en la plataforma de simulación OMNeT++.

Para llevar a cabo esta tarea, en un principio se ha recopilado información sobre el estado del arte en el ámbito de redes de acceso ópticas cuyo despliegue a nivel real se está llevando a cabo de forma masiva en los últimos años. De este modo, se ha procedido a trabajar en el análisis del comportamiento de una red de acceso EPON, como la ya implementada en el simulador de redes ópticas OMNeT++ por otros Trabajos Fin de Grado previos. Así mismo, se ha llevado a cabo un análisis de las prestaciones del uso de la estrategia de redes neuronales en controladores PID para la asignación dinámica del ancho de banda, incluyendo las señales de “*bias*” en las que se basa nuestro estudio. Todo esto fue explicado en el Capítulo 4, considerando diferentes parámetros dentro de la red bajo un patrón de tráfico rafagoso, para ver los resultados que se extraían al usar un método de sintonización automática de PIDs basado en redes neuronales. Esta red neuronal diseñada e implementada lleva a cabo una sintonización automática en tiempo real que hace evolucionar a los parámetros de sintonización gracias a unas funciones de activación y a la evolución de la matriz de pesos de las diferentes capas de la red. En base a esto, el ancho de banda asignado se ajusta constantemente para asegurar los niveles mínimos garantizados de los usuarios de acuerdo a los requisitos de calidad de su perfil. Si se necesita cambiar el nivel garantizado de algunos perfiles por condicionantes de la red, el PID reajusta automáticamente el ancho de banda en tiempo real y cambia los anchos de banda máximos permitidos de acuerdo con el error cometido, por lo que la capacidad de adaptación a los cambios en el patrón de tráfico o en las condiciones de la red es bastante robusta y rápida. En cuanto al impacto que tiene la inclusión de las señales de “*bias*”, indicar que se obtiene una convergencia ligeramente más rápida, hasta que se

alcanzan los valores establecidos para los anchos de banda garantizados, y del mismo modo los parámetros de sintonización tienen unos valores más estables, donde antes se obtenían mayores picos y variaciones. De todo este estudio, se ha concluido, que esta estrategia es una buena opción para la obtención de los parámetros de sintonización del controlador PID a la hora de realizar una asignación dinámica y eficiente de ancho de banda para diferentes perfiles de abonado en redes PON.

Por otro lado, en este Trabajo Fin de Grado (TFG), también se llevó a cabo la implementación del algoritmo híbrido TWDM en base al estándar de la siguiente generación NGPON2, por lo que en primer lugar, hubo que hacer una amplia búsqueda de información sobre el estado del arte de este estándar y sobre los algoritmos TWDM. Todo esto se explicó en el Capítulo 5, donde se vio que al consultar toda esta bibliografía, es una gran tendencia la utilización de receptores y transmisores sintonizables, lo que implica considerar un tiempo de conmutación o tiempo de *tuning* cada vez que uno de ellos pasa de transmitir de una longitud de onda a otra distinta. Una vez desarrollado todo esto en el simulador de redes de acceso del que ya se disponía en OMNeT++, se comprobó su correcto funcionamiento analizando la influencia de estos tiempos de conmutación tanto para el ancho de banda total aprovechado (*throughput*) de la red como en el ancho de banda aprovechado por ONU. Como conclusión, se puede decir que, tanto para las cargas altas como para las bajas, el impacto del tiempo de *tuning* en el aprovechamiento del ancho de banda total disponible es mucho menor a medida que aumenta el tiempo de ciclo. En cuanto a la influencia de variar los tiempos de ciclo máximos, se puede indicar también para cargas altas y bajas que, a medida que el tiempo de *tuning* aumenta, se ve que sólo afecta al ancho total aprovechado cuanto menor es el tiempo de ciclo.

## 6.2 Líneas Futuras

La implementación y validación de nuevas estrategias de asignación dinámica de ancho de banda en el simulador de redes de acceso en la plataforma de simulación OMNeT++, ha propiciado la aparición de nuevas líneas de investigación que se pueden seguir una vez concluido este Trabajo Fin de Grado (TFG).

Por un lado, sería de gran interés que se utilizará la Red Neuronal descrita para controlar la gestión del ancho de banda de manera general, no como sucede ahora, donde esta Red Neuronal está integrada con el controlador PID sintonizarlo de forma adecuada y dinámica.

Por otra parte, se puede continuar trabajando en la implementación de otros algoritmos o esquemas de asignación dinámica de ancho de banda ya existentes, o proponiendo nuevos algoritmos que mejoren las prestaciones de los ya desarrollados aplicados a las nuevas arquitecturas NGPON2. En concreto, en nuestro algoritmo híbrido TWDM implementado, según lo tenemos diseñado, están funcionando todas las longitudes de onda disponibles al mismo tiempo, permitiendo la conmutación en cada ciclo al trabajar

con un transmisor y un receptor sintonizables. En este sentido, resultaría de gran interés reducir el número de conmutaciones que se producen para que sea un algoritmo más eficiente.

# 7

## Bibliografía

- [1] A. Banerjee, G. Kramer, Y. Ye, S. Dixit y B. Mukherjee, “Advances in Passive Optical Networks (PONs)”, en *Emerging optical network technologies: Architectures, Protocols and Performance*, K. M. Sivalingnan y S. Subramaniam, Eds. Spring Street, Nueva York: Springer 2005.
- [2] <http://newslog.itu.int/archives/1212>.
- [3] A. Vargas and OpenSim Ltd., “Omnet++ User Manual”. Disponible *on-line* en: <http://www.omnetpp.org/doc/omnetpp/manual/usman.html>
- [4] A. Vargas and OpenSim Ltd., “Omnet++ User Guide”. Disponible *on-line* en: <http://www.omnetpp.org/doc/omnetpp/UserGuide.pdf>
- [5] B. Lung, “PON architecture Future proofs FTTH,” *Lightwave*, vol. 16(10), pp. 104-107, Nov. 1999.
- [6] M. Pesavento y A. Kesley, “PONs for the Broadband Local Loop,” *Lightwave*, vol. 16(10), pp. 68-74, Nov. 1999.
- [7] IEEE 802.3 (2006) Call For Interest: 10 Gbps PHY for EPON. Disponible en: <http://www.ieee802.org/3/cfi>.
- [8] Página Web Oficial de OMNeT++: <http://www.omnetpp.org/>
- [9] Generador de tráfico *Self-Similar* desarrollado por Kramer. Disponible en: <http://www.csif.cs.ucdavis.edu/~kramer/research.html>.
- [10] Versión 2 del Generador de tráfico *Self-Similar* desarrollado por Kramer. Disponible en: [http://www.csif.cs.ucdavis.edu/~kramer/code/trf\\_gen2.html](http://www.csif.cs.ucdavis.edu/~kramer/code/trf_gen2.html).

- [11] P. Isasi, I. M. Galván, *Redes de Neuronas Artificiales: Un enfoque práctico*. Ed. Pearson Educación, 2004.
- [12] E.N. Sánchez, A.Y. Alanis, *Redes Neuronales: Conceptos fundamentales y aplicaciones a control automático*. Ed. Pearson Educación, 2006.
- [13] M.A. Keshari and A. Mehetab, Study of the design and tuning methods of pid controller based on fuzzy logic and genetic algorithm, Department of Electronics and Communication Engineering National Institute of Technology, Thesis, Bachelor in Electronics, 2012.
- [14] B. Yang, J. Chang, H. Su, J. Peng, S. Zhang, S. Guo, L. Zhang, C. Srinivasakannan, Z. Liu, Z. Li and Z. Cao Z, “Self-Adaptive PID Controller Integrated with RBFNN Identification Applied to Microwave Drying Process,” *Journal of Convergence Information Technology*, vol. 8, pp. 779-783, January 2013.
- [15] S. Shoujun and L. Weiguo, *Application of Improved PID Controller in Motor Drive System, PID Control, Implementation and Tuning*, InTech, 2011, Available from: <http://www.intechopen.com/books/pid-control-implementation-and-tuning/application-of-improved-pid-controller-in-motor-drive-system>.
- [16] K. Vikas, G. Prerna and A.P. Mittal, “ANN based self-tuned PID like adaptive controller design for high performance PMSM position control,” *Expert Systems with Applications*, vol. 41, pp. 7995-8002, December 2014.
- [17] J. Peng and R. Dubay, “Identification and adaptive neural network control of a DC motor system with dead-zone characteristics,” *ISA Transactions*, vol. 50, pp. 588-598, July 2011.
- [18] K.J. Aström, T. Hägglund, *PID Controllers: Theory, Design and Tuning*, Ed. Research Triangle Park, NC, Instrument Soc. Amer., 2nd edition, 1995.
- [19] K.J. Aström, T. Hägglund, *Advanced PID control*. Ed. Research Triangle Park, NC: ISA-The Instrumentation, Systems, and Automation Society, 2006.
- [20] R. Isermann, *Digital Control Systems Vol I: Fundamentals, Deterministic Control*. Ed. Springer-Verlag, 2nd edition, 1989.
- [21] G. Kramer, B. Mukherjee and G. Pesavento, “Interleaved Polling with Adaptive Cycle Time (IPACT): A Dynamic Bandwidth Distribution Scheme in an Optical Access Network,” *Photonic Network Communications*, vol. 4, pp. 89-107, January 2002.

- [22] G. Lin, G. Liu, "Tuning PID Controller Using Adaptive Genetic Algorithms," en 5th International Conference on Computer Science & Education, Hefei (China), Agosto 2010.
- [23] L. Fan, E.M. Joo, "Design for Auto-tuning PID Controller Based on Genetic Algorithms," en 4th IEEE Conference on Industrial Electronics and Applications, (ICIEA 2009), Xi'an (China), Mayo 2009.
- [24] P.J. Rensburg, I.S. Shaw, J.D. Wyk, "Adaptive PID-control using a Genetic Algorithm," en 2nd International Conference on Knowledge-Based Intelligent Electronic System, Adelaide (Australia), Abril 1998.
- [25] J.C. Shen, "New tuning method for PID controller," ISA Transactions, vol. 41, no. 4, pp. 473-484, Octubre 2002.
- [26] P. Wang, D.P. Kwok, "Optimal design of PID process controllers based on Genetic Algorithms," Control Engineering Practice, vol. 2, no. 4, pp. 641-648, Agosto 1994.
- [27] C. Li, J. Lian, "The Application of Immune Genetic Algorithm in PID Parameter Optimization for Level Control System," en IEEE International Conference on Automation and Logistics, Jinan (China), Agosto 2007.
- [28] C.K. Chen, H.H. Kuo, J.J. Yan, T.L. Liao, "GA-based PID active queue management control design for a class of TCP communication networks," Expert Systems with Applications, vol. 36, no. 2 parte 1, pp. 1903-1913, Marzo 2009.
- [29] Jan Jantzen and Carl Jakobsen, "Turning PID Controller Tuning Into a Simple Consideration of Settling Time".
- [30] Sofiane Khadraoui, Micky Rakotondrabe, and Philippe Lutz, "PID-Structured Controller Design for Interval Systems: Application to Piezoelectric Microactuators".
- [31] E. Poutin A. Pomerleau, "PID tuning for integrating and unstable processes".
- [32] Biao Yang, Jun Chang, Hezhou Su, Jinhui Peng, Shimin Zhang, Shenghui Guo, Libo Zhang, C. Srinivasakannan, Zilian Liu, Zhimin Li, Zhanyuan Cao, "Self-Adaptive PID Controller Integrated with RBFNN Identification Applied to Microwave Drying Process".
- [33] Ali Zribi, Mohamed Chtourou, Mohamed Djemel, "A New PID Neural Network Controller Design for Nonlinear Processes".



- [34] Fernando Izaurieta y Carlos Saavedra, “Redes Neuronales Artificiales”.
- [35] Rodrigo Salas, “Redes Neuronales Artificiales”.
- [36] H. Xu, J. Lai, Z. Yu and J. Liu, “Based on Neural Network PID Controller Design and Simulation,” *In Proceedings of the 2012 2nd International Conference on Computer and Information Application (ICCIA 2012)*, Taiyuan, China, pp. 844-866.
- [37] L. Luoren and L. Jinling, “Research of PID Control Algorithm Based on Neural Network,” *In Proceedings of Energy Procedia*, 2011, pp. 6988-6993.
- [38] W. Lu, “The PID Controller Based on the Artificial Neural Network and the Differential Evolution Algorithm,” *Journal of Computers*, vol. 7, pp. 2368-2375, October 2012.
- [39] ITU-T Recomendación G.989, “40-Gigabit-capable passive optical networks (NG-PON2): Definitions, abbreviations and acronyms” Telecommunication Standardization Sector of ITU, 2015.
- [40] ITU-T Recomendación G.989.2, “40-Gigabit-capable passive optical networks 2 (NG-PON2): Physical media dependent (PMD) layer specification” Telecommunication Standardization Sector of ITU, 2014.
- [41] Yuanqiu Luo, Xiaoping Zhou, Frank Effenberger, Xuejin Yan, Guikai Peng, Yinbo Qian, and Yiran Ma, “Time- and Wavelength-Division Multiplexed Passive Optical Network (TWDM-PON) for Next-Generation PON Stage 2 (NG-PON2)”.
- [42] Ruizhi Tang, Xin Chen, Bangjiang Lin, Paikun Zhu, Juhao Li, Zhangyuan Chen, Yongqi He, “Wavelength and Time Slot Assignment Scheme for TWDM-PONs Based on Inter-ONU Priority Rating”.
- [43] Man Soo Han, “Dynamic Wavelength and Bandwidth Allocation with QoS for TWDM PON”.
- [44] Man Soo Han, “Energy Efficient Dynamic Wavelength and Bandwidth Allocation for TWDM PON”.
- [45] Dianbo Zhao, Yuanqiu Luo, Jianhe Gao, and Huafeng Lin, “Multiple Wave length Load Balancing in TWDM-PON”.

- [46] Zhengxuan Li, Lilin Yi, Meihua Bi, Jun Li, Hao He, Xuelin Yang and Weisheng Hu, “Experimental Demonstration of a Symmetric 40-Gb/s TWDM-PON”.
- [47] Wolfgang Pöhlmann, Bernhard Deppisch, Thomas Pfeiffer, Carlo Ferrari, Mark Earnshaw, Alex Duque, Robert L. Farah, Joe Galaro, James Kotch, Man Fai Lau, Doutje T. van Veen, and Peter Vetter, “Low Cost TWDM by Wavelength-Set Division Multiplexing”.
- [48] Yong Guo, Songlin Zhu, Guohua Kuang, Yongjia Yin, Dezhi Zhang, and Xin Liu, “Demonstration of a Symmetric 40 Gbit/s TWDM-PON Over 40 km Passive Reach Using 10 G Burst-Mode DML and EDC for Upstream Transmission”.
- [49] Hang Zhao, Sheng Hu, Jialin Zhao, Yao Zhu, Yonglin Yu, Liam P. Barry, “Chirp-Compensated DBR Lasers for TWDM-PON Applications”.
- [50] ITU-T Recomendación G.989.1, “40-Gigabit-capable passive optical networks (NG-PON2): General requirements” Telecommunication Standardization Sector of ITU, 2013.
- [51] ITU-T Recomendación G.989.3, “40-Gigabit-capable passive optical networks (NG-PON2): Transmission convergence layer specification” Telecommunication Standardization Sector of ITU, 2015.
- [52] A. Buttaboni, M. D. Andrade, and M. Tornatore, “A Multi-Threaded Dynamic Bandwidth and Wavelength Allocation Scheme With Void Filling for Long Reach WDM/TDM PONs,” *J. Lightw. Technol.*, vol. 31, no. 8, pp. 1149-1157, April 2013.
- [53] X. Hu, X. Chen, Z. Zhang, and J. Bei, “Dynamic Wavelength and Bandwidth Allocation in Flexible TWDM Optical Access Network,” *IEEE Commun. Lett.*, vol. 18, no. 12, pp. 2113-2116, Dec. 2014.
- [54] H. Yang, W. Sun, J. Li, and W. Hu, “Energy Efficient TWDM Multi-PON System With Wavelength Relocation,” *J. Opt. Commun. Netw.*, vol. 6, no. 6, pp. 571-577, June. 2014.
- [55] R. Wang, H. H. Lee, S. S. Lee, and B. Mukherjee, “Energy Saving via Dynamic Wavelength Sharing in TWDM-PON,” *IEEE J. Sel. Areas Commun.*, vol. 32, no. 8, pp. 1566-1574, Aug. 2014.

- [56] H. Wang, S. Su, G. Rentao and J. Yuefeng “A minimum wavelength tuning scheme for dynamic wavelength assignment in TWDM-PON,” 14th International Conference on Optical Communications and Networks (ICO CN) 2015.
- [57] H. Wang, Y. Liang, G. Rentao, J. Yuefeng, M. Yiran, Z. Chengliang and X. Wang, “LP-DWBA: a DWBA algorithm based on linear prediction in TWDM-PON”, 14th International Conference on Optical Communications and Networks (ICO CN), 2015.
- [58] M.P.I. Dias, E. Wong, P.V Dung and L. Valcarenghi, “Offline energy-efficient dynamic wavelength and bandwidth allocation algorithm for TWDM-PONs, IEEE International Conference on Communications (ICC), 2015, June 2015.
- [59] M. Pubuduni Imali Dias, Dung Pham Van, Luca Valcarenghi, and Elaine Wongl, “Energy-efficient TWDM-PON with VCSEL ONUs”
- [60] L. Valcarenghi, K. Kondepu, P. Castoldi, “Reducing Delay Penalties in Energy-Efficient TWDM PON through Reconfiguration Threshold Adaptation”
- [61] Ruizhi Tang, Xin Chen, Bangjiang Lin, Paikun Zhu, Juhao Li, Zhangyuan Chen, Yongqi He, “Wavelength and Time Slot Assignment Scheme for TWDM-PONs Based on Inter-ONU Priority Rating”
- [62] Anna Buttaboni, Marilet De Andrade, and Massimo Tornatore, "Dynamic Bandwidth and Wavelength Allocation with Coexistence of Transmission Technologies in TWDM PONs"
- [63] Hark Yoo, Dongsoo Lee, Man Soo Han, "Optimal Wavelength Provisioning with Fuzzy Logic Control for Power Saving in TWDM-PONs"
- [64] Abhishek Dixit, Bart Lannoo, Didier Colle, Mario Pickavet, and Piet Demeester, "Dynamic Bandwidth Allocation with Optimal Wavelength Switching in TWDM-PONs"
- [65] Hongxiang Wang, Suyan Su, Rentao Gu y Yuefeng Ji, “A minimum wavelength tuning scheme for dynamic wavelength assignment in TWDM-PON”.