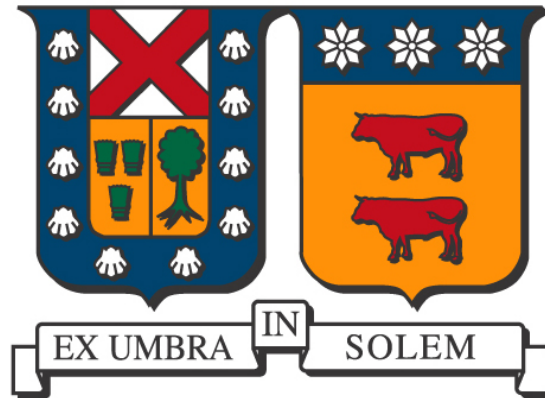


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
SANTIAGO - CHILE



“Diseño e implementación de un simulador basado en SDN y aprendizaje por refuerzo para la gestión eficiente de recursos en Redes Ópticas Pasivas (PON)”

Alex Aravena Tapia

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL
TELEMÁTICO**

PROFESOR GUÍA:

Patricia Franco Troya

PROFESOR CORREFERENTE:

Patricia Morales Calvo

AGOSTO - 2026



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: Diseño e implementación de un simulador basado en SDN y aprendizaje por refuerzo para la gestión eficiente de recursos en Redes Ópticas Pasivas (PON)

Nombre del candidato(a): Alex Aravena Tapia

Carrera / Grado: Ingeniería Civil Telemática

Campus: San Joaquín

Departamento: Ingeniería Electrónica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Patricia Franco Troya, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (marcar una opción):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 04/08/2026

Firma: _____

Estudiante o Candidato(a):

Fecha: 04/08/2026

Firma: _____

PATRICIA FRANCO
TROYA

Firmado digitalmente por
PATRICIA FRANCO TROYA
Fecha: 2026.08.04 15:06:02 -04'00'

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Agradecimientos

A cada uno de los que llevo en la sangre, mi familia, que supo darme luz cuando el camino se nublaba, y que soportó mis silencios y mis tormentas sin dejar de estar.

A Karina y Paulo, que vivían pendientes de que llegara bien después de las jornadas eternas en la universidad, y que siempre estuvieron con la mano tendida para lo que se ofreciera.

A mi pareja, que apareció en la cuesta más empinada y se quedó a mi lado. Fuiste mi cable a tierra cuando ya no daba más. Gracias por cada escapada para despejar la cabeza, por el amor que hiciste florecer en mí y por todos los ánimos que me diste en este camino.

A mis amigos, que son la prueba viva de que sin ellos esto no habría sido posible. Las jornadas de estudio, los trabajos en grupo, las juntas recreativas y todos esos momentos que se quedan para siempre en la memoria.

Y al final, van ustedes dos. Mamá, papá... gracias por darme todas las herramientas para crecer, por empujarme con sus expectativas y por creer en mí más que yo mismo. Solo me pidieron que estudiara; todo lo demás, ustedes lo hicieron. No se puede pedir más cuando ya se tiene lo mejor.

Papá, esta memoria va dedicada a ti. Me enseñaste que la felicidad es lo más importante y que hay que hacer el bien siempre, el bien a todos, el mal a nadie. Siempre recordaré tu sonrisa. Gracias.

Rubén Aravena Uribe

Resumen

La creciente demanda de servicios digitales sensibles al retardo exige mecanismos flexibles de asignación de recursos en redes ópticas pasivas (PON), especialmente en el enlace ascendente, donde múltiples unidades ópticas de red comparten el mismo medio. En este contexto, esta memoria desarrolla y valida *PonLab Simulator*, un *framework* modular para construir topologías PON, configurar escenarios de simulación, integrar algoritmos de asignación dinámica de ancho de banda (DBA) y visualizar métricas de desempeño. La herramienta incorpora una interfaz gráfica, un motor basado en ciclos de sondeo de $125\ \mu\text{s}$, soporte para control centralizado mediante redes definidas por *software* (SDN) y una integración preliminar de agentes de Aprendizaje por Refuerzo Profundo (DRL).

La validación experimental consideró una topología residencial G-PON con una OLT y ocho ONUs, operando a 2.5 Gbps en bajada y 1.25 Gbps en subida. Se compararon tres paradigmas de control: el algoritmo heurístico GIANT, un controlador SDN y el agente inteligente Smart-RL. Los resultados mostraron que SDN alcanzó el mayor *throughput* promedio, con 45.607 MB/s, frente a 42.900 MB/s de GIANT y 20.061 MB/s de DRL. En cambio, Smart-RL obtuvo el mejor desempeño temporal, con un retardo medio de 0.593 ms y un jitter IPDV de 0.069 ms.

Estos resultados evidencian que la optimización de redes PON no debe evaluarse solo por el caudal, sino también por latencia y variabilidad temporal. Así, *PonLab Simulator* constituye una plataforma funcional y extensible para estudiar estrategias DBA heurísticas, centralizadas e inteligentes.

Palabras clave: aprendizaje por refuerzo profundo, asignación dinámica de ancho de banda, redes definidas por *software*, redes ópticas pasivas, simulación de redes.

Abstract

The growing demand for delay-sensitive digital services requires flexible resource allocation mechanisms in Passive Optical Networks (PONs), especially in the upstream channel, where multiple Optical Network Units share the same transmission medium. In this context, this thesis develops and validates *PonLab Simulator*, a modular framework for building PON topologies, configuring simulation scenarios, integrating Dynamic Bandwidth Allocation (DBA) algorithms, and visualizing performance metrics. The tool includes a graphical interface, a simulation engine based on 125 μ s polling cycles, support for centralized control through Software-Defined Networking (SDN), and a preliminary integration of Deep Reinforcement Learning (DRL) agents.

The experimental validation considered a residential G-PON topology with one OLT and eight ONUs, operating at 2.5 Gbps downstream and 1.25 Gbps upstream. Three control paradigms were compared: the GIANT heuristic algorithm, an SDN controller, and the Smart-RL intelligent agent. The results showed that SDN achieved the highest average throughput, reaching 45.607 MB/s, compared with 42.900 MB/s for GIANT and 20.061 MB/s for DRL. Conversely, Smart-RL obtained the best temporal performance, with an average mean delay of 0.593 ms and an average IPDV jitter of 0.069 ms.

These results show that PON optimization should not be assessed only through throughput, but also through latency and temporal variability. Therefore, *PonLab Simulator* provides a functional and extensible platform for studying heuristic, centralized, and intelligent DBA strategies.

Keywords: deep reinforcement learning, dynamic bandwidth allocation, network simulation, passive optical networks, software-defined networking.

Glosario

Aprendizaje por refuerzo Técnica de inteligencia artificial en la cual un agente aprende a tomar decisiones óptimas interactuando con un entorno simulado, basando su ajuste de políticas en la obtención de recompensas o penalizaciones.

Ciclo de sondeo Intervalo periódico de tiempo (típicamente 125 μ s) en el que la OLT consulta el estado de las colas de las ONUs para calcular y asignar concesiones de transmisión en el canal ascendente.

Contenedor de tráfico (T-CONT) Entidad lógica dentro de una ONU que agrupa flujos de tráfico con requisitos de calidad de servicio similares. Cada T-CONT tiene una prioridad y un perfil de servicio (ej. best effort, tiempo real) y es gestionado individualmente por el algoritmo DBA.

DBA Sigla para asignación dinámica de ancho de banda. Refiere al mecanismo utilizado para distribuir los recursos de transmisión de forma eficiente, adaptándose en tiempo real a las variaciones de demanda en la red.

Divisor óptico (splitter) Dispositivo pasivo de la red de distribución óptica que divide la señal de fibra óptica proveniente de la OLT hacia múltiples ONUs en el sentido descendente, y combina las señales ascendentes de las ONUs hacia la OLT.

Equidad (Fairness) Medida de la uniformidad en la distribución de recursos entre los usuarios de la red. Se cuantifica comúnmente mediante el Índice de Equidad de Jain, que toma valores entre 0 y 1, donde 1 indica asignación perfectamente equitativa.

Framework Corresponde a un entorno de trabajo estructurado que integra diversas herramientas, librerías y componentes, utilizado para facilitar el desarrollo, modelado y ejecución de simulaciones de manera modular.

Jitter (IPDV) Variación estadística del retardo de extremo a extremo entre paquetes consecutivos de un mismo flujo. En redes PON se distingue entre jitter intra-ventana (dentro de una misma concesión) e inter-ventana (entre concesiones consecutivas), siendo este último crítico para servicios sensibles como voz y video.

Latencia (Delay) Tiempo transcurrido desde que un paquete ingresa al buffer de la ONU hasta que es recibido exitosamente en la OLT. Es una métrica clave de calidad de servicio, y en el contexto de esta memoria se mide en milisegundos.

Mapa de ancho de banda (BWmap) Estructura de datos generada por la OLT que especifica, para cada ONU y cada T-CONT, el instante de inicio y la duración de la ventana de transmisión en el canal ascendente.

OLT Sigla para "Optical Line Terminal" o terminal de línea óptica. Constituye el equipo de cabecera en una red óptica pasiva, encargado de gestionar y proveer los servicios hacia las unidades de red óptica en las instalaciones de los clientes.

ONU Sigla para "Optical Network Unit" o unidad de red óptica. Representa el equipo terminal situado en el lado del usuario final, el cual recibe la señal proveniente de la cabecera y la acondiciona para su utilización.

OpenFlow Protocolo de comunicaciones empleado en redes definidas por "software" que hace posible la interacción directa entre un controlador central y el plano de reenvío de los dispositivos físicos de la red.

QoS Sigla para calidad de servicio. Involucra los mecanismos e indicadores orientados a garantizar un nivel de rendimiento adecuado en la transmisión de datos, priorizando flujos críticos para reducir la latencia o pérdida de paquetes.

Red de Distribución Óptica (ODN) Conjunto de componentes pasivos (fibra óptica, divisores, conectores y empalmes) que proporcionan el enlace físico entre la OLT y las ONUs en una red PON.

Redes ópticas pasivas Se define como una red en la que la fibra óptica transporta las señales durante todo o gran parte del trayecto hasta el usuario final. Se conoce también como la “última milla” entre el proveedor de servicios de Internet y el cliente.

SDN Sigla para redes definidas por "software". Corresponde a un enfoque de arquitectura que desacopla el plano de control del plano de datos, permitiendo una gestión centralizada, dinámica y programable de los recursos.

Sondeo sin eventos (event-less polling) Técnica de simulación que optimiza el rendimiento computacional evitando la generación de eventos explícitos para cada ciclo de sondeo; en su lugar, el motor avanza el reloj verificando automáticamente los múltiplos de 125 μ s, lo que reduce drásticamente la sobrecarga de la cola de eventos.

Throughput Término que denota la tasa de transferencia efectiva o el volumen de datos que logra transmitirse de forma exitosa a través de un canal de comunicación en un periodo de tiempo determinado.

Throughput (Tasa de transferencia) Cantidad de datos útiles (payload) transmitidos con éxito por unidad de tiempo, medida en MB/s o Gbps. Se calcula como la suma de los tamaños de los paquetes recibidos dividida por el tiempo total de observación.

Siglas

ANN Red Neuronal Artificial (Artificial Neural Network).

CIR Tasa de Información Comprometida (Committed Information Rate).

DBA Asignación Dinámica de Ancho de Banda.

DBRu Reporte de Ancho de Banda Dinámico ascendente (Dynamic Bandwidth Report upstream).

DQN Red Q Profunda (Deep Q-Network).

DRL Aprendizaje por Refuerzo Profundo (Deep Reinforcement Learning).

EqD Retardo de Ecualización (Equalization Delay).

GEM Método de Encapsulamiento GPON (GPON Encapsulation Method).

GPON Red Óptica Pasiva con capacidad Gigabit (Gigabit-capable Passive Optical Network).

GUI Interfaz Gráfica de Usuario (Graphical User Interface).

LSTM Memoria a Corto y Largo Plazo (Long Short-Term Memory).

NETCONF Protocolo de Configuración de Red (Network Configuration Protocol).

NG-PON Red Óptica Pasiva de próxima generación (Next-Generation PON).

ODN Red de Distribución Óptica (Optical Distribution Network).

OLT Optical Line Terminal.

OMCI Interfaz de Gestión y Control de la ONU (ONU Management and Control Interface).

ONOS Sistema Operativo de Red Abierto (Open Network Operating System).

ONU Optical Network Unit.

PIR Tasa de Información Pico (Peak Information Rate).

PON Red Óptica Pasiva.

PPO Optimización de Políticas Proximales (Proximal Policy Optimization).

QoS Calidad de Servicio.

SDN Redes Definidas por “Software”.

VLAN Red de Área Local Virtual (Virtual Local Area Network).

XGS-PON Red Óptica Pasiva de 10 Gbps simétrica (10-Gigabit Symmetric Passive Optical Network).

YANG Lenguaje de Modelado de Datos para NETCONF/RESTCONF.

Tabla de Contenido

1	Introducción	1
1.1	Definición del problema	1
1.2	Acercamiento a la solución	2
1.3	Objetivos	2
1.3.1	Objetivo Principal	2
1.3.2	Objetivos Específicos	3
1.4	Estructura	3
2	Estado del Arte y Marco Teórico	5
2.1	Antecedentes	5
2.2	Infraestructura Física y Gestión de Recursos (PON y DBA)	5
2.2.1	Arquitectura de Redes Ópticas Pasivas (PON)	5
2.2.2	Asignación Dinámica de Ancho de Banda (DBA)	6
2.3	Paradigma de Control Abierto y Programable (SDN)	8
2.4	Motor de Optimización Inteligente	10
2.5	Métricas de Evaluación y Calidad de Servicio (QoS)	11
2.6	Estado del Arte	13
2.6.1	Evolución de las estrategias heurísticas para DBA	13
2.6.2	Control programable mediante SDN para redes PON	17
2.6.3	Aplicación de inteligencia artificial y aprendizaje por refuerzo en redes . .	20
2.6.4	Simuladores y plataformas experimentales para PON	23
2.6.5	Brecha Identificada y Oportunidad de Investigación	28
3	Diseño, arquitectura e implementación del simulador PonLab	31
3.1	Requisitos del Sistema	31
3.1.1	Requisitos funcionales	31
3.1.2	Requisitos no funcionales	32
3.1.3	Requisitos de interfaces	32

3.1.4	Entorno de desarrollo y ejecución	32
3.2	Perfiles de usuario	34
3.3	Diseño de Arquitectura del Sistema	36
3.3.1	Flujo general de funcionamiento del sistema	36
3.3.2	Diagrama de Arquitectura	37
3.3.3	Variables, Parámetros y Salidas	39
3.4	Módulos	43
3.4.1	Usuario	43
3.4.2	Topologías	44
3.4.3	Interfaz	45
3.4.4	Simulación	46
3.4.5	Algoritmos y SDN	49
3.4.6	Salida Gráfica/Textual	50
3.4.7	Módulo de aprendizaje por refuerzo	51
3.5	Diseño de Interfaces	54
3.5.1	Modelo de Navegación	54
3.5.2	Vistas aplicación	55
3.6	Metodología Experimental	61
3.6.1	Escenario de simulación y parámetros	62
3.6.2	Modelos de Comparación	64
3.6.3	Métricas de Evaluación	64
4	Validación funcional y resultados experimentales	65
4.1	Validación Funcional del Framework PonLab	65
4.2	Monitoreo Operacional del Esquema Heurístico Tradicional	68
4.3	Evaluación de la Arquitectura de Control Centralizado SDN	72
4.4	Resultados Preliminares del Módulo de Aprendizaje por Refuerzo (DRL)	76
4.5	Discusión de Resultados	79
4.6	Análisis comparativo de rendimiento en el enlace ascendente	80
4.6.1	Análisis de Throughput	81

4.6.2	Análisis de Latencia (Mean Delay)	82
4.6.3	Análisis de Jitter (IPDV)	84
4.7	Interpretación global del compromiso entre caudal y QoS	85
4.8	Alcances y limitaciones de la evaluación	86
5	Conclusiones	87

Referencias

Anexos

Anexo A Validación del Cumplimiento de Objetivos Específicos

A.1	Validación del OE1: Diseño de la arquitectura para asignación de recursos
A.2	Validación del OE2: Integración de técnicas heurísticas, SDN e Inteligencia Artificial
A.3	Validación del OE3: Integración de herramientas gráficas y textuales
A.4	Validación del OE4: Rendimiento del simulador en escenarios realistas y contraste con la literatura

Índice de figuras

2.1	Diagrama de una Red Óptica Pasiva.	6
2.2	Diagrama de funcionamiento del DBA.	8
2.3	Arquitectura de Redes Definidas por Software [5].	9
2.4	Diagrama interacción agente-entorno en el proceso de decisión de Markov [16] . .	11
2.5	Interfaz de simulador NS-3 (imagen referencial)	24
2.6	Interfaz de simulador OMNet++ (imagen referencial)	25
2.7	Interfaz de simulador OPNET (imagen referencial)	26
2.8	Interfaz de simulador MiniNet (imagen referencial)	27
3.1	Diagrama de flujo general.	36
3.2	Diagrama de arquitectura.	38
3.3	Diagrama de navegación.	54
3.4	Pantalla de PonLab Simulator	56
3.5	Interfaz Principal Ponlab simulador	56
3.6	Panel de Dispositivos	58
3.7	Generación de topología con 1 OLT conectada a 2 ONU's	59
3.8	Panel de Propiedades de un terminal de linea óptico	59
3.9	Paneles de apartado de simulación.	60
3.10	Panel de Aprendizaje Reforzado	61
3.11	PonLab Simulator modo oscuro	62
3.12	Esquema de la topología y escenario de simulación en PonLab.	63
4.1	Panel de creación y configuración de parámetros para el dispositivo OLT_base. . .	66
4.2	Interfaz gráfica de configuración de parámetros para una ONU línea base.	66
4.3	Topología residencial completa de 1 OLT y 8 ONUs construida en el entorno gráfico.	67
4.4	Topología de red con integración automática del paradigma de control centralizado SDN.	68
4.5	Interfaz de simulación de PonLab: Selección del algoritmo base GIANT y ajuste del tiempo de ejecución.	70

4.6	Evolución temporal del Throughput en el canal upstream usando el módulo DBA Heurístico.	70
4.7	Retardo promedio registrado para las 8 ONUs bajo el control tradicional basado en GIANT.	71
4.8	Comportamiento del Jitter (IPDV) upstream para el escenario base heurístico. . . .	71
4.9	Evolución temporal del Throughput en el canal upstream bajo la orquestación del Controlador SDN.	72
4.10	Retardo promedio registrado para las 8 ONUs bajo la gestión del plano de control SDN.	73
4.11	Comportamiento del Jitter (IPDV) upstream bajo el esquema de control centralizado.	74
4.12	Métricas consolidadas en el panel de resumen global de la arquitectura SDN.	74
4.13	Interfaz de monitoreo y estado operacional del componente controlador SDN. . . .	75
4.14	Distribución de ancho de banda y tráfico por clase de servicio en el entorno SDN. .	76
4.15	Entorno gráfico de PonLab para el plano de control inteligente: (a) interfaz de ajuste de hiperparámetros y entrenamiento con PPO, y (b) configuración de la prueba utilizando el modelo RL generado.	77
4.16	Evolución temporal del Throughput en el canal upstream bajo la gestión del agente DRL.	78
4.17	Retardo promedio registrado para las 8 ONUs bajo el plano de control inteligente. .	78
4.18	Comportamiento del Jitter IPDV Mean upstream operando con el algoritmo Smart-RL.	79
4.19	Comparación temporal del <i>throughput</i> en el enlace ascendente para los tres paradigmas evaluados. Las líneas segmentadas representan el valor promedio de cada enfoque durante la simulación de 10 segundos.	81
4.20	Comparación temporal del <i>Mean Delay</i> en el enlace ascendente para los tres paradigmas evaluados. Las líneas segmentadas representan el valor promedio de cada enfoque durante la simulación de 10 segundos.	83
4.21	Comparación temporal del <i>Jitter IPDV Mean</i> en el enlace ascendente para los tres paradigmas evaluados. Las líneas segmentadas representan el valor promedio de cada enfoque durante la simulación de 10 segundos.	84

Índice de tablas

1	Resumen de características de algoritmos DBA.	17
2	Tabla resumen de enfoques de aprendizaje por maquina en redes	23
3	Características de entornos de simulación para redes.	27
4	Eventos y respuestas del sistema	33
5	Respuestas del sistema y sus parámetros	33
6	Perfil de usuario: OLT (Proveedor de Servicio)	35
7	Perfil de usuario: ONU (Usuario Final)	35
8	Enumeración de módulos internos del sistema.	38
9	Matriz de trazabilidad entre requisitos funcionales y módulos del sistema.	39
10	Salidas del Simulador PonLab - Topología	41
11	Salidas del Simulador PonLab - Gráficos de Análisis	41
12	Salidas del Simulador PonLab - Datos de Simulación	42
13	Salidas del Simulador PonLab - Reportes	42
14	Parámetros de simulación	63
15	Matriz de validación funcional de requisitos del sistema y trazabilidad de objetivos.	69
16	Resumen comparativo de métricas de rendimiento por paradigma de control.	81
17	Matriz de trazabilidad y cumplimiento de Objetivos Específicos.	

1. Introducción

El crecimiento exponencial del tráfico en redes de telecomunicaciones, impulsado por servicios como video bajo demanda, aplicaciones en la nube y conectividad 5G, ha planteado nuevos desafíos en la planificación y gestión eficiente de redes de acceso óptico. Las Redes ópticas pasivas (PON) representan una tecnología clave para satisfacer la demanda de mayor ancho de banda con una infraestructura escalable y de bajo costo. Sin embargo, los métodos tradicionales de asignación de recursos presentan limitaciones al enfrentar condiciones dinámicas de tráfico, calidad de servicio (QoS) diferenciada y heterogeneidad en los dispositivos [1] [2] [3].

1.1 Definición del problema

Actualmente, la gestión de recursos en redes ópticas pasivas enfrenta múltiples desafíos técnicos. Entre ellos se destaca la rigidez de las plataformas propietarias utilizadas por muchos fabricantes [4], la falta de mecanismos estandarizados para el control programable de la red, y la dificultad de incorporar nuevas políticas de asignación de recursos sin re diseñar completamente el sistema [5]. Del mismo modo, el incremento sostenido del tráfico y la necesidad de garantizar QoS impone una presión adicional sobre los algoritmos tradicionales, que no responden de forma eficiente a variaciones en tiempo real ni a escenarios con múltiples tipos de usuarios y servicios [6] [7].

En este contexto, el sistema propuesto busca ofrecer una solución flexible, programable y extensible, mediante el uso de controladores SDN que permiten desacoplar el plano de control del plano de datos. Esto habilita la implementación de nuevas políticas de gestión de ancho de banda y facilita la interoperabilidad entre equipos heterogéneos, al mismo tiempo que se mantiene la compatibilidad con arquitecturas existentes de tipo PON [8].

Por consiguiente, el problema central de la presente memoria radica en la necesidad de superar la rigidez y las limitaciones de adaptabilidad de las arquitecturas tradicionales de gestión en redes ópticas pasivas. Para dar respuesta a este desafío, se hace necesario evaluar de qué manera el diseño de un entorno integral de simulación (que incorpore control programable mediante redes definidas por 'software' y políticas avanzadas de asignación) permite optimizar la distribución dinámica de recursos y garantizar la calidad de servicio ante escenarios de alta variabilidad de tráfico.

1.2 Acercamiento a la solución

Para abordar este problema, se investigaron propuestas recientes que integran controladores SDN con arquitecturas PON, destacando soluciones como la implementación de agentes Open-Flow capaces de comunicarse directamente con “chipsets” de OLT, evitando capas intermedias de virtualización [8]. Estas propuestas han demostrado ser efectivas para la configuración de servicios de red en tiempo real, con bajos requisitos de complejidad y alta flexibilidad.

Además, se han identificado enfoques que incorporan el uso de modelos jerárquicos y herramientas de programación orientadas a modelos (como YANG/NETCONF), los cuales permiten representar con mayor precisión los recursos y estados de la red, así como habilitar su visualización y monitoreo continuo mediante telemetría [4]. Se revisaron arquitecturas más amplias de control, como las basadas en orquestación flexible con ONOS y tecnologías de contenedores, las cuales permiten escalar servicios sobre redes ópticas mediante la integración de componentes ligeros y la automatización del control [5].

En relación al ámbito de la planificación de recursos, destacan algoritmos de asignación dinámica de ancho de banda que apuntan a priorizar el tráfico tanto por ONU, por tipo de tráfico, como otras opciones. Ejemplos de lo anterior son el algoritmos GATED [9], IPACT [10] y SR-DBA [11]. Aunque estos enfoques apuntan al mismo objetivo, difieren en sus métodos para asignar el ancho de banda. Por consiguiente, se utilizarán como punto de partida para evaluar esquemas existentes y desarrollar una solución que alcance o supere a las documentadas en la literatura.

1.3 Objetivos

1.3.1 Objetivo Principal

Desarrollar e implementar un simulador de redes de acceso ópticas para la gestión eficiente de recursos mediante el modelado, simulación e integración de topologías de red.

1.3.2 Objetivos Específicos

- **OE1:** Diseñar una arquitectura que influya en la asignación de recursos en redes PON, con el fin de representar adecuadamente escenarios realistas, mediante el desarrollo de un entorno de simulación que las integre de forma precisa.
- **OE2:** Integrar diferentes técnicas, incluyendo heurísticas - agentes de redes definidas por “software” e inteligencia artificial, para optimizar la asignación de recursos comparando con metodologías ya existentes.
- **OE3:** Integrar herramientas gráficas y textuales para visualizar topologías, parámetros de red y resultados de asignación mediante el uso de bibliotecas gráficas de Python.
- **OE4:** Validar el rendimiento del “simulador” para escenarios de prueba que reflejen condiciones reales de uso y carga contrastando con la literatura.

1.4 Estructura

La presente memoria se estructura de la siguiente forma:

- **Capítulo 1: Introducción.** Define el problema asociado a la rigidez en la gestión del canal ascendente en redes PON, establece los objetivos generales y específicos, y delimita el alcance de la solución.
- **Capítulo 2: Estado del Arte y Marco Teórico.** Proporciona los fundamentos de la arquitectura física PON y el mecanismo DBA básico. Asimismo, analiza los paradigmas de Redes Definidas por Software (SDN) y Aprendizaje por Refuerzo Profundo (DRL), concluyendo con una revisión crítica de soluciones similares en la literatura.
- **Capítulo 3: Diseño, arquitectura e implementación del simulador PonLab.** Detalla los requisitos funcionales y no funcionales del sistema, describe la arquitectura de software modular de la herramienta y define el modelo de navegación junto con el diseño de las interfaces gráficas y el escenario experimental.

- **Capítulo 4: Validación funcional y resultados experimentales.** Consolida la validación funcional de la plataforma y expone de forma integrada las mediciones operacionales (throughput, latencia y jitter) de los tres paradigmas evaluados (GIANT, SDN y Smart-RL), contrastando sus rendimientos bajo la perspectiva del compromiso entre volumen de datos y calidad de servicio (QoS).
- **Capítulo 5: Conclusiones.** Sintetiza los hallazgos principales en relación con los objetivos planteados, destaca los aportes metodológicos del framework modular y propone futuras líneas de investigación para extender las capacidades del simulador.

2. Estado del Arte y Marco Teórico

2.1 Antecedentes

El crecimiento sostenido del tráfico global de datos, impulsado por servicios de ultra alta definición y la proliferación de dispositivos interconectados, ha forzado la evolución de las redes de acceso [8]. Para satisfacer estas demandas, la industria ha transitado desde las arquitecturas **Gigabit PON (GPON)** hacia redes ópticas pasivas de próxima generación (**NG-PON**), como **XGS-PON** [12] y **50G-PON** [13]. Sin embargo, aprovechar al máximo la capacidad de estas redes se vuelve un problema cuando seguimos dependiendo de estrategias antiguas para gestionar el tráfico [3]. Para entender cómo dar el salto hacia soluciones más eficientes, este capítulo ordena la teoría en tres conceptos clave: cómo funciona la red óptica a nivel físico, cómo el paradigma de software (SDN) nos entrega un mayor control, y cómo podemos aplicar inteligencia artificial para optimizar el rendimiento general.

2.2 Infraestructura Física y Gestión de Recursos (PON y DBA)

Este primer concepto establece las bases físicas del medio de transmisión y los mecanismos clásicos para evitar la contención de tráfico.

2.2.1 Arquitectura de Redes Ópticas Pasivas (PON)

Las redes PON consisten en topologías de punto a multipunto donde la fibra óptica se despliega sin elementos activos en la ruta de la señal entre la fuente (proveedor) y el destino (usuario) [2]. Para hacer más evidente esta característica, es útil contrastarlas con las Redes Ópticas Activas (AON): mientras que las redes AON dependen de equipos electrónicos intermedios que requieren de un suministro de energía constante para conmutar o dirigir los datos, las redes PON utilizan exclusivamente dispositivos pasivos, como los divisores ópticos (splitters), que no necesitan alimentación eléctrica, lo que las hace considerablemente más eficientes [12]. Sus elementos principales son:

- **Terminal de Línea Óptica (OLT):** Ubicada en la oficina central, coordina las transmisiones

y asigna el ancho de banda [12].

- **Unidad de Red Óptica (ONU/ONT):** Dispositivo terminal en las instalaciones del usuario que agrupa el tráfico en contenedores lógicos denominados T-CONTs, los cuales poseen distintos requerimientos de Calidad de Servicio (QoS) [12].
- **Divisor Óptico (Splitter):** Es un dispositivo pasivo que forma parte de la Red de Distribución Óptica (ODN). Su función principal es conectar la fibra única proveniente de la OLT para distribuir la señal hacia múltiples ONUs en la dirección de bajada (downstream). Asimismo, en la dirección de subida (upstream), actúa como un acoplador (coupler) que recibe y agrupa las solicitudes y los datos enviados desde las diferentes ONUs para transmitirlos de regreso a la OLT [12].

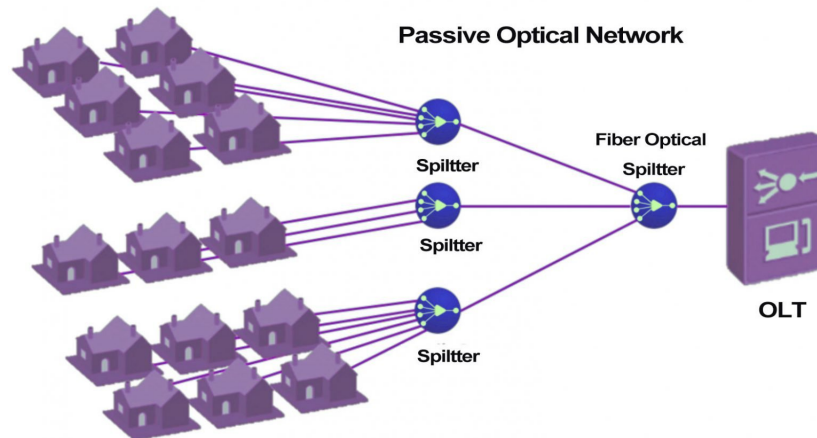


Figura 2.1: Diagrama de una Red Óptica Pasiva.

2.2.2 Asignación Dinámica de Ancho de Banda (DBA)

En las redes PON, mientras el tráfico descendente (de OLT a ONU) se transmite sin problemas por difusión, el canal ascendente es un medio compartido. Si múltiples usuarios (ONUs) intentan enviar datos simultáneamente, se producen colisiones. Para evitarlo, interviene el mecanismo **DBA**, un sistema de coordinación que asigna ventanas de tiempo específicas a cada equipo [2].

El funcionamiento del DBA depende de la interacción constante de tres elementos clave:

- **El Programador (Scheduler):** Ubicado en la OLT, es el motor lógico que centraliza las peticiones de los usuarios y toma las decisiones de reparto del ancho de banda [12].
- **Contenedores de Tráfico (T-CONT):** Unidades lógicas en las ONUs que agrupan los datos según su nivel de urgencia o prioridad (por ejemplo, separando el tráfico crítico de video de las descargas en segundo plano) [12].
- **Mensajes de Control:** La comunicación entre equipos se realiza mediante reportes de estado (mensajes REPORT) desde las ONUs, y mensajes de concesión o mapas de tráfico (mensajes GATE o BWmap) desde la OLT [11].

Ciclo Operativo del DBA

Este proceso de retroalimentación se ejecuta miles de veces por segundo a través de cuatro fases continuas:

1. **Reporte de Demanda:** Cada ONU revisa el estado de sus buffers (memorias temporales) y envía a la OLT un mensaje REPORT detallando cuánta información tiene acumulada en cada T-CONT [14].
2. **Cálculo de Asignación:** La OLT recopila los reportes de toda la red. Aquí, el algoritmo DBA evalúa la demanda total frente a la capacidad de la fibra y calcula qué porción de tiempo le corresponde a cada usuario, garantizando la estabilidad de los servicios críticos [14].
3. **Distribución de Turnos:** La central envía un mensaje GATE a las ONUs indicando el micro-segundo exacto en el que pueden comenzar a transmitir y la duración de su turno [14].
4. **Transmisión de Datos:** Finalmente, cada ONU dispara sus ráfagas de datos estrictamente en la ventana asignada y guarda silencio el resto del tiempo para evitar interferencias [14].

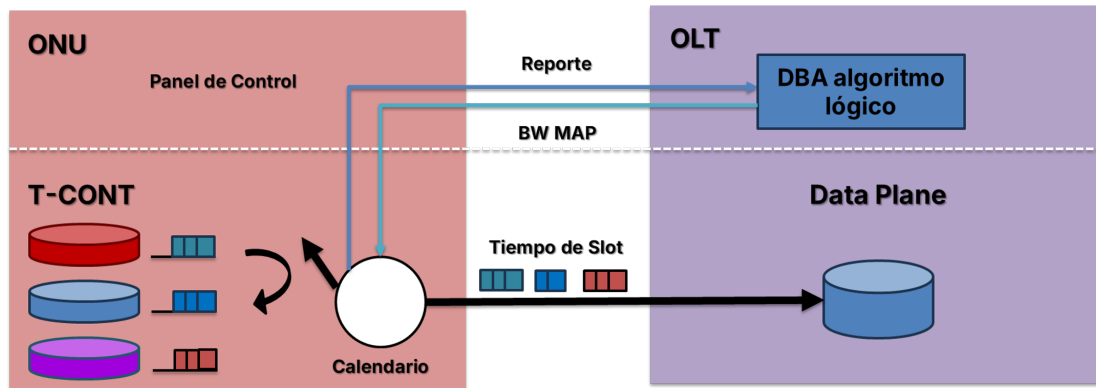


Figura 2.2: Diagrama de funcionamiento del DBA.

2.3 Paradigma de Control Abierto y Programable (SDN)

Para superar la rigidez del hardware de los algoritmos heurísticos tradicionales, la teoría de redes modernas introduce la separación estructural entre el reenvío de paquetes y la lógica de enrutamiento.

Redes Definidas por Software (SDN) en Redes Ópticas

El paradigma de Redes Definidas por Software abstrae la lógica de control desde los dispositivos de hardware hacia un ecosistema centralizado, programable y basado en software. Como se ilustra en la jerarquía de tres capas de la Figura 2.3, esta arquitectura desacopla las funciones de red mediante componentes especializados y puntos de interfaz normalizados:

- **Capa de Infraestructura (Infrastructure Layer / Plano de Datos):** Ubicada en la base de la arquitectura, contiene los elementos de conmutación físicos o emulados de la red de acceso óptico (como la OLT y las ONUs). Estos dispositivos pierden autonomía en la toma de decisiones globales y se limitan a ejecutar el reenvío de paquetes a alta velocidad a nivel de silicio, recolectando simultáneamente métricas locales y estados de colas (T-CONTs).

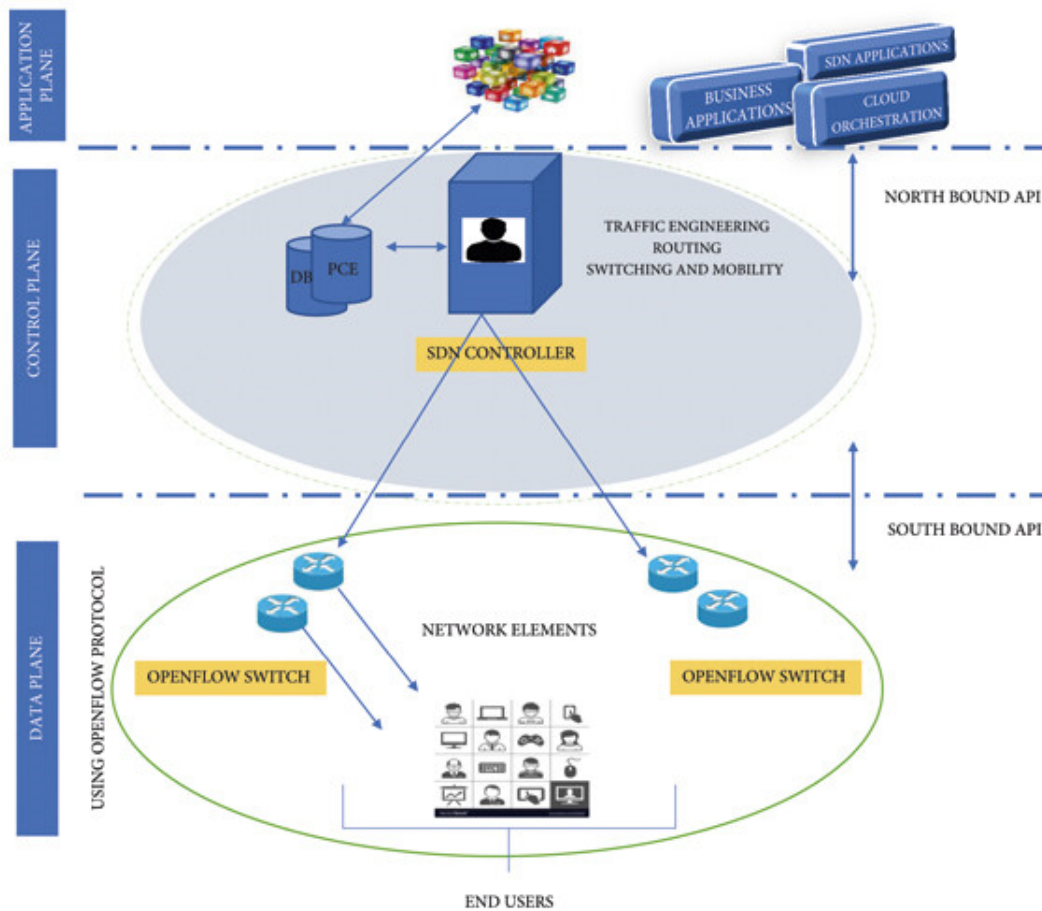


Figura 2.3: Arquitectura de Redes Definidas por Software [5].

- **Capa de Control (Control Layer / Plano de Control):** Actúa como el elemento central de orquestación. Aloja al Controlador SDN, el cual mantiene una visión global, abstracta y en tiempo real de la topología y el estado de los recursos de red. Este plano se comunica con la infraestructura inferior mediante la *Southbound Interface* (SBI) utilizando protocolos lógicos como **OpenFlow** [5] o interfaces de configuración orientadas a modelos como **YANG/NETCONF** [4], abstrayendo el hardware subyacente.
- **Capa de Aplicación (Application Layer / Plano de Aplicación):** Situada en la sección superior, define las políticas operacionales y la lógica de negocio a través de módulos de software independientes. La comunicación entre esta capa y el controlador se realiza mediante la *Northbound Interface* (NBI), comúnmente implementada a través de APIs de tipo

REST.

Interacción del Motor Inteligente: Es precisamente dentro de esta *Application Layer* donde se integra el agente inteligente basado en Aprendizaje por Refuerzo Profundo (DRL) [2.4]. El agente opera como una aplicación de red que consume la telemetría recolectada por el controlador a través de la NBI (representando el *estado* del entorno) y computa dinámicamente las políticas optimizadas de asignación de ancho de banda. Estas decisiones se transmiten de vuelta como *acciones* hacia la Capa de Control, la cual se encarga de traducirlas en mapas de concesión físicos ejecutables por la infraestructura de la red PON.

2.4 Motor de Optimización Inteligente

Con la flexibilidad otorgada por SDN (2.3), es posible abandonar la Heurística tradicional (2.2.2) y delegar la gestión del ancho de banda a la Inteligencia Artificial.

Aprendizaje por Refuerzo Profundo (DRL) para Gestión de Red

Es una técnica de inteligencia artificial que se formaliza matemáticamente a través de un **Proceso de Decisión de Márkov Finito (MDP)** [1]. En este marco idealizado, un **agente** aprende a tomar las mejores decisiones posibles mediante la interacción continua y en pasos de tiempo discretos con un **entorno**. En esta memoria, el DRL se integra como el “cerebro” del framework desarrollado, permitiendo que la red aprenda a gestionar sus recursos de forma autónoma, eliminando la necesidad de depender de modelos matemáticos explícitos del entorno [7], [15], [16] [4].

El funcionamiento del sistema se basa en un ciclo cerrado de retroalimentación compuesto por tres elementos clave en cada instante de tiempo t :

- **Estado (s_t):** Es la representación estructurada de la telemetría actual de la red. Proporciona al agente un modelado de cómo están distribuidos los recursos, detallando específicamente el nivel de llenado de los buffers de las ONUs. Esta información se concatena como un vector de entrada para evaluar la demanda antes de tomar una decisión [15].
- **Acción (a_t):** Es la decisión ejecutada por el agente en respuesta al estado de la red. Se define como la asignación exacta de recursos que dictamina qué porción de ancho de banda o ventana de transmisión se le otorgará a cada ONU en el siguiente ciclo operativo [15].

- **Recompensa (r_{t+1}):** Es una señal numérica de retroalimentación que el entorno devuelve en el paso siguiente como consecuencia de la acción tomada, para cuantificar su éxito. El sistema recibe un “premio” cuando logra mantener la latencia baja y un “castigo” cuando se producen retrasos o pérdida de datos [7], [15].

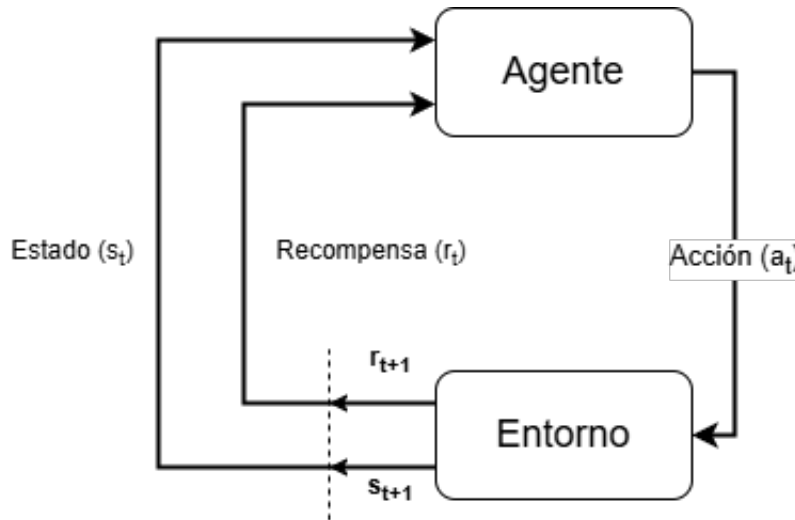


Figura 2.4: Diagrama interacción agente-entorno en el proceso de decisión de Markov [16]

A través de esta interacción, el objetivo del agente es descubrir una **política óptima** (π^*) que maximice el valor esperado de la suma acumulada de estas recompensas a lo largo del tiempo. A diferencia de los métodos tradicionales que usan reglas fijas, este enfoque permite que el simulador se adapte dinámicamente frente a patrones de tráfico impredecibles, optimizando la red según el comportamiento de los usuarios en tiempo real [15], [16].

2.5 Métricas de Evaluación y Calidad de Servicio (QoS)

Para establecer criterios de validación experimental en herramientas que integran los tres conceptos anteriores, se definen formalmente las variables independientes y dependientes del rendimiento de red:

- **Tasa de Transferencia (Throughput):** Se define formalmente como la tasa a la cual un sistema de destino recibe exitosamente los datos a través de una red, midiéndose típicamente

en bits por segundo (bps). Para evaluar el rendimiento general, el throughput promedio se calcula como la cantidad total de bits útiles (payload) recibidos con éxito dividida entre el tiempo total que tomó dicha transferencia [17].

Tomando como base esta definición teórica de Kurose que establece el rendimiento como el total de bits de un archivo (F) dividido por el tiempo de transferencia (T), para un entorno de transmisión por paquetes como PON, esta métrica se puede expresar mediante la siguiente ecuación fundamental:

$$Th = \frac{\sum_{i=1}^M P_i}{T} \quad (1)$$

Donde P_i es el tamaño en bits de la carga útil del paquete i transmitido con éxito hacia la ONU, M es el número total de paquetes recibidos, y T es el intervalo de tiempo total analizado. En conjunto, la maximización de este valor denota la utilización eficiente de la capacidad de línea del enlace óptico.

- **Latencia (Delay):** El diferencial de tiempo desde la llegada de un paquete al buffer de la ONU hasta su extracción exitosa en la OLT. Para servicios críticos modernos, la literatura establece un umbral objetivo inferior a 10 mili-segundos [18].
- **Jitter (IPDV):** El jitter se define como la varianza estadística en los tiempos de llegada de los paquetes, siendo una métrica crítica para garantizar la Calidad de Servicio (QoS) en aplicaciones sensibles al retardo, como el tráfico de voz y video.

En la asignación de recursos de redes PON, se clasifica en dos tipos principales:

1. **Intra-ventana:** Variación del retardo entre dos paquetes consecutivos de una misma ONU dentro de una única ventana de transmisión.
2. **Inter-ventana:** Variación del retardo del primer paquete emitido entre dos ventanas de transmisión consecutivas. Es el factor dominante para evaluar la distribución del retardo en tráfico isócrono.

Matemáticamente, el jitter inter-ventana $J_i(j)$ entre la ventana i y la ventana anterior ($i - 1$) de la ONU j se expresa como:

$$J_i(j) = D_i(j) - D_{i-1}(j) \quad (2)$$

Donde $D_i(j)$ representa el retardo experimentado por el primer paquete en la ventana i -ésima. El objetivo fundamental de los algoritmos DBA es minimizar esta variación para que la red mantenga una distribución de retardo centralizada, predecible e independiente de las fluctuaciones de carga del sistema [19].

2.6 Estado del Arte

La asignación dinámica de ancho de banda constituye uno de los principales desafíos operacionales en redes ópticas pasivas, debido a la naturaleza compartida del canal ascendente entre múltiples ONUs. A medida que las redes PON evolucionan hacia mayores capacidades de transmisión y soportan aplicaciones con requerimientos heterogéneos de calidad de servicio, la literatura ha propuesto diversas estrategias orientadas a optimizar la utilización del enlace, reducir la latencia y garantizar equidad entre usuarios.

Las soluciones desarrolladas pueden agruparse en cuatro grandes categorías.

1. **Algoritmos heurísticos tradicionales**, basados en reglas de planificación implementadas directamente en la OLT.
2. **Arquitecturas de redes definidas por software (SDN)**, los cuales permiten desacoplar la lógica de control y habilitar políticas dinámicas de asignación.
3. **Inteligencia artificial y aprendizaje automático** para anticipar condiciones de tráfico o tomar decisiones adaptativas en tiempo real.
4. **Entornos de simulación** han sido desarrollados para evaluar el desempeño de estas estrategias bajo escenarios controlados y reproducibles.

2.6.1 Evolución de las estrategias heurísticas para DBA

Los primeros enfoques se basan en algoritmos heurísticos implementados directamente en la OLT. Estos mecanismos toman decisiones a partir de reportes periódicos de ocupación de buffers

enviados por las ONUs y generan concesiones de transmisión para el siguiente ciclo operativo.

Algoritmo GIANT:

El algoritmo GigaPON Access Network (GIANT) [3], [12] fue el primer DBA introducido para GPON (en 2006) y utiliza un mecanismo de sondeo fijo. El algoritmo divide la asignación de recursos en dos fases: (a) la Asignación de Fase Garantizada (GPA) orientada a contenedores de tráfico prioritarios (TCONT 2 y 3), y (b) la Asignación de Fase Excedente (SPA) para tráfico de menor prioridad (TCONT 3 y 4).

Utiliza un modelo de programación tipo round-robin regido por un contador descendente (temporizador) que se resta en cada duración de trama. Cuando este contador expira, recién se asigna el ancho de banda a las clases de servicio.

Ventajas:

- Mejora la utilización del ancho de banda.
- Reduce los tiempos ociosos del canal ascendente mediante ciclos adaptativos.
- Mantiene una distribución equilibrada de recursos entre las ONUs activas.

Desventajas:

- Su rendimiento disminuye en escenarios con tráfico altamente variable.
- No incorpora mecanismos predictivos ni de aprendizaje.
- Presenta limitaciones para adaptarse dinámicamente a requerimientos complejos de QoS.

De acuerdo con la revisión de Memon et al. (2025) en [3] sobre las evaluaciones originales de Leligou et al.[20], GIANT funciona de manera óptima para cargas de tráfico inferiores al 50 %. Sin embargo, los estudios demuestran que la latencia se incrementa exponencialmente si la carga supera ese 50 %, y su utilización de ancho de banda se desploma al 60 % bajo tráfico en ráfagas.

Algoritmo IPACT:

Interleaved Polling with Adaptive Cycle Time (IPACT) [2], [3] es un programador “en línea” que se basa en un esquema de sondeo entrelazado. En este esquema, la OLT sondea a la siguiente ONU antes de que finalice (o siquiera llegue) la transmisión de la ONU sondeada previamente.

La OLT hace un seguimiento constante de los reportes (REQUEST) de las ONUs para calcular y conceder las siguientes ventanas de tiempo sobre la marcha, generando un ciclo de transmisión de tiempo variable (adaptativo).

Ventajas:

- Mejora la utilización del canal ascendente.
- Reduce la latencia respecto a esquemas DBA convencionales.
- Se adapta dinámicamente a la demanda de las ONUs.

Desventajas:

- Puede generar desigualdades en la asignación de recursos.
- Su rendimiento disminuye en escenarios de congestión severa.
- No incorpora mecanismos inteligentes de predicción.

De acuerdo con la revisión de Memon et al. en [3] sobre los análisis originales de Kramer et al., el algoritmo IPACT logra reducir el tiempo de inactividad del canal entre un 15 % y un 20 %. Adicionalmente, los resultados concluyen que, en un escenario con 32 ONUs, el algoritmo produce un retardo promedio de 12 ms.

Algoritmo GATED:

En el algoritmo Gated (disciplina de servicio “Gated”) [3], [9], el servidor de la OLT atiende únicamente a los paquetes que ya están esperando en la memoria de la ONU en el momento exacto en que ésta es sondeada. Es un método de programación de cálculo simple en el que la OLT concede el tamaño de ventana exactamente según la demanda que pide la ONU, sin aplicar cálculos compuestos, pero sujeto a un máximo predeterminado (por ejemplo, 50.000 bytes). Si la ONU necesita transmitir más información, debe enviar una nueva solicitud de concesión.

Ventajas:

- Implementación simple y fácil de gestionar.
- Garantiza una asignación ordenada de las transmisiones.
- Presenta bajo costo computacional.

Desventajas:

- Baja adaptación frente a cambios en la demanda de tráfico.
- Puede generar mayores tiempos de espera bajo alta carga.
- No considera diferenciación avanzada de QoS.

De acuerdo con la revisión de Memon et al. (2025) en [3] sobre los análisis de Rout, el algoritmo GATED logra mantener una alta eficiencia del 90 % bajo condiciones de tráfico estable. No obstante, los resultados evidencian que su utilización de ancho de banda cae drásticamente a un 70 % cuando la red experimenta tráfico en ráfagas (bursty traffic).

Algoritmo SR-DBA:

El Status Reporting DBA (SR-DBA) [3], [12] basa su funcionamiento en un proceso de retroalimentación explícita. Bajo este esquema, cada ONU envía regularmente información del estado y nivel de ocupación de sus memorias temporales (mediante mensajes de reporte o el campo DBRu) hacia la OLT. En base a estos reportes, la OLT calcula y asigna los mapas de ancho de banda (BWmap) dinámicamente de acuerdo con las demandas precisas y actuales de cada equipo.

Ventajas:

- Prioriza el tráfico según sus requerimientos de servicio.
- Mejora el soporte para aplicaciones sensibles al retardo.
- Permite una mejor gestión de la calidad de servicio (QoS).

Desventajas:

- Presenta una mayor complejidad de implementación.
- Requiere clasificación previa del tráfico.
- Puede incrementar la carga de procesamiento en la OLT.

De acuerdo con la literatura[3], [11], el esquema SR-DBA logra asignar el ancho de banda con alta precisión, pero a costa de añadir una sobrecarga (overhead) debido a la señalización constante de los reportes. Además, los resultados experimentales de Weng et al. en [11] concluyen que su latencia

es severamente afectada por la distancia: un alcance de 20 km genera un Retardo de Ecualización (EqD) de $200 \mu s$ que retrasa las respuestas de la OLT hasta en 3 tramas, empeorando el desempeño general frente a escenarios de distancias cortas (1 km) donde el EqD es de apenas $20 \mu s$.

Conclusiones respecto a las heurísticas.

Los algoritmos DBA revisados comparten el objetivo de asignar eficientemente el ancho de banda, pero se diferencian en sus criterios de decisión y en su adaptabilidad al tráfico dinámico, lo que impacta en métricas como throughput, latencia y QoS. Por ello, es pertinente comparar sus características para identificar limitaciones que han motivado la búsqueda de soluciones con SDN e inteligencia artificial. La Tabla 1 resume esta comparación.

Tabla 1: Resumen de características de algoritmos DBA.

Algoritmo	Equidad	QoS	Adaptabilidad	Complejidad	Latencia
GATED	Alta	Baja	Baja	Baja	Alta
IPACT	Media	Media	Media	Baja	Media
GIANT	Alta	Media	Media	Media	Media-Baja
SR-DBA	Media	Alta	Media	Media	Baja

2.6.2 Control programable mediante SDN para redes PON

Las limitaciones de los algoritmos DBA tradicionales han motivado la búsqueda de mecanismos que permitan una gestión más flexible y adaptable de los recursos en redes PON. En este contexto, las arquitecturas basadas en Redes Definidas por Software (SDN) han emergido como una alternativa para desacoplar las políticas de control de los dispositivos físicos de acceso, permitiendo modificar dinámicamente los criterios de asignación de ancho de banda sin intervenir directamente el hardware de la red.

Diversos trabajos han explorado la incorporación de controladores SDN en entornos PON con el

objetivo de mejorar la inter-operabilidad entre fabricantes, facilitar la automatización de la configuración de servicios y proporcionar una visión global del estado de la red para la toma de decisiones. Entre las propuestas más relevantes destacan los enfoques basados en modelos de información mediante **YANG/NETCONF** y las implementaciones que utilizan agentes **OpenFlow** para interactuar directamente con los equipos de acceso óptico.

Gestión programable basada en YANG/NETCONF

La investigación aborda la falta de capacidades inteligentes en la operación y mantenimiento de redes PON, causada por arquitecturas tradicionales cerradas y verticales que dependen de interfaces propietarias por fabricante, lo que ralentiza el despliegue de nuevos servicios y dificulta una gestión unificada.

Como solución, propone un controlador PON basado en SDN con diseño jerárquico (capas de configuración básica y de funciones de servicio), que emplea modelos YANG estandarizados vía NETCONF para abstraer el hardware de OLT y ONU, y utiliza telemetría (gRPC/UDP) para la recolección de datos en tiempo real con granularidad de segundos [4]. Este controlador permite configuración automática mediante un entorno gráfico sin código (codeless framework), monitoreo detallado de tráfico y estado, y gestión fina de recursos como VLANs y QoS, logrando visibilidad completa del ciclo de vida de los servicios [4].

Los resultados reportan mayor agilidad operativa y simplificación de la gestión, al romper la dependencia de proveedores mediante modelos YANG estándar que garantizan inter-operabilidad multi-fabricante, mientras que el enfoque sin código reduce tiempos y barreras de desarrollo, facilitando una automatización escalable con menores costos [4].

En cuanto al alcance, aunque el framework define estructuras para parámetros DBA en el modelo YANG, el controlador actúa en la capa de gestión y orquestación sin reemplazar la ejecución en tiempo real del algoritmo DBA en el hardware MAC de la OLT; sin embargo, no es meramente pasivo, pues integra un módulo de IA con modelos LSTM que procesa los datos de telemetría para diagnósticos de fallas, predicción de tráfico y generación de estrategias en un ciclo cerrado de control SDN [4].

Integración de agentes OpenFlow para redes PON

De Pintos, Merayo et al. en [8] abordan la complejidad en la gestión de las redes **XGS-PON** y la dependencia de sistemas propietarios e inflexibles, buscando resolver la dificultad de integrar soluciones SDN sin recurrir a hardware externo (como minicomputadoras) o a frameworks de abstracción excesivamente complejos con múltiples contenedores y dependencias de software (como el proyecto VOLTHA).

Como solución, proponen la implementación de un agente SDN basado en OpenFlow con un diseño de tres bloques (Adaptador ONOS, Capa de Traducción y Adaptador OLT) operado por un controlador SDN (ONOS). Este agente evita la emulación de capas SDN en los dispositivos de hardware; en su lugar, se comunica de forma directa con la API específica del chipset de la OLT mediante llamadas a procedimientos remotos (RPC/gRPC) y con las ONU mediante mensajes OMCI [8]. Esta arquitectura permite la configuración automática de servicios (como Internet con etiquetado VLAN simple o doble) a través de una interfaz de usuario intuitiva basada en menús, facilitando el monitoreo estadístico de flujos y una gestión fina de recursos que incluye perfiles de ancho de banda (CIR y PIR), puertos GEM y Alloc-IDs [8].

En cuanto a sus pruebas experimentales, los autores concluyen que se produce una drástica reducción en la complejidad de despliegue y una curva de aprendizaje mucho más corta frente a arquitecturas como SEBA/NEM. La arquitectura garantiza una alta inter-operabilidad multi-fabricante utilizando un código unificado, y demuestra una automatización veloz, logrando tiempos de configuración y despliegue de servicios en tiempo real de aproximadamente un segundo [8].

En lo respectivo a su alcance y limitaciones, el sistema es fundamentalmente una herramienta de gestión y configuración; delega la ejecución del algoritmo DBA al hardware de la OLT (limitándose a configurar los límites de las colas y los schedulers), y a diferencia de otras propuestas más modernas, no integra módulos de inteligencia artificial para la toma de decisiones autónomas o predictivas, manteniéndose como un sistema reactivo a las instrucciones del administrador o del controlador SDN [8].

Conclusiones de paradigmas de SDN

Los trabajos revisados evidencian una tendencia hacia el control programable en redes PON: las propuestas con YANG/NETCONF priorizan estandarización y automatización, mientras que las basadas en OpenFlow buscan la integración en ecosistemas SDN. Aunque ambas ofrecen plataformas flexibles para la asignación de recursos, ninguna optimiza dinámicamente el ancho de banda, lo que abre la puerta a mecanismos avanzados como inteligencia artificial y aprendizaje por refuerzo.

2.6.3 Aplicación de inteligencia artificial y aprendizaje por refuerzo en redes

Aunque las arquitecturas SDN ofrecen flexibilidad para monitorear y controlar redes PON, la asignación óptima de recursos sigue siendo un desafío abierto. Por ello, la investigación reciente ha incorporado inteligencia artificial para mejorar el uso del ancho de banda, reducir latencias y adaptarse a tráfico variable. Las propuestas se agrupan en tres enfoques: predicción de tráfico, aprendizaje supervisado y aprendizaje por refuerzo, cada uno con diferentes requisitos de entrenamiento, nivel de adaptación y capacidad para entornos dinámicos.

Predicción de tráfico para asignación de recursos

Este enfoque busca anticipar la demanda futura de ancho de banda mediante modelos predictivos entrenados con información histórica de tráfico. La predicción obtenida es posteriormente utilizada por algoritmos DBA convencionales para mejorar sus decisiones de asignación.

Deep-DBA: Predicción de Demanda mediante LSTM. Hatem et al. en [6] proponen Deep-DBA, un esquema predictivo que intenta estimar las demandas futuras de ancho de banda de las ONUs para evitar el intercambio constante de mensajes de control. Este método emplea aprendizaje profundo, específicamente Redes Neuronales Recurrentes (LSTM). Su resultado principal es la drástica reducción de la sobrecarga de control y tiempos de inactividad, mejorando la utilización del ancho de banda. No obstante, su limitación es el alto costo computacional y la pérdida de precisión a medida que aumentan los ciclos predecidos.

Controlador SDN Jerárquico con IA Predictiva. De acuerdo con la investigación en [4], se presenta un controlador SDN jerárquico que busca predecir el tráfico altamente variable y periódico de los puertos PON procesando datos de telemetría. Este enfoque emplea un modelo neuronal LSTM adaptado con características periódicas, logrando como resultado principal una alta precisión (sesgo de solo 7.3 %) con menor tiempo de cómputo frente a modelos lineales. Su principal limitación es que opera exclusivamente en la capa de orquestación (SDN), sin reemplazar la ejecución en tiempo real del DBA en el hardware de la OLT.

Revisión de Métodos Predictivos (MCP y DFDBA). Memon et al. en [3] analizan en su revisión diversos métodos predictivos (como MCP y DFDBA) que intentan estimar ranuras de tiempo futuras y demandas de tráfico basándose en ciclos pasados. Estas estrategias emplean desde modelado estadístico (búferes circulares) hasta modelos avanzados de inteligencia artificial, logrando como resultado principal reducir la latencia hasta un 40 % al ahorrar el tiempo de propagación de concesiones. Sin embargo, sus limitaciones incluyen la alta complejidad computacional y su falta de generalización para redes industriales diversas.

Aprendizaje supervisado aplicado a DBA

A diferencia de los métodos predictivos, los enfoques supervisados buscan aprender directamente relaciones entre estados de la red y decisiones de asignación a partir de conjuntos de datos previamente etiquetados.

Predicción de Latencia y Asignación mediante ANN. Ruan et al. en [18] emplean aprendizaje supervisado para predecir la latencia de subida resultante de distintas decisiones de asignación de ancho de banda. Este método utiliza una Red Neuronal Artificial (ANN) entrenada con múltiples características de la red (como carga de tráfico y distancias). Su resultado principal es la estimación de asignaciones óptimas que logran reducir significativamente la latencia frente a esquemas tradicionales. Su limitación principal radica en su dependencia del conocimiento previo de la red y el alto costo computacional (tiempo y memoria de CPU) requerido para el entrenamiento.

Aprendizaje por refuerzo para asignación DBA

La línea de investigación más reciente busca eliminar la dependencia de modelos predictivos o datos etiquetados mediante agentes capaces de aprender directamente de la interacción con la red. En este enfoque, la asignación de ancho de banda se formula como un problema secuencial de toma de decisiones donde el agente optimiza métricas de rendimiento a partir de recompensas obtenidas durante la operación.

Asignación Adaptativa con Multi-Armed Bandit Ruan et al. en [18] formulan la asignación de ancho de banda como un problema multi-armed bandit para estimar decisiones óptimas sin conocimiento previo de la red. Emplean métodos como Q-learning, SARSA y Diferencia Temporal (TD). Su resultado principal es una reducción de latencia de subida de hasta un 50 %. Su limitación radica en que la fase inicial de exploración genera altas desviaciones de latencia y la tasa de aprendizaje exige un ajuste manual previo.

Gestión Inteligente de Latencia con SARSA Zhou et al. en [15] buscan actualizar pro-activamente las políticas de ancho de banda máximo para garantizar latencias estrictas. Utilizan el algoritmo SARSA interactuando con el entorno PON para actualizar una tabla de decisiones (Q-table). Como resultado principal, logran gestionar la latencia promedio manteniéndola por debajo de 1 ms frente a cargas dinámicas. Su limitación es que el agente opera en una capa superior asíncrona (no en tiempo real), dependiendo del DBA rápido subyacente.

Optimización PPO para Entornos Industriales Cao et al. en [7] intentan ajustar dinámicamente los pesos y prioridades de los servicios para redes industriales 50G-PON. Emplean el algoritmo de Optimización de Políticas Próximas (PPO) evaluando factores como la recompensa de los usuarios. Su resultado es un aumento del 25 % en el rendimiento (throughput) y una reducción del 33 % en la latencia bajo cargas altas. Su limitación es la dependencia constante de bucles de retroalimentación en tiempo real y su consumo de memoria.

Asignación Multi-banda en Redes Ópticas Elásticas (EON) Morales et al. en [21] abordan el problema de asignación de enrutamiento, banda, modulación y espectro (RBMLSA). Evalúan seis

agentes en un entorno simulado, empleando algoritmos como TRPO, PPO2 y DQN. Su resultado principal es una reducción significativa en la probabilidad de bloqueo utilizando TRPO, el cual demostró la mayor estabilidad. Su limitación es que la expansión a múltiples bandas genera un espacio de observación masivo, exigiendo redes neuronales muy complejas.

Conclusiones de la aplicaciones de aprendizaje de máquina.

En términos generales, las investigaciones revisadas evidencian una transición desde enfoques orientados a la predicción y estimación del comportamiento de la red hacia mecanismos capaces de tomar decisiones de asignación de recursos de forma autónoma. Mientras los métodos predictivos y supervisados dependen de información histórica o conjuntos de entrenamiento previamente definidos, las técnicas de aprendizaje por refuerzo permiten adaptar dinámicamente las políticas de asignación en función del estado actual de la red. Esta capacidad de aprendizaje continuo ha convertido al aprendizaje por refuerzo en una de las líneas más prometedoras para la optimización del ancho de banda en redes PON, aunque su validación continúa dependiendo de entornos de simulación y experimentación adecuados.

Tabla 2: Tabla resumen de enfoques de aprendizaje por maquina en redes

Enfoque	Objetivo	Info. requerida	Adapt.	Limitación principal
Predicción tráfico	Anticipar demanda	Datos históricos	Media	Requiere DBA externo
Aprendizaje supervisado	Estimar asignaciones	Datos etiquetados	Media	Dependencia del entrenamiento
Aprendizaje por refuerzo	Optimizar recursos	Interacción entorno	Alta	Alto costo exploración

2.6.4 Simuladores y plataformas experimentales para PON

La validación de algoritmos DBA, arquitecturas SDN y técnicas de inteligencia artificial requiere entornos capaces de reproducir el comportamiento de redes de acceso óptico bajo condiciones controladas. En este contexto, la literatura ha utilizado principalmente dos tipos de herramientas: simuladores de propósito general orientados al análisis de protocolos y rendimiento de red, y plataformas especializadas en redes definidas por software enfocadas en la evaluación de mecanismos

de control programable. Cada enfoque presenta ventajas y limitaciones en términos de realismo, flexibilidad y capacidad para integrar nuevas estrategias de gestión de recursos.

Simuladores de propósito general

NS-3 es un simulador de eventos discretos ampliamente utilizado en investigación de redes por su código abierto y modelos detallados de protocolos. Diversos trabajos sobre PON y DBA lo han empleado para evaluar throughput, latencia y utilización de ancho de banda bajo distintas condiciones de tráfico, gracias a la precisión de sus modelos y la capacidad de reproducir escenarios complejos de forma controlada.

Wu et al. en [22] proponen el primer módulo XG-PON para el simulador NS-3 con el objetivo de evaluar el rendimiento de la red y su interacción con protocolos superiores, motivados por la carencia de simuladores rápidos, de código abierto y con pilas de protocolos completas. Como resultado principal, desarrollaron una herramienta de simulación plenamente funcional y configurable orientada a la capa de control MAC, pero su limitación identificada radica en la simplificación extrema de la capa física, ya que omite los retardos diferenciales reales de la propagación de señales ópticas.

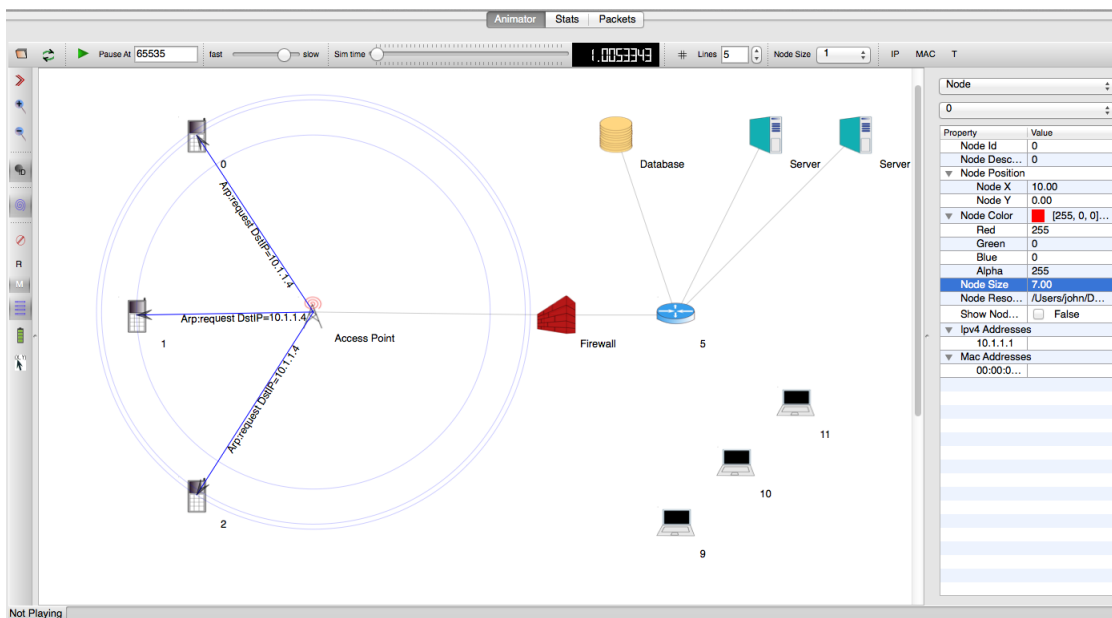


Figura 2.5: Interfaz de simulador NS-3 (imagen referencial)

OMNet++: corresponde a un simulador modular orientado a eventos discretos ampliamente utili-

zado en investigación académica y desarrollo de redes de comunicaciones. Su arquitectura basada en componentes reutilizables ha permitido la implementación de modelos específicos para redes ópticas y entornos experimentales enfocados en la evaluación de protocolos y mecanismos de gestión de recursos.

Hatem et al. en [6] emplean el simulador OMNeT++ con el objetivo de generar conjuntos masivos de datos de entrenamiento y validar su esquema predictivo Deep-DBA, motivados por la necesidad de evaluar con precisión la reducción de la sobrecarga de mensajes de control en redes NG-EPON. Como resultado principal, las simulaciones demostraron una drástica disminución del tiempo de inactividad manteniendo un rendimiento similar a los esquemas en línea, aunque su limitación identificada radica en el alto costo computacional de entrenamiento y el desperdicio de ancho de banda ante predicciones incorrectas en ciclos extendidos.

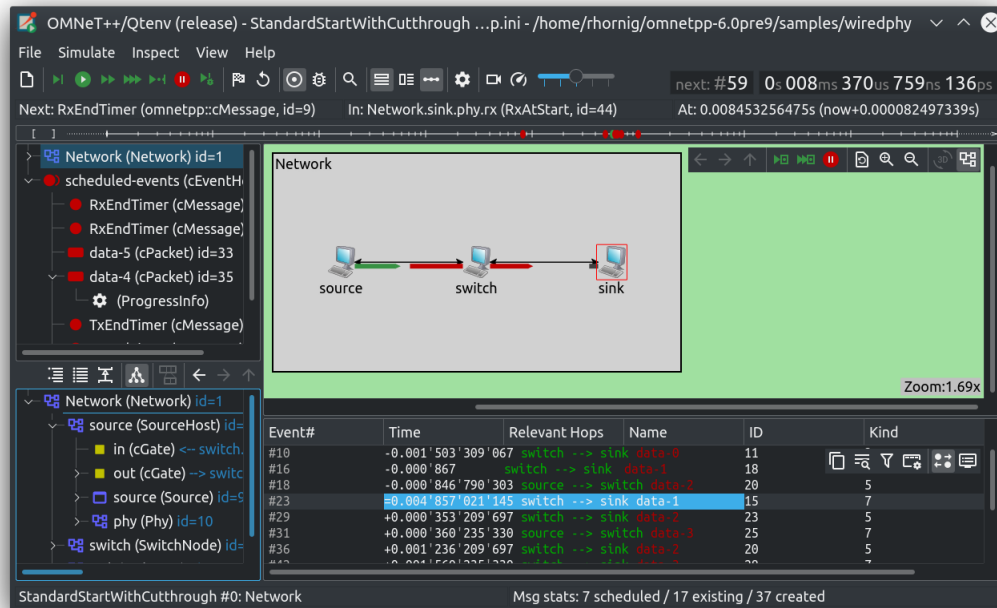


Figura 2.6: Interfaz de simulador OMNeT++ (imagen referencial)

OPNET: (actualmente comercializado como Riverbed Modeler) ha sido ampliamente utilizado en entornos industriales y académicos para evaluar el desempeño de arquitecturas de red y protocolos de comunicación. Gracias a sus bibliotecas predefinidas y herramientas de análisis, diversos estudios han empleado esta plataforma para analizar comportamiento de tráfico, calidad de servicio y

planificación de recursos en redes de acceso.

Cao et al. en [7] emplean el simulador OPNET con el objetivo de verificar el rendimiento de su algoritmo MF-DWBA y evaluar la satisfacción del usuario, motivados por la necesidad de emular un entorno industrial 50G-PON a gran escala (hasta 640 ONUs) de manera realista. Como resultado principal, las simulaciones demostraron que el sistema logra mantener latencias ultrabajas ($1-10 \mu s$) y mejora la satisfacción en un 15 % bajo cargas altas, aunque su limitación identificada radica en que, ante un aumento repentino y excesivo de solicitudes, la red aún sufre degradación debido a las restricciones físicas y de capacidad de procesamiento del hardware de la OLT.

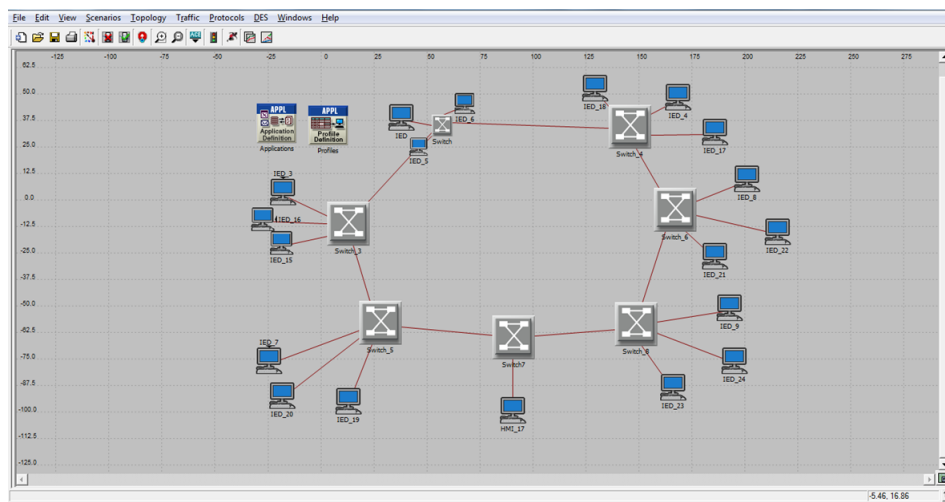


Figura 2.7: Interfaz de simulador OPNET (imagen referencial)

Plataformas para redes definidas por software

MiniNet: es una plataforma de emulación ampliamente utilizada para la investigación en redes definidas por software. Diversos trabajos orientados a SDN emplean esta herramienta para desplegar topologías virtuales compuestas por switches OpenFlow, hosts y controladores, permitiendo validar políticas de control y mecanismos de orquestación antes de su implementación en infraestructura real.

Lantz et al. en [23] emplean el emulador Mininet-Optical con el objetivo de emular interactivamente redes ópticas de paquetes controladas por SDN de extremo a extremo, motivados por la necesidad de una herramienta de código abierto para sistemas desagregados frente a los costosos y limitados bancos de pruebas de hardware. Como resultado principal, desarrollaron una plataforma virtual

que integra modelos de deterioro físico (como OSNR) y se conecta exitosamente al controlador comercial ONOS, aunque su limitación radica en que opera a bajas tasas de datos reales y aún requiere validar su escalabilidad frente a matrices de tráfico y cantidades de dispositivos realistas.

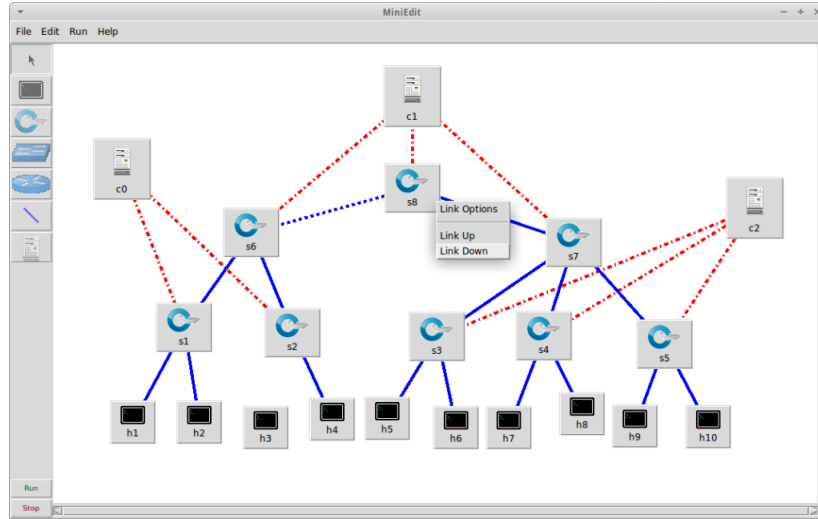


Figura 2.8: Interfaz de simulador MiniNet (imagen referencial)

Características y Limitaciones de los entornos en redes PON.

Los trabajos revisados muestran que tanto los simuladores de propósito general (NS-3, OMNeT++, OPNET) como las plataformas orientadas a SDN (Mininet) son herramientas clave para validar propuestas en redes ópticas. Los primeros permiten modelar protocolos, tráfico y mecanismos DBA con distintos niveles de detalle, mientras que Mininet facilita la experimentación con control programable y controladores SDN. Sin embargo, cada herramienta está orientada a objetivos específicos y presenta limitaciones para representar simultáneamente todos los elementos involucrados en la gestión inteligente de redes PON, como se resume en la Tabla 3.

Tabla 3: Características de entornos de simulación para redes.

Herramienta	Tipo	PON	SDN	IA/RL	Principal ventaja	Principal limitación
NS-3	Simulación	Sí	Parcial	Parcial	Modelado detallado de protocolos	Simplificación de ciertos aspectos físicos
OMNeT++	Simulación	Sí	Parcial	Sí	Alta modularidad y extensibilidad	Mayor complejidad de implementación
OPNET	Simulación	Sí	Limitado	Sí	Entorno maduro y modelos predefinidos	Plataforma propietaria
Mininet	Emulación	Limitado	Nativo	Parcial	Integración nativa con SDN	No modela DBA ni PON de forma nativa

2.6.5 Brecha Identificada y Oportunidad de Investigación

Brechas identificadas.

A partir del análisis crítico de la literatura, se han detectado cuatro brechas metodológicas y tecnológicas fundamentales en el ámbito de la simulación y evaluación de estrategias de Asignación Dinámica de Ancho de Banda (DBA) para redes PON:

1. **Fragmentación metodológica para la evaluación cruzada de paradigmas:** La literatura sobre DBA se ha desarrollado en silos metodológicos. Las heurísticas y las arquitecturas SDN se evalúan predominantemente en simuladores de eventos discretos (NS-3, OMNeT++) orientados a la capa de enlace, mientras que los algoritmos de Aprendizaje por Refuerzo (RL) se entrenan y prueban en entornos de propósito general (como Gymnasium) que abstraen o simplifican en exceso los protocolos de control de acceso. Esta fragmentación impide una comparación justa, sistemática y reproducible entre paradigmas, ya que las métricas de rendimiento (latencia, throughput, equidad) se obtienen bajo supuestos y condiciones de simulación radicalmente diferentes.
2. **Sobrecosto computacional innecesario de la capa física para problemas de control:** Si bien los simuladores de red tradicionales ofrecen alta fidelidad al modelar fenómenos físicos (atenuación, OSNR, etc.), esta complejidad resulta superflua y contraproducente para el estudio del problema lógico del DBA. La dinámica del DBA se define en la capa MAC, gobernada por temporizadores, concesiones (grants), reportes (reports) y la gestión de colas (T-CONT). La inclusión de la capa física en el bucle de simulación, especialmente en el entrenamiento de agentes de RL que requieren millones de episodios, añade un costo computacional desmedido que ralentiza la experimentación y la convergencia, convirtiéndola en un cuello de botella para la innovación en este campo.
3. **Limitada interoperabilidad y falta de abstracción visual en entornos de simulación:** La configuración y operación de las plataformas existentes es eminentemente técnica y basada en scripts (archivos .cc, .ned o .py), lo que impone una barrera de entrada significativa. La

carencia de una interfaz gráfica interactiva que permita la manipulación dinámica de topologías, el intercambio “en caliente” de algoritmos de asignación y la visualización en tiempo real de métricas de rendimiento (e.g., evolución de colas, latencia por ONU, asignación de BWmap) limita la agilidad en la fase de investigación y codiseño, dificultando la depuración, la docencia y la transferencia tecnológica.

4. **Ausencia de un punto de referencia (benchmark) para la evaluación de agentes inteligentes:** La comunidad de RL aplicada a DBA carece de un entorno de pruebas estandarizado que defina un conjunto de escenarios de tráfico (carga, patrones, número de ONUs) y un protocolo de evaluación claro. Esto dificulta la comparación objetiva de los diferentes algoritmos (PPO, DQN, SARSA, etc.) reportados en la literatura, ya que cada autor utiliza su propia implementación y condiciones de prueba ad-hoc.

Oportunidad Identificada: Un Simulador Unificado y Ágil para DBA en PON

Las limitaciones descritas revelan una oportunidad clara para el desarrollo una nueva herramienta de software que sirva como nexo entre los diferentes paradigmas de investigación. Esta propuesta consiste en el diseño, implementación y validación de un simulador de eventos discretos, modular y con interfaz visual, centrado exclusivamente en la lógica de control de la capa MAC de las redes PON.

La esencia de la solución reside en tres pilares fundamentales que abordan directamente las brechas identificadas:

1. **Motor de Eventos PON:** Un núcleo de simulación que abstrae por completo la capa física, modelando los ciclos de sondeo, la generación de tráfico a nivel de ráfaga, la estructura del BWmap y el comportamiento de las colas (T-CONT). Esto garantiza una ejecución rápida, haciendo viable el entrenamiento intensivo de agentes de RL.
2. **Interfaz de Programación Unificada:** Un conjunto de herramientas estandarizadas que permitan inyectar y evaluar en un mismo ecosistema: **algoritmos heurísticos** (clásicos como IPACT, GIANT), **políticas de control programable** (simulando un controlador SDN que modifica parámetros de concesión) y **agentes de aprendizaje por refuerzo**.

3. **Plataforma Visual Interactiva:** Una interfaz gráfica (GUI) que permite la configuración visual de topologías (número de ONUs, distancias, perfiles de tráfico), el monitoreo en tiempo real de métricas de desempeño (throughput, latencia, fairness) y la visualización de los resultados en un formato estandarizado que pueda servir como benchmark para la comunidad.

3. Diseño, arquitectura e implementación del simulador Pon-Lab

En este capítulo se detalla el proceso de diseño y desarrollo del marco del simulador. Para ello, se aborda formalmente la especificación de los requisitos del sistema y el modelado de los perfiles de usuario que interactúan con la plataforma. Asimismo, se expone la arquitectura modular del software, la especificación de sus componentes internos, la estructura del modelo de navegación e interfaces gráficas, y finalmente, la metodología experimental considerada para su validación.

3.1 Requisitos del Sistema

A continuación se presentan los requisitos funcionales y no funcionales del sistema que están directamente vinculados a las necesidades a cumplir del simulador.

3.1.1 Requisitos funcionales

- **RF1:** El “simulador” debe estar enfocado principalmente en resolver problemas de asignación de ancho de banda.
- **RF2:** El “simulador” debe permitir modelar y diseñar topologías de redes ópticas.
- **RF3:** El “simulador” debe permitir incluir dos tipos de dispositivos principales: OLT (Optical Line Terminal) y ONU (Optical Network Unit).
- **RF4:** El “simulador” debe permitir modificar los parámetros de funcionamiento de las OLT.
- **RF5:** El “simulador” debe permitir modificar los parámetros de funcionamiento de las ONU.
- **RF6:** El “simulador” debe permitir la integración e implementación de diferentes algoritmos de asignación de recursos.
- **RF7:** El “simulador” debe permitir la visualización de los resultados de la asignación de recursos de forma gráfica o textual.

3.1.2 Requisitos no funcionales

- **RNF1:** El “simulador” debe tener una interfaz gráfica que permita al usuario crear y editar topologías de red de forma intuitiva.
- **RNF2:** El “simulador” debe ser capaz de ejecutarse en sistemas operativos basados en Linux o Windows.
- **RNF3:** El “simulador” debe permitir agregar nuevos algoritmos DBA sin reestructurar la arquitectura principal.
- **RNF4:** El código del “simulador” debe estar modularizado y documentado para permitir su modificación y extensión por parte de otros desarrolladores.
- **RNF5:** El “simulador” debe permitir la importación y exportación de topologías de red, facilitando así la reutilización y la portabilidad de recursos del sistema.

3.1.3 Requisitos de interfaces

A continuación, se detallan las relaciones funcionales entre los estímulos externos y las acciones de control del sistema. La Tabla 4 clasifica los eventos detectados junto con las respuestas del sistema correspondientes, estableciendo la lógica de activación. Por su parte, la Tabla 5 complementa esta estructura especificando los parámetros de entrada y salida asociados a cada respuesta, delimitando así la estructura de datos requerida para su correcta ejecución e integración.

3.1.4 Entorno de desarrollo y ejecución

En esta sección se especifican las plataformas tecnológicas y herramientas necesarias para la construcción y operación del sistema. El entorno de desarrollo agrupa los lenguajes, entornos de desarrollo integrado (IDE), bibliotecas y herramientas de control de versiones empleadas para el diseño y pruebas del software. Por otro lado, el entorno de ejecución define la infraestructura de hardware, sistema operativo y dependencias requeridas en el entorno de producción para garantizar el rendimiento, escalabilidad y correcta inferencia del sistema.

Tabla 4: Eventos y respuestas del sistema

Evento	Descripción	Iniciador	Parámetros	Respuesta
Crear Topología	El usuario inicia el diseño de una nueva red óptica.	Usuario	Nombre de la topología, tipo de red	Se crea un nuevo espacio de diseño en blanco
Agregar Nodo Tipo (OLT/ONU)	El usuario inserta una OLT/ONU en la topología.	Usuario	Puerto, velocidad del puerto, ID, tipo de tecnología	La OLT/ONU se visualiza en la interfaz gráfica.
Modificar parámetros de dispositivos	El usuario puede modificar los parámetros de los nodos dentro de la topología.	Usuario	Potencia de Transmisión, ID nodo, capacidad, demanda, consumo	Se actualizan los datos del nodo.
Aplicar algoritmo de asignación	Selecciona e implementa un algoritmo de asignación de recursos.	Usuario	Tipo de algoritmo, configuración de parámetros	El algoritmo se ejecuta y distribuye el ancho de banda según su desarrollo.
Visualizar resultados de asignación	Muestra distribución del ancho de banda en el tiempo y eficiencia del algoritmo.	Sistema	Topología activa, algoritmo seleccionado	Se despliega un gráfico o tabla con resultados de asignación.

Tabla 5: Respuestas del sistema y sus parámetros

Respuesta	Descripción	Parámetros
Crear nueva topología	Inicializa un espacio en blanco para diseñar una red óptica	Nombre de la topología, tipo de red
Insertar Nodo Tipo (OLT/ONU)	Agrega un nuevo dispositivo OLT/ONU a la vista gráfica	ID de OLT/ONU, ubicación, parámetros iniciales
Actualizar parámetros de OLT/ONU	Cambia los valores de configuración de una OLT/ONU existente	ID de OLT/ONU, nuevos parámetros
Ejecutar algoritmo de asignación	Ejecuta el algoritmo seleccionado para repartir el ancho de banda	Tipo de algoritmo, configuración inicial
Mostrar resultados de asignación	Visualiza métricas de desempeño y distribución de ancho de banda	Topología activa, algoritmo aplicado

Hardware de Desarrollo

- Servidor de simulación o computador. Sistema operativo utilizado **Windows10 - Windows11**, procesador de **5 núcleos o superior**, memoria **o superior**.

Software de Desarrollo

- **Simulador de redes:** para emular topologías y validar algoritmos antes de su implementación real.
- **GitHub:** plataforma para control de versiones, colaboración y documentación del desarrollo.
- **Python 3.13.3:** Lenguaje principal para el desarrollo del simulador y de la interfaz visual.
- **Bibliotecas utilizadas:**
 - Pandas: Manejo de datos.
 - NumPy: Manejo de datos.
 - SciPy: Manipulación y análisis de datos.
 - Seaborn: Visualización de datos estadísticos.
 - Matplotlib: Visualización de gráficas.
 - PyQt5: “Framework” para construcción de interfaz.
 - Gymnasium: Entorno de simulación para entrenar agentes de aprendizaje por refuerzo.

3.2 Perfiles de usuario

Con el objetivo de orientar el diseño de las interfaces y definir los requisitos funcionales según el nivel de especialización requerido, se han establecido dos perfiles de usuario diferenciados. En primer lugar, la Tabla 6 sintetiza las características socio-demográficas, operativas y cognitivas del personal técnico capacitado para operar el simulador en entornos XGS-PON. En segundo lugar, la Tabla 7 describe al consumidor final del servicio, definiendo un perfil heterogéneo cuyas expectativas se centran en la continuidad del servicio y la simplicidad de interacción.

Tabla 6: Perfil de usuario: OLT (Proveedor de Servicio)

Perfil: OLT (Proveedor de Servicio)	
Socio-económico y cultural	Profesional con formación técnica universitaria, perteneciente a un grupo de nivel medio-alto en el ámbito tecnológico.
Ocupacional	Ingeniero/a de telecomunicaciones, administrador/a de redes, técnico/a de ISP.
Etario	25 – 55 años
Características físicas, fisiológicas, psicológicas	Buen estado cognitivo, capacidad de análisis técnico, toma de decisiones, trabajo bajo presión.
Otros	Usuario del simulador para redes XGS-PON. Requiere interfaz técnica clara y soporte a protocolos estándar.

Tabla 7: Perfil de usuario: ONU (Usuario Final)

Perfil: ONU (Usuario Final)	
Socio-económico y cultural	Usuarios residenciales o comerciales, de nivel socioeconómico diverso. Usuarios de servicios de telecomunicaciones.
Ocupacional	No requiere ocupación específica; incluye desde trabajadores/as a estudiantes, pymes o familias.
Etario	18 – 75 años (en general)
Características físicas, fisiológicas, psicológicas	Varía ampliamente. Requiere experiencia mínima con tecnologías digitales. Valoración alta de estabilidad del servicio y velocidad de red.
Otros	Espera servicios eficientes, sin necesidad de interacción técnica. Perfil enfocado en experiencia de usuario.

3.3 Diseño de Arquitectura del Sistema

En esta sección se describe la arquitectura general del sistema, definiendo la estructura técnica, los componentes y la dinámica de interacción necesaria para satisfacer los requerimientos del proyecto. En primer lugar, se expone el flujo general de funcionamiento del sistema junto con su diagrama de arquitectura, ofreciendo una visión integral del procesamiento de información. Posteriormente, se desglosan los módulos de arquitectura que componen la solución y se presenta la matriz de requisitos funcionales y módulos, estableciendo la trazabilidad entre las funcionalidades exigidas y su implementación modular. Finalmente, se especifican las variables, parámetros y salidas clave que rigen el intercambio de datos y el comportamiento interno del sistema.

3.3.1 Flujo general de funcionamiento del sistema

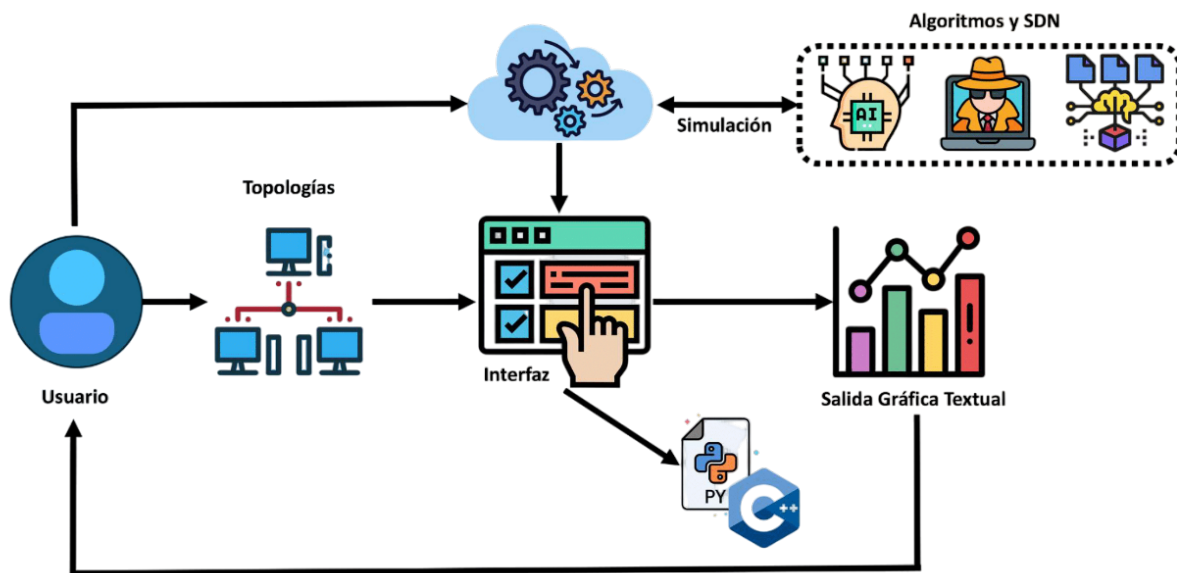


Figura 3.1: Diagrama de flujo general.

En el diagrama 3.1, las flechas indican los flujos de trabajo que entran y salen del sistema, mostrando así las interacciones principales:

- **Usuario:** Es el agente externo principal que interactúa con el sistema. En este proyecto, el usuario puede ser tanto un **proveedor de servicios** de red como un **investigador o desarrollador** interesado en evaluar, diseñar o mejorar esquemas de asignación de recursos en

redes ópticas pasivas. Define los parámetros del experimento, configura las topologías de red y selecciona los algoritmos que desea aplicar . Su interacción ocurre a través de una interfaz gráfica.

- **Topologías:** El usuario puede seleccionar o construir topologías de red. Estas estructuras son luego enviadas al núcleo del sistema para ser utilizadas en la simulación.
- **Interfaz:** Es el módulo central de interacción. recibe las configuraciones, las estructura y las envía a la capa de simulación. También canaliza los resultados hacia la salida gráfica. Se visualiza la conexión con los lenguajes de implementación utilizados.
- **Simulación:** Aquí se ejecuta la lógica del sistema, integrando las topologías, configuraciones y los algoritmos de asignación. La nube de engranajes simboliza el entorno de procesamiento.
- **Algoritmos y SDN:** Esta sección representa los distintos enfoques de asignación de recursos implementados en el sistema, incluyendo algoritmos heurísticos, aprendizaje reforzado y controladores SDN.
- **Salida Gráfica/Textual:** Una vez realizada la simulación, los resultados se presentan visualmente mediante gráficos y métricas. Esto permite al usuario analizar el comportamiento del sistema de forma comprensible y práctica.

3.3.2 Diagrama de Arquitectura

La Figura 3.2 presenta los módulos internos que componen el sistema propuesto para la investigación en redes ópticas. Cada módulo cumple una función especializada dentro del flujo de procesamiento de datos, desde la definición de topologías por parte del usuario hasta la entrega de resultados visuales y análisis de desempeño.

Módulos de Arquitectura

La Tabla 8 describe los módulos mostrados en la Figura 3.2. Cada módulo es presentado junto con su función principal y la sección del documento donde se detalla su implementación.

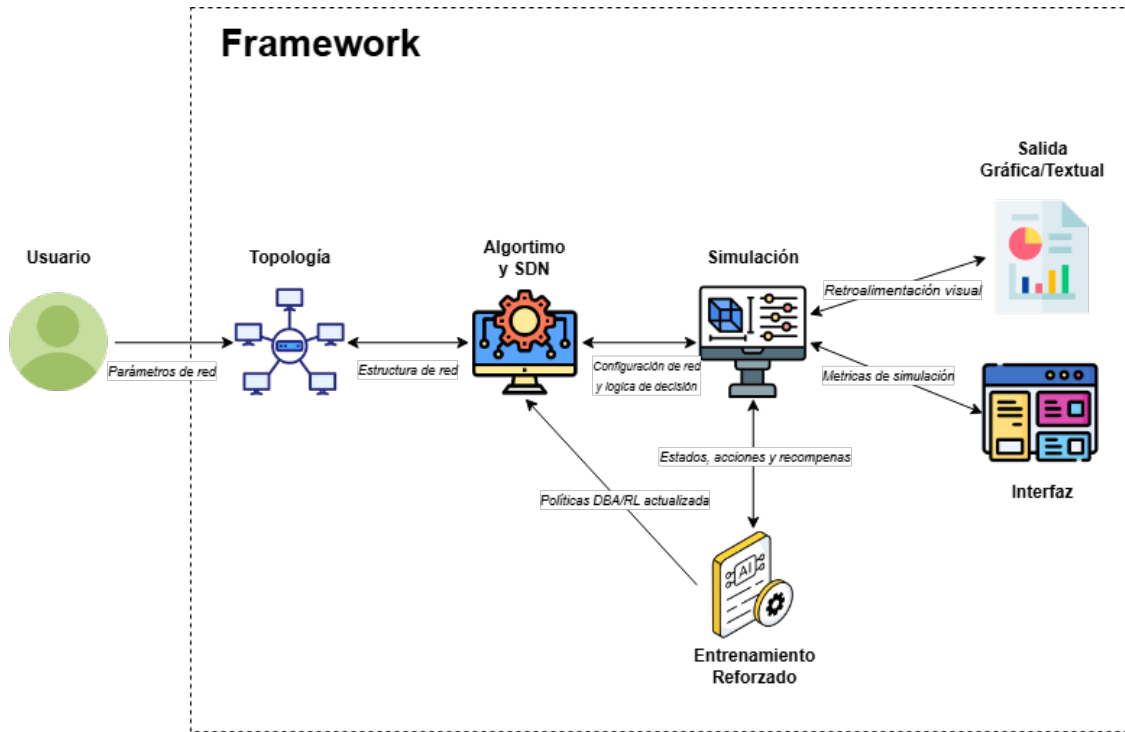


Figura 3.2: Diagrama de arquitectura.

Tabla 8: Enumeración de módulos internos del sistema.

Módulo	Nombre	Descripción
M1	Usuario	Punto de inicio e interacción. Permite al investigador definir parámetros de simulación y topologías de red.
M2	Topologías	Generador de topologías personalizadas o carga de configuraciones predefinidas.
M3	Interfaz	Entorno visual que permite configurar la simulación, visualizar resultados y controlar el flujo del sistema.
M4	Simulación	Núcleo computacional donde se ejecuta la lógica de asignación de recursos, incorporando el algoritmo DBA y el controlador SDN.
M5	Algoritmos y SDN	Componente donde se integran heurísticas, reglas de control y programación SDN (OpenFlow).
M6	Salida Gráfica/Textual	Presenta los resultados de la simulación mediante gráficos, métricas numéricas y análisis textual.
M7	Entrenamiento reforzado	Gestiona el proceso de entrenamiento del agente de aprendizaje reforzado, implementando algoritmos RL (Q-learning, SARSA, PPO) para optimizar la asignación dinámica de ancho de banda.

Matriz de Requisitos Funcionales y Módulos

La Tabla 9 presenta la trazabilidad entre los requisitos funcionales del sistema (3.1.1) y los módulos definidos en la arquitectura (3.3.2). Se indica con una **X** en qué módulo se implementa o contribuye cada requisito.

Tabla 9: Matriz de trazabilidad entre requisitos funcionales y módulos del sistema.

RF	Descripción	M1	M2	M3	M4	M5	M6	M7
RF1	Resolución de problemas de asignación de ancho de banda.		X		X	X		X
RF2	Modelado y diseño de topologías de redes ópticas.		X	X				
RF3	Inclusión de usuarios OLT y ONU.	X		X				
RF4	Modificación de parámetros de funcionamiento de OLT.	X				X		X
RF5	Modificación de parámetros de funcionamiento de ONU.	X				X		X
RF6	Integración e implementación de algoritmos de asignación.				X	X		X
RF7	Visualización de resultados de asignación de recursos.			X			X	

3.3.3 Variables, Parámetros y Salidas

Variables Generales

- **topología_red (Estructura)**: Configuración de OLT, ONUs y conexiones (paths).
- **tiempo_simulación (Entero)**: Duración de la simulación en segundos.

Parámetros Generales

Por cada uno de los elementos se contará con parámetros asociados a modificar dentro de la simulación.

1. OLT:

- **ancho_banda_total** (*float*): Capacidad en Mbps (ej: 1000Mbps).
- **num_ONUs_conectadas** (*entero*): Número máximo de ONUs soportadas..

2. ONU:

- **lista_Tcons** (*Array[Dict]*): Paquetes con prioridad y peso (MB solicitados).
- **scenario_config** (*String*): Configuraciones de escenarios de tráfico, estos son 4 escenarios con diferentes proporciones de Tcont's.

3. RED:

- **distancias_OLT_ONU** (*Array[Float]*): Distancias en m-km entre OLT y cada ONU.

Salidas del sistema

El simulador PonLab genera un conjunto completo de salidas que permiten analizar exhaustivamente el comportamiento de la red PON bajo diferentes configuraciones y escenarios de tráfico. Estas salidas se organizan en cuatro categorías principales que abarcan desde la definición de la topología hasta los reportes de resultados.

1. Topología:

La configuración de la red PON se almacena en archivos que definen la estructura física y el estado operativo de los dispositivos, tal como se detalla en la Tabla 10. Esta información es fundamental para replicar escenarios de simulación y verificar la disposición de los elementos de red.

Tabla 10: Salidas del Simulador PonLab - Topología

Salida	Tipo	Descripción
Archivo.pon	Archivo	Configuración de la topología PON (OLT, ONUs, conexiones)
network_state.json	JSON	Estado actual de dispositivos y conexiones de red

2. Gráficos de Análisis:

El simulador genera automáticamente una serie de visualizaciones que facilitan la interpretación de los resultados. La Tabla 11 presenta los gráficos generados, los cuales cubren aspectos críticos del rendimiento de la red como la evolución temporal de retardos, throughput y utilización de buffers, permitiendo identificar patrones de comportamiento y posibles cuellos de botella en la asignación de recursos.

Tabla 11: Salidas del Simulador PonLab - Gráficos de Análisis

Salida	Tipo	Descripción
delay_evolution.png	Imagen	Evolución temporal del delay por ONU
throughput_evolution.png	Imagen	Evolución del throughput de la red
mean_delay.png	Imagen	Delay promedio vs tiempo
p95_delay.png	Imagen	Percentil 95 de delay vs tiempo
jitter_ipdv.png	Imagen	Evolución del jitter IPDV promedio
buffer_levels.png	Imagen	Niveles de buffer por ONU
utilization.png	Imagen	Utilización de recursos de red
onu_tcont_analysis.png	Imagen	Distribución de tipos TCONT por ONU
onu_temporal_packets.png	Imagen	Análisis temporal de paquetes por ONU

3. Datos de Simulación:

La información cuantitativa de cada ejecución se preserva en formatos estructurados que permiten su posterior procesamiento y análisis. Como se observa en la Tabla 12, el archivo de metadatos documenta las condiciones bajo las cuales se realizó la simulación, mientras que los datos comprimidos contienen el registro completo de eventos y métricas por paso, posibilitando estudios detallados del comportamiento del algoritmo DBA.

Tabla 12: Salidas del Simulador PonLab - Datos de Simulación

Salida	Tipo	Descripción
metadata.json	JSON	Parámetros de configuración y contexto de la simulación (timestamp, algoritmo, escenario, número de ONUs, pasos)
datos_simulacion.json.gz	JSON comprimido	Registro completo de la simulación paso a paso con métricas de delay, throughput y eventos por ONU y TCONT

4. Reportes:

Para facilitar la interpretación inmediata de los resultados, el sistema produce un resumen ejecutivo en texto plano que consolida las métricas más relevantes. La Tabla 13 describe este reporte, el cual incluye fecha, configuración, resultados agregados y referencias a los archivos generados durante la simulación.

Tabla 13: Salidas del Simulador PonLab - Reportes

Salida	Tipo	Descripción
RESUMEN.txt	Texto plano	Reporte ejecutivo con fecha, configuración, resultados agregados (delay promedio, throughput, utilización) y listado de archivos generados

En conjunto, estas salidas proporcionan una trazabilidad completa del proceso de simulación: desde la definición del escenario hasta los resultados agregados, pasando por el registro detallado

de cada evento. Esta estructura permite tanto la validación rápida mediante el resumen ejecutivo como el análisis profundo utilizando los datos comprimidos y las visualizaciones generadas.

3.4 Módulos

Para estructurar la lógica del sistema y garantizar la separación de responsabilidades, la arquitectura se divide en bloques funcionales independientes e interconectados. En la Tabla 8 se enumera la totalidad de estos módulos, sintetizando la función principal de cada uno dentro del flujo del sistema. Posteriormente, en las subsecuentes secciones se profundiza en el propósito, la lógica interna y las interacciones clave de cada componente para asegurar una clara cohesión en el desarrollo y ejecución de la plataforma.

3.4.1 Usuario

- **Propósito:** Proporcionar un punto de interacción directa con el sistema para definir pruebas, configurar parámetros, seleccionar topologías y evaluar resultados. El módulo representa al usuario que utiliza el “framework” para investigación y análisis de redes PON.
- **Alcance:** Desde el punto de vista de este módulo, el sistema es una plataforma interactiva que ofrece control total sobre la ejecución de simulaciones. El usuario puede cargar o definir configuraciones, iniciar simulaciones y recibir resultados a través de la interfaz. No realiza procesamiento interno, sino que actúa como origen y destino de la interacción.
- **Dependencias:**
 - Interfaz: Enviar comandos, ver resultados y controlar la ejecución.
 - Topologías: Definir o seleccionar estructuras de red.
 - Salida Gráfica/Textual: Visualizar los resultados.
- **Supuestos:**
 - Se asume que el usuario posee conocimientos básicos sobre redes PON y parámetros relevantes.

- Se espera que el sistema esté operativo (backend e interfaz) para poder interactuar correctamente.

■ **Restricciones:**

- El usuario solo puede interactuar a través de las opciones expuestas por la interfaz.
- No se permite la ejecución simultánea de múltiples simulaciones desde una misma sesión.

■ **Estructura General:**

Entrada: Comandos de configuración, parámetros de red

Ejecución: Interacción con interfaz para iniciar simulación

Salida: Resultados visualizados, métricas exportadas

3.4.2 Topologías

- **Propósito:** Gestionar la creación, edición, carga o interpretación de topologías de red para ser utilizadas en el proceso de simulación. El módulo permite definir desde cero la arquitectura PON o reutilizar configuraciones preexistentes.
- **Alcance:** El módulo recibe parámetros del usuario (como número de ONUs, niveles de splitters, etc.) y los transforma en una representación estructurada de red. Puede además interpretar topologías a partir de archivos o entradas externas, generando objetos compatibles con el motor de simulación.
- **Dependencias:**
 - Usuario: Proporciona parámetros iniciales.
 - Interfaz: Visualiza y modifica topologías.
 - Simulación: Consume la topología para ejecutar el análisis.
- **Supuestos:**
 - Las topologías son estáticas durante una ejecución.

- El formato de entrada y representación está bien definido.

■ **Restricciones:**

- Límite en la cantidad de nodos o profundidad por motivos de rendimiento.
- Validación estructural para asegurar topologías compatibles con PON.

■ **Estructura General:**

- Entrada: Parámetros de red o archivo de topología
- Ejecución: Generación o interpretación de la estructura
- Salida: Objeto topológico usable por el módulo de simulación

3.4.3 Interfaz

- **Propósito:** Permitir la interacción gráfica y textual entre el usuario y el sistema. Este módulo facilita la definición de parámetros, selección de topologías, inicio de simulaciones y visualización de resultados desde una vista centralizada.
- **Alcance:** Desde la perspectiva de este módulo, el sistema se presenta como un conjunto de funcionalidades integradas accesibles a través de una GUI (Interfaz Gráfica de Usuario) desarrollada en Python/C++. La interfaz actúa como puente entre el usuario y el resto de módulos, gestionando la entrada de datos, control de ejecución y presentación de resultados.
- **Dependencias:**
 - Usuario: Fuente de interacción y comandos.
 - Topologías: Permite seleccionar, visualizar y editar estructuras de red.
 - Simulación: Inicia y controla la ejecución de escenarios de red.
 - Salida Gráfica/Textual: Visualiza los resultados obtenidos.
- **Supuestos:**
 - El entorno cuenta con bibliotecas de GUI compatibles (Tkinter, PyQt, etc.).

- La conexión entre módulos se realiza de forma sincronizada para mantener la coherencia del sistema.

■ **Restricciones:**

- Interfaz debe responder de manera rápida para mantener fluidez.
- Limitaciones gráficas basadas en la resolución del sistema usuario.

■ **Estructura General:**

- Entrada: Selección de topología, parámetros de simulación, comandos
- Ejecución: Lógica de control y actualización de vista
- Salida: Instrucciones a simulación y visualización de métricas

3.4.4 Simulación

- **Propósito:** Ejecutar el comportamiento completo del sistema bajo los parámetros definidos. El módulo se encarga de simular las condiciones de tráfico, aplicar decisiones SDN y calcular métricas de rendimiento en base a las topologías y algoritmos establecidos.
- **Alcance:** Funciona como núcleo del “framework”, recibiendo configuraciones desde la interfaz, interpretando las topologías y ejecutando los eventos de red. Se integra con el módulo de algoritmos y SDN para aplicar control dinámico y alimentar con resultados al módulo de salida.
- **Detalles del Motor de Simulación:** El motor de simulación de PonLab se basa en una arquitectura de eventos discretos híbrida, diseñada para equilibrar el realismo físico con la eficiencia computacional requerida para el entrenamiento de agentes de RL.
 - **Modelo de Eventos:** El simulador utiliza un motor de eventos discretos para gestionar eventos no periódicos, como la generación de paquetes (eventos de llegada), las finalizaciones de transmisión, el envío de mensajes de control y la expiración de temporizadores. Cada evento se almacena en una cola de prioridad ordenada por su marca de tiempo (timestamp) y se procesa secuencialmente.

- **Avance del Reloj (Event-less Polling):** A diferencia de los simuladores de eventos discretos tradicionales, que programan un evento para cada ciclo de sondeo (polling) de 125 μ s, PonLab implementa una optimización clave. El avance del reloj se realiza mediante una verificación temporal automática (event-less polling). Antes de procesar cualquier otro evento, el motor verifica si ha transcurrido el tiempo suficiente (múltiplos de 125 μ s) desde el último ciclo de sondeo. Si es así, se ejecutan todos los ciclos de sondeo que debieron haber ocurrido en ese intervalo de manera sincrónica y sin generar eventos explícitos en la cola principal. Esta innovación reduce drásticamente el overhead computacional, eliminando la gestión de miles de eventos de polling por segundo, lo que acelera significativamente las simulaciones y es especialmente beneficioso para el entrenamiento de agentes de RL.
- **Generación de Paquetes:** El tráfico de red se modela mediante procesos estocásticos configurables. Cada ONU genera paquetes de datos de acuerdo con un perfil de tráfico definido por el usuario (ej. residencial, empresarial). Estos perfiles dictan las tasas de llegada de paquetes y la distribución de estos en los diferentes T-CONTs (Traffic Containers). Se implementan modelos de generación de tráfico como Poisson para simulaciones generales o procesos de ráfaga (bursty) para escenarios de alta demanda. La generación de cada paquete se programa como un evento de llegada en la cola de eventos, utilizando para ello números pseudoaleatorios.
- **Funcionamiento de las Colas (T-CONT):** Cada ONU en el simulador posee múltiples colas lógicas (T-CONT 0-4 según estándar ITU-T G.984), cada una con una prioridad y un perfil de servicio diferente (por ejemplo, Best Effort, Assured, Real-Time, Non-Real-Time). Los paquetes generados se encolan en su T-CONT correspondiente. El estado de estas colas (ocupación en bytes) se monitorea constantemente y es la base de los reportes de demanda que las ONUs envían a la OLT. El simulador modela la capacidad finita de los buffers, descartando paquetes si una cola se desborda y registrando esta pérdida como una métrica.
- **Ciclo de Reporte y Concesión (DBA):** El corazón de la gestión del canal ascendente es el ciclo de sondeo DBA. Este ciclo, ejecutado en cada intervalo de 125 μ s, consta de las

siguientes fases:

1. Reporte de Demanda (Report): La OLT solicita el estado de las colas a las ONUs. El simulador, en lugar de enviar un mensaje físico, obtiene de forma eficiente los valores de ocupación de los buffers de cada ONU, conformando así el vector de solicitudes.
2. La OLT, utilizando el algoritmo DBA seleccionado (GIANT, IPACT, o el algoritmo Smart-RL basado en el modelo de RL cargado), procesa las solicitudes y calcula la asignación de ancho de banda para el siguiente ciclo. Este cálculo define el tamaño de la ventana de transmisión (concesión o grant) para cada T-CONT de cada ONU.
3. La OLT comunica a las ONUs las concesiones asignadas y los tiempos de inicio de sus transmisiones. En el simulador, esto se traduce en la programación de eventos de transmisión para los paquetes que han sido concedidos".
4. Las ONUs transmiten los paquetes de sus buffers según las ventanas de tiempo concedidas. El simulador modela el tiempo de transmisión de cada paquete en función de su tamaño y la velocidad de línea, programando un evento de finalización de transmisión para el momento en que se complete su envío.

■ Dependencias:

- Interfaz: Instrucciones de inicio, parámetros de simulación.
- Topologías: Estructura de red base.
- Algoritmos y SDN: Lógica de asignación de recursos.
- Salida Gráfica/Textual: Canal de exportación de resultados.
- Algoritmos y SDN: Lógica de asignación de recursos (heurística o controlador SDN).
- Módulo de Aprendizaje por Refuerzo: Proporciona el modelo de agente entrenado (Smart-RL) y la función de recompensa que guía las decisiones de asignación durante la fase de inferencia o entrenamiento.
- Salida Gráfica/Textual: Canal de exportación de resultados y métricas.

■ Supuestos:

- Se dispone de un motor de simulación compatible con los eventos definidos.
- El sistema puede ejecutar simulaciones sin interrupciones de recursos.
- Las configuraciones de red (topología, tráfico) son estáticas durante una ejecución de simulación.

■ **Restricciones:**

- El tiempo de simulación debe mantenerse menor al tiempo real cuando se ejecuten múltiples escenarios consecutivos.
- El consumo de memoria debe ser bajo para manejar simulaciones con un gran número de ONUs y largas duraciones.
- La precisión de los resultados depende de la correcta parametrización de los procesos estocásticos de generación de tráfico.

■ **Estructura General:**

- Entrada: Parámetros de simulación, topología, controlador y algoritmo DBA.
- Ejecución: Bucle principal que procesa eventos de la cola de prioridad y ejecuta el ciclo de sondeo mediante verificación temporal automática (event-less polling), gestionando la generación de tráfico, la ocupación de buffers, el ciclo de reporte y concesión, y aplicando las políticas de asignación de recursos.
- Salida: Métricas de desempeño agregadas (throughput, latencia promedio, jitter) y un registro detallado paso a paso del estado de la red (ocupación de buffers, asignaciones, eventos por ONU).

3.4.5 Algoritmos y SDN

- **Propósito:** Ejecutar algoritmos de asignación dinámica de ancho de banda (DBA) y operar como controlador SDN para gobernar la lógica de decisión sobre recursos de red durante la simulación. El módulo está orientado a comparar el algoritmo desarrollado desde cero con otros enfoques y validar su eficiencia.

- **Alcance:** Este módulo forma parte del núcleo computacional. Una vez iniciada la simulación, interpreta el estado de la red y toma decisiones sobre la asignación de recursos. También expone una API de control compatible con SDN (ej. OpenFlow).
- **Dependencias:**
 - Simulación: Ejecuta el motor que hace uso del controlador y algoritmos.
 - Topologías: Define la estructura sobre la cual se deben aplicar decisiones.
- **Supuestos:**
 - El entorno de simulación permite inyectar lógica SDN en tiempo de ejecución.
 - Las decisiones tomadas se reflejan en eventos de tráfico simulados.
- **Restricciones:**
 - El desempeño depende de la eficiencia del algoritmo desarrollado.
 - Latencias de decisión deben buscar fidelidad con el escenario de simulación.
- **Estructura General:**
 - Entrada: Estado actual de la red desde.
 - Ejecución: Aplicación de lógica de asignación DBA y control SDN.
 - Salida: Decisiones aplicadas sobre los flujos o anchos de banda.

3.4.6 Salida Gráfica/Textual

- **Propósito:** Presentar los resultados obtenidos del módulo de simulación en forma visual (gráficos, diagramas) y textual (informes, métricas), facilitando su análisis e interpretación por parte del usuario.
- **Alcance:** Este módulo recibe datos estructurados de simulación y los transforma en salidas comprensibles. Funciona como terminal de consulta y evaluación de experimentos de simulación. No ejecuta decisiones ni cálculos críticos, sino que procesa visualmente los resultados.

■ **Dependencias:**

- Simulación: Proporciona las métricas e indicadores.
- Interfaz: Muestra los resultados en pantalla o permite exportarlos.

■ **Supuestos:**

- Las métricas recibidas están estructuradas según el formato predefinido.
- Las bibliotecas gráficas y de exportación están correctamente instaladas.

■ **Restricciones:**

- Tiempo de generación de resultados típicos.
- Soporte para exportación en formatos estándar (CSV, PNG, PDF).

■ **Estructura General:**

- Entrada: Métricas de simulación.
- Ejecución: Procesamiento y transformación visual/textual.
- Salida: Resultados mostrados o exportados al usuario.

3.4.7 Módulo de aprendizaje por refuerzo

- **Propósito:** Gestionar el proceso de entrenamiento del agente de aprendizaje reforzado para la optimización de asignación dinámica de ancho de banda en redes PON. El módulo implementa los algoritmos de RL (Q-learning, SARSA, PPO) y coordina la interacción entre el agente y el entorno de simulación para generar políticas de control inteligentes que se adapten a diferentes patrones de tráfico y requisitos de latencia.
- **Alcance:** Este módulo opera como núcleo de inteligencia artificial del “framework”, ejecutando los ciclos de entrenamiento del agente RL a través de episodios de simulación. Recibe el estado actual de la red (ocupación de buffers, demandas de tráfico, métricas de latencia) como entrada del módulo de Simulación, toma decisiones de asignación de ancho de banda como acciones, y recibe recompensas basadas en el desempeño de latencia obtenido. El

módulo mantiene y actualiza las políticas aprendidas (Q-tables o redes neuronales) y proporciona decisiones optimizadas al módulo de Algoritmos y SDN durante la operación.

■ Dependencias:

- Simulación: Proporciona el entorno de entrenamiento, estados de la red y recompensas.
- Algoritmos y SDN: Recibe las decisiones del agente RL y las aplica como políticas DBA.
- Interfaz: Permite configurar parámetros de entrenamiento y visualizar el progreso del aprendizaje.
- Salida Gráfica/Textual: Presenta métricas de convergencia, evolución de recompensas y políticas aprendidas.

■ Supuestos:

- El entorno de simulación proporciona estados observables y recompensas consistentes.
- Los algoritmos de RL convergirán hacia políticas eficientes con suficientes episodios de entrenamiento.
- El framework permite múltiples ejecutores de entrenamiento para comparar diferentes algoritmos.
- Los patrones de tráfico durante el entrenamiento son representativos de los escenarios reales de operación.

■ Restricciones:

- El tiempo de entrenamiento debe mantenerse razonable para permitir experimentación iterativa.
- El consumo de memoria durante el entrenamiento no debe exceder los recursos disponibles del sistema.
- La convergencia del algoritmo debe ser monitoreada para evitar entrenamiento excesivo (overfitting).
- Las políticas aprendidas deben ser exportables e importables para reutilización.

■ **Estructura General:**

- Entrada: Estado de la red (S_t), configuración de algoritmo RL, parámetros de entrenamiento.
- Ejecución: Ciclo de entrenamiento (estado $\rightarrow$ acción $\rightarrow$ recompensa $\rightarrow$ actualización de política).
- Salida: Políticas entrenadas, métricas de convergencia, decisiones de asignación optimizadas.

Especificación del Agente de Aprendizaje por Refuerzo

El agente de RL se define mediante un espacio de observación de dimensión $(N \times 3) + 1$, compuesto por las solicitudes, retardos y buffers normalizados de cada ONU, junto con la utilización total del canal. El espacio de acción es un vector continuo de N pesos que se normalizan para asignar el ancho de banda de forma proporcional a cada ONU.

La función de recompensa es una combinación ponderada multiobjetivo que busca equilibrar cinco métricas de rendimiento: utilización del canal (25 %), satisfacción de demanda (30 %), equidad (20 %), penalización por retardo (15 %) y penalización por ocupación de buffer (10 %). Estos pesos son configurables desde la interfaz gráfica, permitiendo al usuario priorizar distintos objetivos según el escenario.

Por defecto, el entrenamiento se realiza durante 100 episodios, cada uno con una duración de 10 segundos de tiempo simulado y un paso de entorno de 1 ms, lo que totaliza 10 000 pasos por episodio. Todos estos parámetros (número de episodios, duración y resolución temporal) son ajustables mediante los controles del panel de RL.

El agente se implementa con el algoritmo PPO de Stable-Baselines3, empleando por defecto una red neuronal con dos capas ocultas de 64 neuronas cada una, activación $\tanh$, *learning rate* de 10^{-5} , *batch size* de 64 y factor de descuento $\gamma = 0,99$. La interfaz permite modificar la arquitectura de la red, los hiperparámetros del algoritmo y la semilla aleatoria (configurable para garantizar reproducibilidad).

Finalmente, el modelo se selecciona según la mayor recompensa promedio obtenida en un episodio de validación con semilla diferente, evitando así el sobreajuste y garantizando la genera-

lización de la política aprendida. El usuario puede elegir entre múltiples criterios de selección o conservar todos los *checkpoints* generados durante el entrenamiento.

3.5 Diseño de Interfaces

3.5.1 Modelo de Navegación

Este diagrama de navegación representa los caminos de que los usuarios podrán seguir al interactuar con la plataforma, mostrando su flujo de movimiento entre las diferentes secciones.

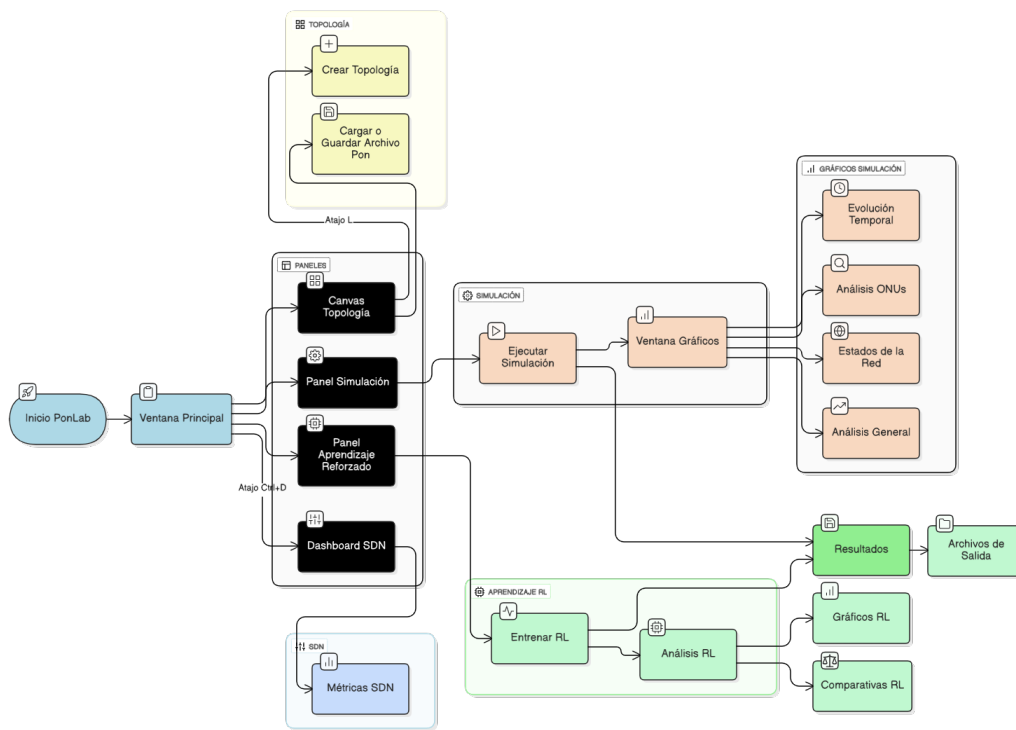


Figura 3.3: Diagrama de navegación.

El flujo de la aplicación comienza en un punto central (“Inicio PonLab”) que dirige al usuario hacia la **Ventana Principal**. Desde ahí, se accede a cuatro módulos principales:

- **Canvas de Topología:** Espacio visual para diseñar redes, donde el usuario puede:
 - Crear nuevas topologías.
 - Cargar y guardar proyectos en formato **.pon**.

- **Panel de Simulación:** Permite ejecutar simulaciones de red. Al iniciarlas, se habilitan salidas como:
 - Ventana de Gráficos (evolución temporal, estados de la red, análisis de rendimiento y análisis de ONUs).
 - Resultados exportables para análisis posterior.
- **Panel de RL (Reinforcement Learning):** Orientado al entrenamiento de modelos de aprendizaje por refuerzo, generando:
 - Gráficos de desempeño del modelo.
 - Comparativas de estrategias.
 - Resultados integrados con las salidas generales.
- **Dashboard SDN:** Consolida métricas específicas de la red definida por software (SDN), accesibles también mediante atajos desde la Ventana Principal.

Finalmente, todos los resultados y gráficos se almacenan en una sección de **Archivos de Salida**, lo que permite mantener un registro de la evolución de las simulaciones y los entrenamientos. Este diseño facilita la interacción fluida entre el diseño de topologías, la ejecución de simulaciones, el análisis mediante RL y la supervisión de métricas en SDN.

3.5.2 Vistas aplicación

Para el modelado y creación de la interfaz se implementa la librería de PyQt, con ello a continuación se describen las diferentes vistas del simulador.

Pantalla de Carga

Al ejecutar el programa se inicializa una pantalla de carga mostrando el logo e información de la aplicación.



Figura 3.4: Pantalla de PonLab Simulator

Interfaz

Una vez se hace carga de la aplicación se despliega una ventana con la siguiente distribución:

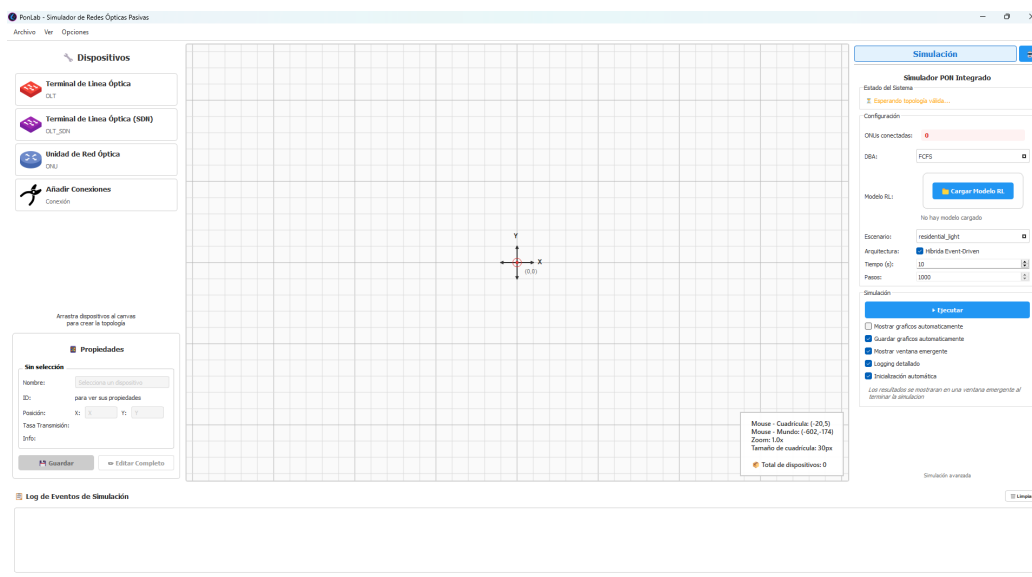


Figura 3.5: Interfaz Principal Ponlab simulador

Esta compuesto de un **Navbar** en la parte superior, con botones respectivos como:

- **Archivo:** Objetivo de abrir, cerrar, guardar, exportar archivos de topologías.
- **Ver:** En este menú se maneja el estado de los paneles del simulador, permitiendo ocultar y mostrar estos al alcance de un click.

- **Opciones:** contiene las configuraciones de tema y más información del simulador.

Adicionalmente presenta 3 paneles, el primero correspondiente a las funcionalidades de diseño de topologías con el nombre de “**Dispositivos**”. El otro panel encargado a las funcionalidades de las simulaciones. Y finalmente en la parte inferior de la aplicación una consola que despliega alertas y logs del sistema.

Panel de Dispositivos

Encargado de mostrar los dispositivos con funcionalidades de drag & drop, a excepción del botón para activar y desactivar las conexiones entre dispositivos mostrado de color verde. Adicionalmente se presenta un mini-panel de propiedades para editar los ajustes de los dispositivos.

Diseñado de topologías

A la hora de arrastrar dispositivos estos tienen la capacidad de conectarse habilitando la función **Añadir Conexiones (shortcut = L)**.

Panel de Propiedades dispositivos

A la hora de hacer doble click en un dispositivo se despliega un panel de propiedades con información del dispositivo. En este existen casillas donde se puede modificar propiedades.

Simulación

Encargado de generar diferentes simulaciones, con parámetros configurables como el algoritmo de asignación, el tiempo de simulación y además de desplegar una vista propia cuando se carga un modelo de aprendizaje reforzado.

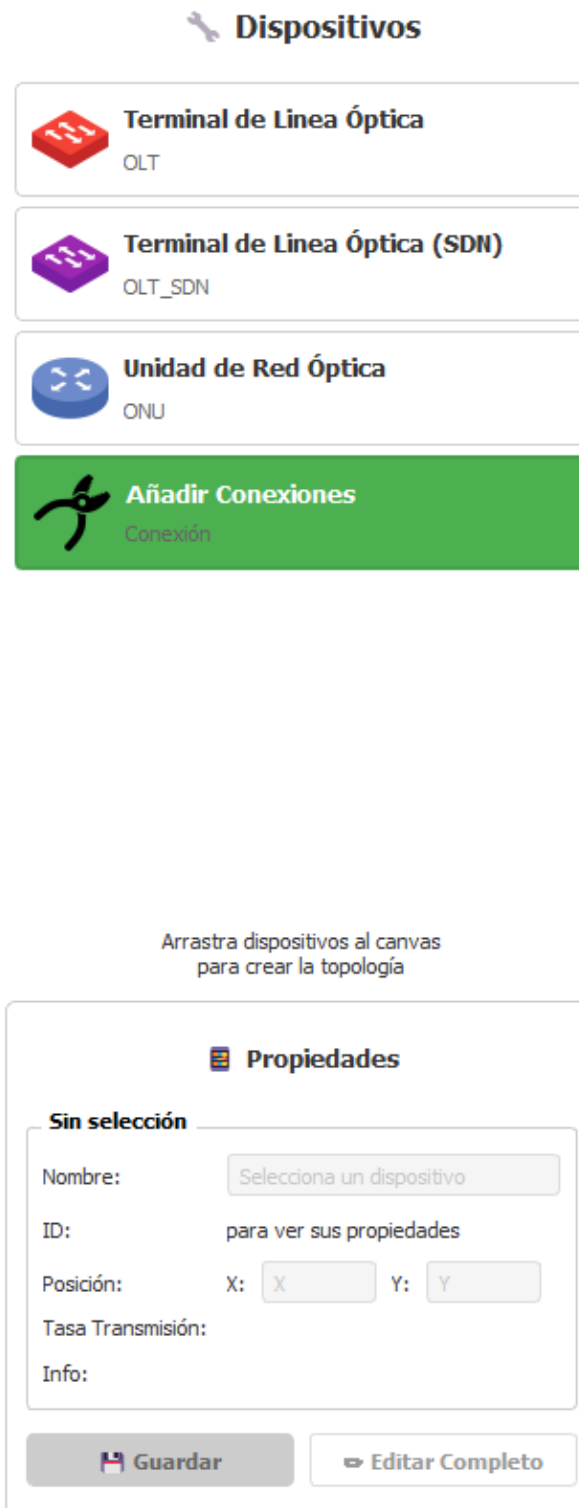


Figura 3.6: Panel de Dispositivos

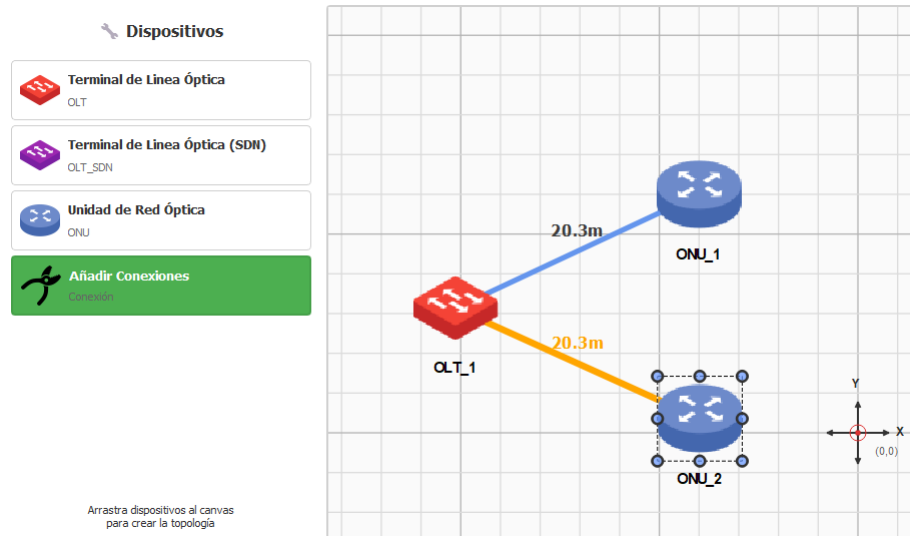
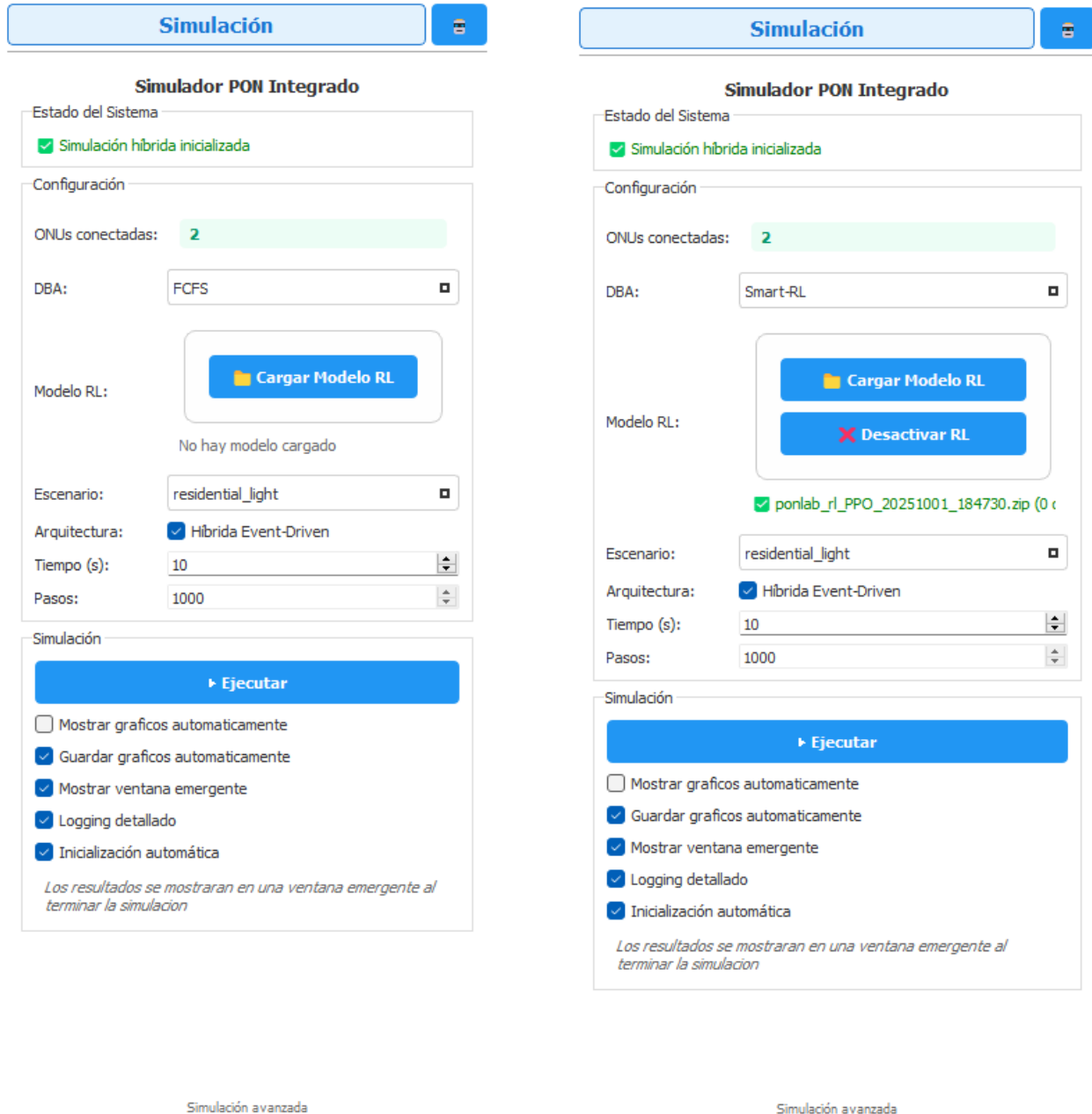


Figura 3.7: Generación de topología con 1 OLT conectada a 2 ONU's

Figura 3.8: Panel de Propiedades de un terminal de linea óptico

Panel Aprendizaje Reforzado

Apretando el icono a la derecha del título de **Simulación**, se actualiza el panel al de aprendizaje reforzado. Este encargado de entrenar modelos bajo los parámetros que convenga el usuario. Con los modelos entrenados se pueden cargar en el panel de simulación para desplegar los algoritmos con aprendizaje.



(a) Panel de Simulación.

(b) Panel de Simulación Aprendizaje Reforzado

Figura 3.9: Paneles de apartado de simulación.

Soporte Modo Oscuro

Además la aplicación cuenta con un tema oscuro para los momentos de descanso visual. Este se activa desde el menú de opciones, los resultados se visualizan como la figura 3.11.

Aprendizaje Reforzado

💡 La simulación con RL ahora está integrada en la pestaña principal de simulación. Use 'Smart-RL' como algoritmo DBA después de cargar un modelo entrenado.

Entrenamiento

Entorno PON

ONUs: 3

Tráfico: residential_medium

Duración (s): 1,0

Timestep (ms): 0,5

Algoritmo de Aprendizaje

Algoritmo: PPO

Learning Rate: 0,000001

Batch Size: 64

Log de Entrenamiento

[10:09:19] ✂ Log limpiado

Limpiar Log

Entrenamiento de agentes inteligentes

Figura 3.10: Panel de Aprendizaje Reforzado

3.6 Metodología Experimental

Para evaluar la eficacia y precisión del framework propuesto, el diseño experimental se centró en un escenario de alta demanda que reflejó el despliegue de una red residencial G-PON. Esta experimentación buscó medir el rendimiento de la red mediante el contraste del motor de simulación

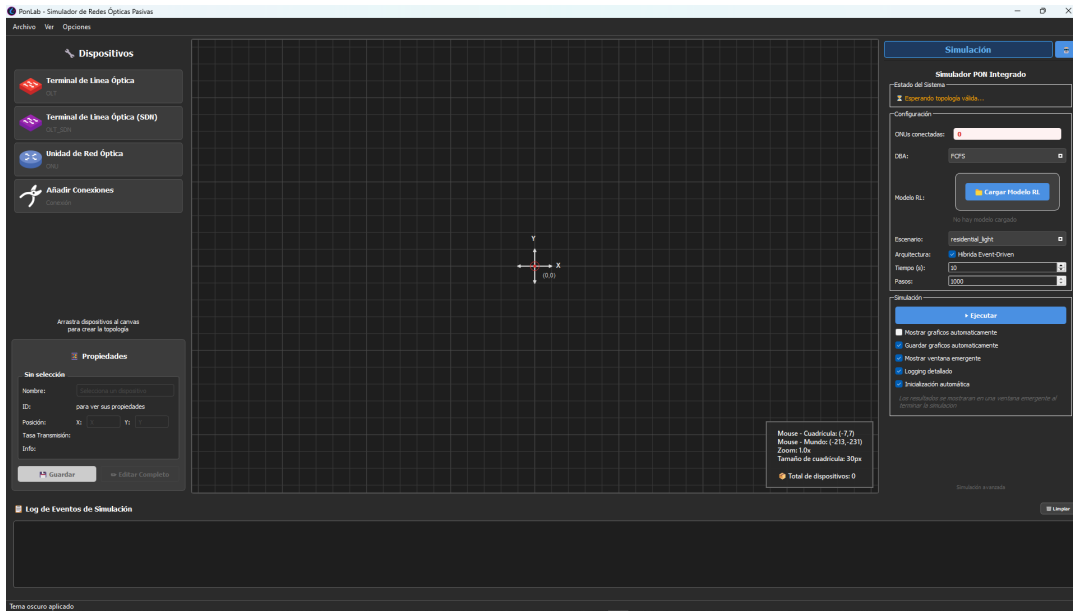


Figura 3.11: PonLab Simulator modo oscuro

interno frente a diferentes paradigmas de asignación de recursos.

3.6.1 Escenario de simulación y parámetros

La evaluación se realizó mediante el modelado de una topología compuesta por un (1) OLT conectado a ocho (8) ONUs, las cuales operaron bajo tasas de transmisión de 2.5 Gbps en enlace descendente y 1.25 Gbps en enlace ascendente. Las simulaciones se ejecutaron durante un tiempo de 10 segundos, utilizando un ciclo de sondeo fijo de $125 \mu s$, el cual estuvo sincronizado con los estándares de red PON para garantizar la fidelidad de los ciclos de reporte y concesión.

Simplificaciones del Entorno Físico: Con el objetivo de centrar el análisis en el rendimiento de los algoritmos de asignación de ancho de banda (DBA) y la orquestación SDN, el modelo físico implementado en PonLab asumió condiciones de transmisión ideales. En este sentido, se despreciaron fenómenos como la atenuación por distancia, las pérdidas por inserción debido al número de cortes o empalmes en la fibra, y otros efectos no lineales del medio óptico. Esta decisión metodológica permitió aislar el impacto de las políticas de control sobre métricas de red como el throughput, la latencia y el jitter, sin interferencias por degradación de señal de la capa física.

Como se observó en la Figura 3.12, el escenario implementado utilizó una arquitectura de estrella donde el tráfico ascendente fue recolectado por la OLT. Este diagrama no solo representó

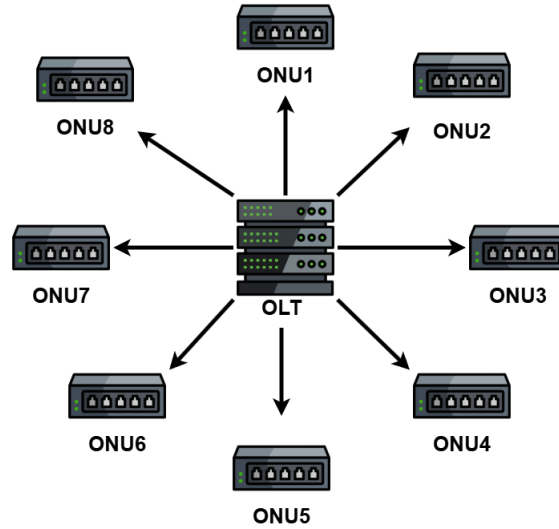


Figura 3.12: Esquema de la topología y escenario de simulación en PonLab.

la conectividad física del entorno simulación, sino que ilustró los puntos estratégicos de control donde el controlador SDN o el agente DRL interactuaron dinámicamente con el sistema. Cada ONU modeló un tráfico residencial heterogéneo, lo que permitió poner a prueba la capacidad de respuesta de los algoritmos ante ráfagas de datos concurrentes.

A modo de síntesis, la Tabla 14 resume las configuraciones operativas y los parámetros fundamentales establecidos para el entorno de simulación.

Tabla 14: Parámetros de simulación

Parámetro	Valor
Arquitectura de Red	G-PON
Tasa de Línea (Descendente / Ascendente)	2.5 Gbps / 1.25 Gbps
Número de ONUs	8
Perfil de Tráfico	Residencial
Algoritmo DBA (Línea Base)	GIANT
Ciclo de Sondeo (<i>Polling Cycle</i>)	125 μ s
Duración de la Simulación	10 s

3.6.2 Modelos de Comparación

Para validar la capacidad analítica del framework, se estableció una comparación de rendimiento (*throughput*, *latencia* y *jitter*) entre tres módulos de control distintos:

1. **DBA Heurístico (Línea Base):** Algoritmos tradicionales de asignación. Para este escenario de simulación ocuparemos **GIANT**.
2. **Controlador SDN:** Orquestación centralizada basada en reglas, encargada de separar el plano de control del plano de datos.
3. **Agente DRL (Deep Reinforcement Learning):** Toma de decisiones autónoma basada en recompensas orientadas a la optimización de retardos. Cabe destacar que este módulo fue integrado en un estado de desarrollo activo con fines de investigación abierta.

3.6.3 Métricas de Evaluación

La validación de los resultados se realizó a través de las siguientes métricas de Calidad de Servicio (QoS):

1. **Tasa de Transferencia (Throughput):** Se evaluó el volumen de datos de carga útil transmitidos con éxito en el tiempo. Esta métrica permitió comparar qué paradigma de control logró una mejor utilización de la capacidad del canal en escenarios de alta densidad.
2. **Latencia (Delay):** Se evaluó el retardo promedio que experimentaron los paquetes. Se utilizó como referencia los umbrales críticos de la literatura (latencia inferior a 10 ms) para determinar el grado de madurez y convergencia de cada modelo evaluado frente a tráfico exigente.
3. **Jitter (IPDV):** Variación del retardo entre paquetes, fundamental para garantizar la estabilidad en servicios en tiempo real. El análisis se centró en el enlace ascendente, con el objetivo de evaluar si los paradigmas lograron mantener este valor acotado entre 1 ms y 2 ms.
4. **Equidad (Fairness):** Utilización del Índice de Equidad de Jain (*Jain's Fairness Index*) para medir la distribución justa del ancho de banda entre las ONUs concurrentes. Cabe destacar

que esta métrica no representó un criterio de validación transversal para los tres métodos, sino que se utilizó como un indicador exclusivo para evaluar la eficiencia del controlador SDN. Esto se debió a que la orquestación centralizada SDN manejó explícitamente esta variable dentro de su lógica de enrutamiento para asegurar una distribución equitativa de los recursos.

4. Validación funcional y resultados experimentales

Como se detalló en la metodología (Sección 3.6), el valor principal de este entorno de pruebas es que permite contrastar directamente el desempeño de tres enfoques distintos para la asignación de recursos: un esquema DBA tradicional basado en GIANT, una arquitectura con controlador centralizado SDN y un agente inteligente entrenado mediante Aprendizaje por Refuerzo Profundo (DRL).

Haciendo uso de la interfaz gráfica del simulador, se configuraron las topologías, los perfiles de tráfico y los parámetros de red para cada simulación, llevando a cabo una única ejecución por cada una de las tres estrategias de control (DBA, SDN y RL). A continuación, se presentan las mediciones objetivas arrojadas por la herramienta durante dichas simulaciones. La presentación de resultados se organiza en dos etapas: primero, la validación funcional del software y, posteriormente, el reporte de las métricas obtenidas para los tres modelos de control mediante indicadores como el *throughput*, la *latencia* y el *jitter*.

4.1 Validación Funcional del Framework PonLab

Antes de evaluar el rendimiento de los algoritmos en el canal ascendente, se verificó la capacidad de *PonLab Simulator* para modelar, configurar y enlazar correctamente los distintos componentes de una red PON a través de su entorno gráfico.

El proceso comenzó con la parametrización individual de cada dispositivo. En la Figura 4.1 se muestra el panel de configuración del nodo central, denominado `olt_base`, donde se ingresaron las variables de tráfico del diseño experimental. De igual manera, la Figura 4.2 ilustra la configuración de una de las ocho ONUs residenciales simuladas, un paso clave para asegurar que todos los usuarios concurrentes compartieran las mismas propiedades y perfiles base durante las pruebas.

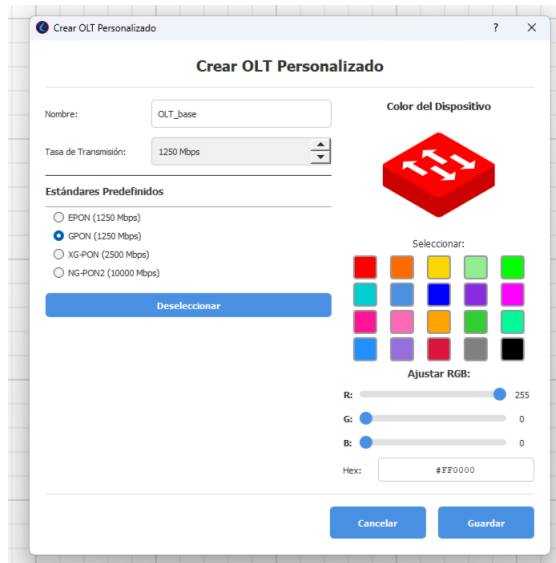


Figura 4.1: Panel de creación y configuración de parámetros para el dispositivo *OLT_base*.

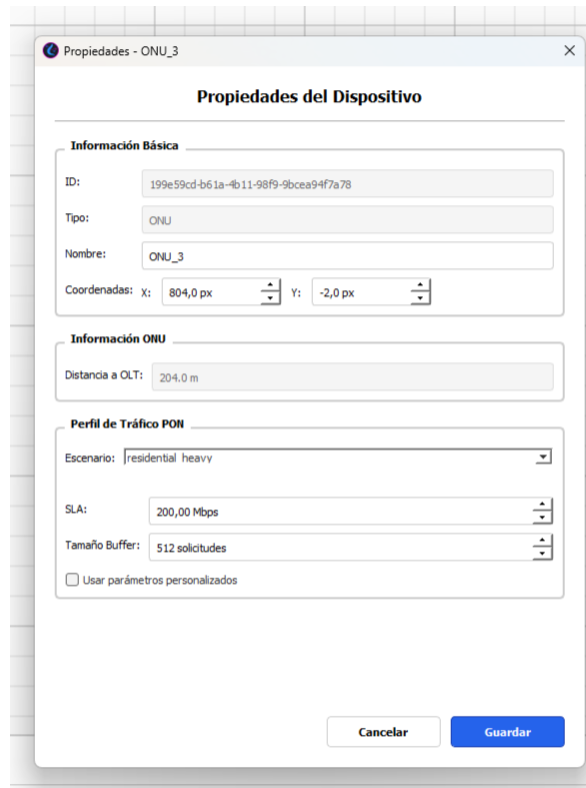


Figura 4.2: Interfaz gráfica de configuración de parámetros para una ONU línea base.

Con los nodos definidos, el siguiente paso fue la construcción de la topología punto a multi-punto. Haciendo uso de la herramienta interactiva “Añadir Conexiones (L)”, se enlazaron con éxito

las ocho ONUs hacia la OLT, consolidando la red residencial que se expone en la Figura 4.3. Este procedimiento demostró que el entorno gráfico asocia los elementos visuales y mapea la estructura en el plano de datos de forma consistente.

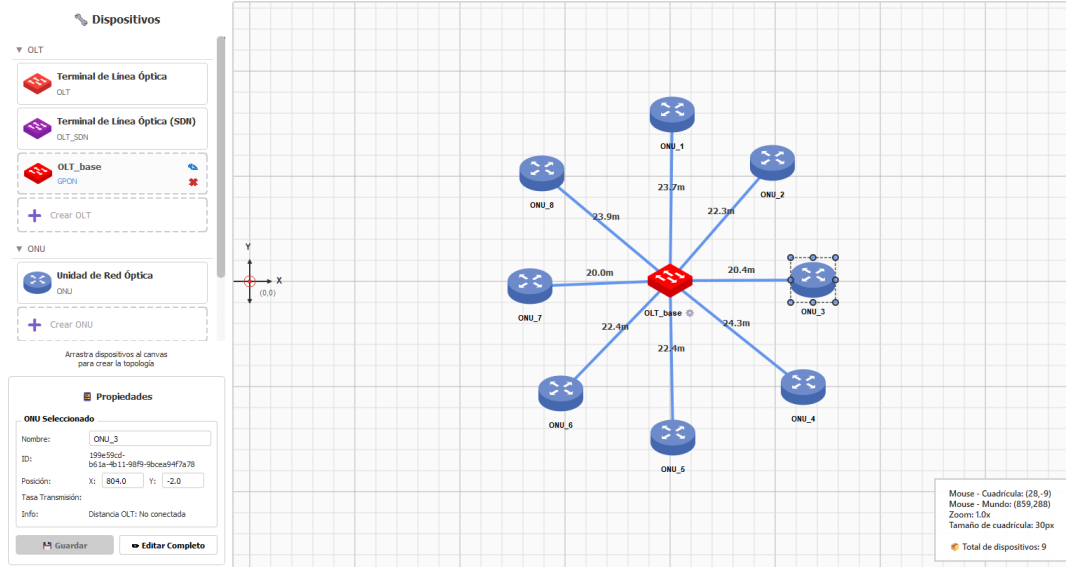


Figura 4.3: Topología residencial completa de 1 OLT y 8 ONUs construida en el entorno gráfico.

Por último, se puso a prueba la flexibilidad de la plataforma frente a cambios dinámicos de paradigma en el plano de control. Al sustituir el nodo raíz convencional por un componente del tipo OLT SDN, el simulador acopló de manera automática las dependencias lógicas y las reglas de enrutamiento centralizado, tal como se observa en la Figura 4.4.

Para consolidar la verificación del sistema frente a las especificaciones de diseño, la Tabla 15 resume el protocolo de pruebas funcionales ejecutado sobre la plataforma. En esta evaluación se contrastaron las acciones del usuario y del motor de simulación con los Requisitos Funcionales (RF) y No Funcionales (RNF) definidos en la Sección 3.1, vinculándolos directamente con el cumplimiento de los objetivos del proyecto.

Esta verificación confirma que *PonLab Simulator* actúa como un marco integrado y operativo. Como se explicita en el Anexo A, este procedimiento constituye la evidencia práctica del cumplimiento del **OE1** (al validar la representación precisa del modelo MAC en el motor) y del **OE3** (al demostrar la correcta interacción de la interfaz gráfica desarrollada en Python para la gestión de topologías y parámetros de red).

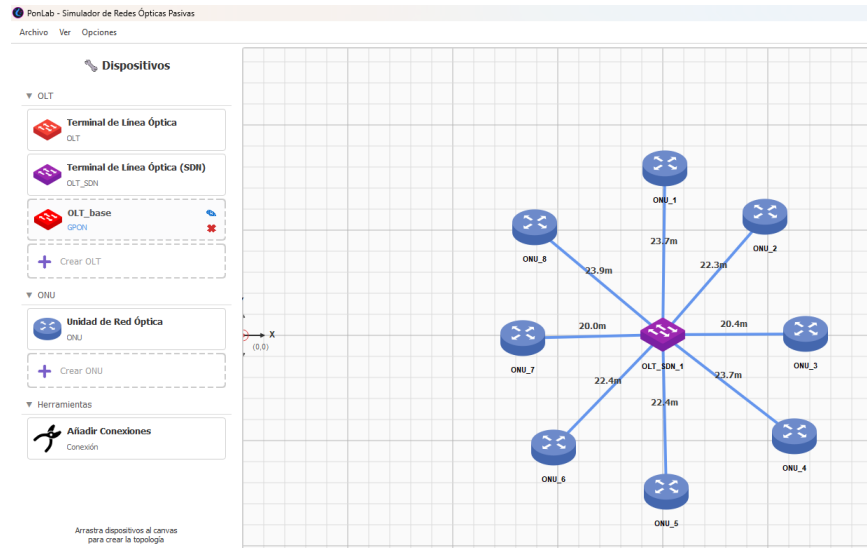


Figura 4.4: Topología de red con integración automática del paradigma de control centralizado SDN.

4.2 Monitoreo Operacional del Esquema Heurístico Tradicional

Para iniciar el reporte de rendimiento en el canal ascendente, se estableció el escenario base utilizando el algoritmo heurístico tradicional GIANT (*GigaPON Access Network*). Este caso representa la referencia convencional del estudio y se ejecuta directamente como mecanismo DBA del simulador. A partir de esta base experimental se contrastan posteriormente dos enfoques adicionales: una arquitectura con control centralizado SDN y un módulo inteligente Smart-RL, este último operando con su propio agente entrenado por Aprendizaje por Refuerzo Profundo.

Antes de mostrar las curvas de tráfico, se utilizó la interfaz de PonLab para definir las condiciones de la simulación. La Figura 4.5 muestra el panel de configuración global del simulador, en donde se seleccionó el módulo heurístico *baseline* y se ajustó el tiempo de ejecución a un intervalo continuo de 10 segundos. Cabe destacar que, con el objetivo de garantizar la fidelidad con el estándar físico de las redes G-PON, la duración del ciclo de transmisión no es un parámetro manipulable desde la interfaz gráfica; por el contrario, opera como una constante estricta de $125 \mu s$ definida nativamente en el código del motor de eventos. Este paso confirmó la capacidad del entorno visual para alternar y cargar lógicas de red de forma dinámica, manteniendo intactas las restricciones físicas del modelo subyacente.

Una vez concluida la simulación de 10 segundos, la herramienta muestra de forma automática

Tabla 15: Matriz de validación funcional de requisitos del sistema y trazabilidad de objetivos.

Requisito	Prueba Ejecutada	Resultado Esperado	Resultado Observado	Estado
RF1, RF6	Selección e inyección de algoritmos DBA (GIANT, SDN, Smart-RL).	Distribución del ancho de banda según la lógica del algoritmo.	Asignación correcta de ventanas de transmisión por T-CONT.	Exitoso
RF2, RNF1	Creación y edición visual de topologías en el canvas (<i>drag-and-drop</i>).	Renderización interactiva y enrutamiento gráfico de nodos.	Disposición dinámica de OLT, ONUs y enlaces mediante atajos (tecla L).	Exitoso
RF3, RF4, RF5	Configuración de parámetros en paneles de OLT y ONU.	Actualización de atributos de tráfico, buffers y capacidades.	Mapeo correcto de variables en el objeto de simulación.	Exitoso
RF7, RNF5	Despliegue de gráficos temporales y exportación de archivos (.json, .txt).	Generación automática de métricas y almacenamiento estructurado.	Exportación completa de métricas de QoS y reportes resumidos.	Exitoso
RNF2, RNF3	Ejecución del entorno y adición modular de nuevos agentes RL.	Operación multiplataforma y extensibilidad sin reestructuración.	Carga y ejecución del módulo Smart-RL sobre Python/PyQt5.	Exitoso

las gráficas del comportamiento de las 8 ONUs concurrentes bajo una curva de tráfico ascendente.

La Figura 4.6 presenta la evolución temporal del *throughput* promedio en el enlace ascendente (*upstream*) durante una simulación de 10 segundos utilizando el algoritmo GIANT. La gráfica corresponde a una serie temporal expresada en MB/s, donde se observa un comportamiento oscilatorio a lo largo de toda la simulación. Los valores registrados fluctúan entre un mínimo de 19.429 MB/s y un máximo de 78.324 MB/s, con un promedio general de 42.900 MB/s. Además, la curva exhibe variaciones continuas con múltiples incrementos y descensos instantáneos distribuidos durante todo el intervalo de tiempo evaluado.

Por otra parte, la Figura 4.7 muestra la evolución del *Mean Delay* medido durante la simulación. La gráfica representa el retardo promedio de transmisión en milisegundos a lo largo del tiempo, evidenciando fluctuaciones continuas entre las distintas muestras registradas. Los valores observados

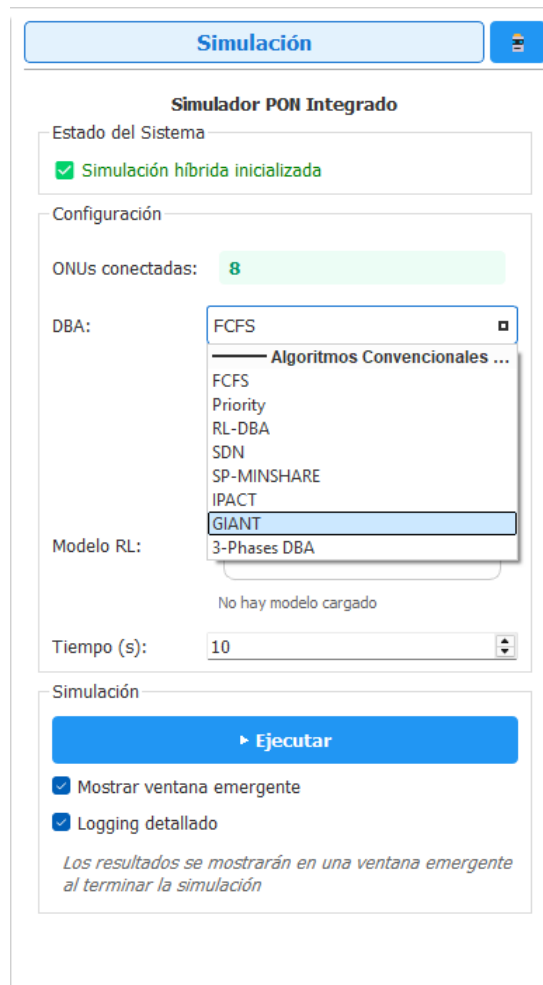


Figura 4.5: Interfaz de simulación de PonLab: Selección del algoritmo base GIANT y ajuste del tiempo de ejecución.

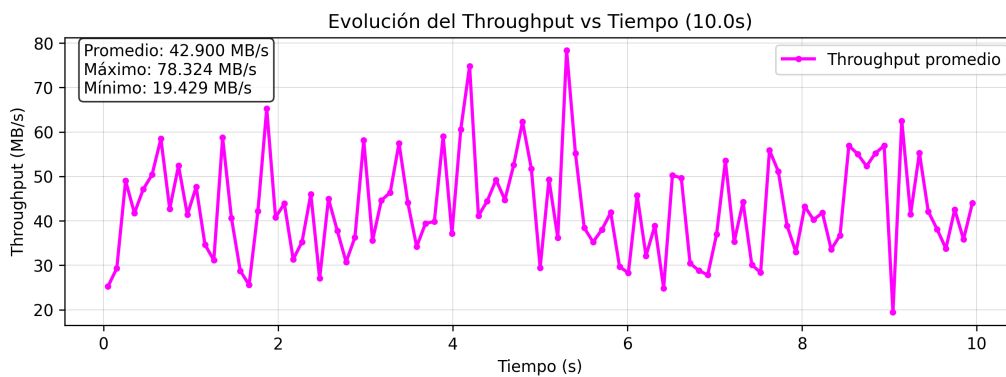


Figura 4.6: Evolución temporal del Throughput en el canal upstream usando el módulo DBA Heurístico.

varían entre un mínimo de 0.690 ms y un máximo de 3.873 ms, alcanzando un promedio global

de 1.509 ms. Asimismo, la curva presenta múltiples picos instantáneos distribuidos durante toda la simulación, destacando uno de mayor magnitud cercano al segundo 4.2.

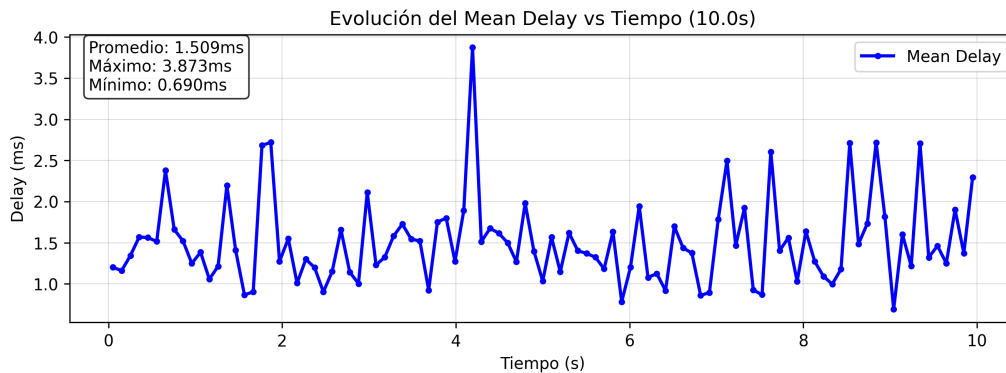


Figura 4.7: Retardo promedio registrado para las 8 ONUs bajo el control tradicional basado en GIANT.

Finalmente, la Figura 4.8 expone la evolución temporal del *Jitter IPDV* (*Instantaneous Packet Delay Variation*) en el enlace ascendente. La gráfica presenta la variación temporal del retardo expresada en milisegundos durante los 10 segundos de simulación. Los registros muestran oscilaciones continuas a lo largo de toda la prueba, con valores comprendidos entre un mínimo de 0.139 ms y un máximo de 0.928 ms, mientras que el promedio general corresponde a 0.422 ms. Además, se observan incrementos puntuales en torno a los segundos 1.7 y 4.1, donde la curva alcanza sus valores más elevados.

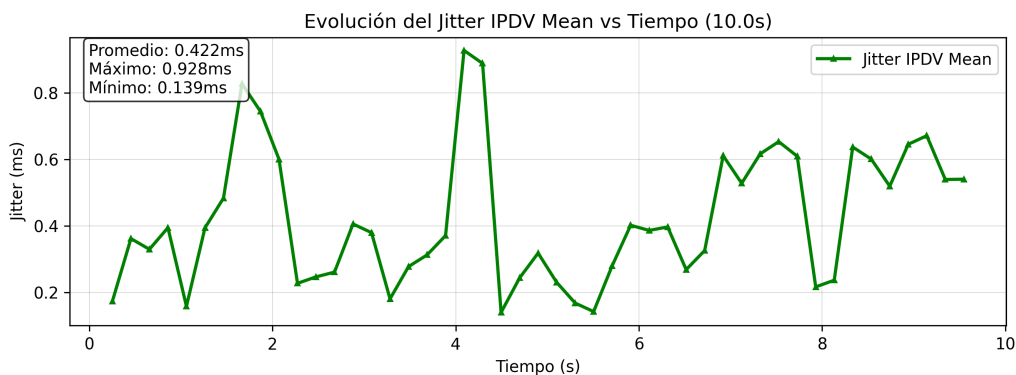


Figura 4.8: Comportamiento del Jitter (IPDV) upstream para el escenario base heurístico.

4.3 Evaluación de la Arquitectura de Control Centralizado SDN

Continuando con el análisis comparativo, el segundo escenario experimental consistió en la ejecución del plano de control centralizado. Para esta simulación se mantuvo intacta la estructura y distribución de la topología residencial de ocho ONUs concurrentes, pero sustituyendo el nodo raíz convencional por un dispositivo compatible con redes definidas por software, tal como se validó visualmente en la Figura 4.4. Bajo este esquema, la lógica de bajo nivel de GIANT continúa procesando la asignación física de slots temporales en el canal ascendente, mientras que los criterios de distribución de recursos y priorización son administrados dinámicamente por la capa del controlador central.

La Figura 4.9 presenta la evolución temporal del *throughput* promedio en el enlace ascendente durante los 10 segundos de simulación bajo una arquitectura PON controlada mediante una OLT SDN. La serie temporal, expresada en MB/s, exhibe un comportamiento dinámico con oscilaciones continuas distribuidas a lo largo de todo el intervalo de observación. Los registros obtenidos muestran un valor mínimo de 25.684 MB/s y un máximo de 68.472 MB/s, mientras que el promedio global alcanzado corresponde a 45.607 MB/s. Asimismo, la curva mantiene variaciones moderadas entre muestras consecutivas, sin evidenciar interrupciones abruptas ni periodos prolongados de degradación del caudal de transmisión.

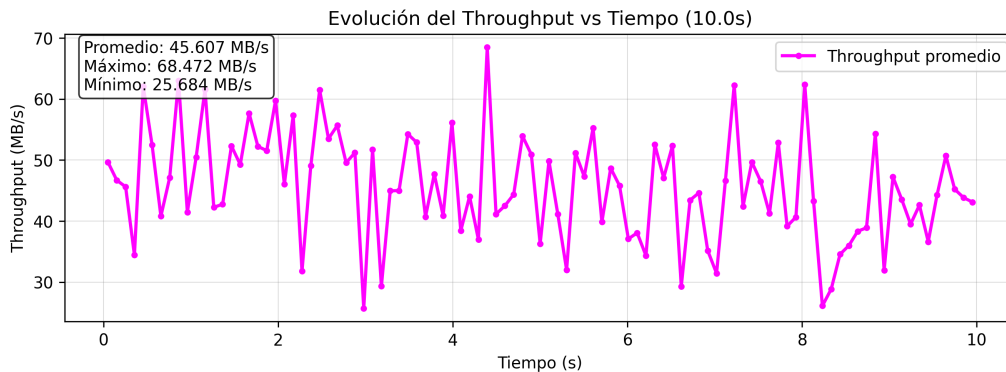


Figura 4.9: Evolución temporal del Throughput en el canal upstream bajo la orquestación del Controlador SDN.

Por su parte, la Figura 4.10 muestra la evolución temporal del retardo promedio (*Mean Delay*) registrado durante la simulación. La gráfica, expresada en milisegundos, refleja una distribución

temporal relativamente estable con fluctuaciones periódicas de baja amplitud entre las distintas muestras observadas. Los valores medidos oscilan entre un mínimo de 0.841 ms y un máximo de 3.238 ms, estableciendo un promedio general de 1.589 ms durante toda la ejecución. Adicionalmente, se observan incrementos puntuales alrededor de los segundos 2.9, 4.4 y 8.8, donde la curva alcanza los valores de mayor magnitud registrados en el experimento.

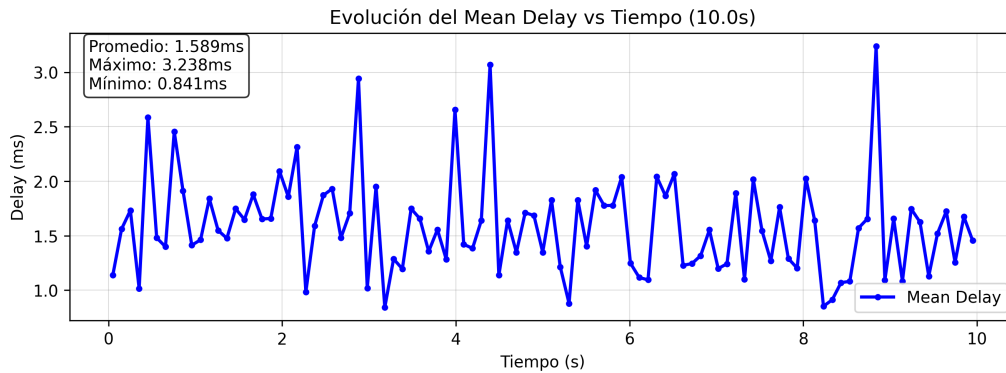


Figura 4.10: Retardo promedio registrado para las 8 ONUs bajo la gestión del plano de control SDN.

Asimismo, la Figura 4.11 expone el comportamiento temporal del *Jitter* IPDV (*Instantaneous Packet Delay Variation*) en el enlace ascendente. La gráfica representa la variación instantánea del retardo en milisegundos durante los 10 segundos de simulación, evidenciando oscilaciones continuas y distribuidas de manera irregular a lo largo de toda la prueba. Los valores registrados varían entre un mínimo de 0.089 ms y un máximo de 0.791 ms, mientras que el promedio global corresponde a 0.376 ms. De igual forma, se identifican incrementos localizados cercanos a los segundos 3.0, 4.4 y 8.8, donde la curva alcanza sus mayores niveles de variabilidad temporal.

Una característica propia del entorno PonLab al utilizar una OLT basada en SDN es la habilitación de una interfaz de telemetría avanzada para el monitoreo de métricas operacionales y estadísticas internas del plano de control. La Figura 4.12 presenta el panel de resumen global del sistema al finalizar la simulación, consolidando indicadores relacionados con desempeño, utilización de recursos y comportamiento dinámico del controlador.

Entre las métricas reportadas se observa un total de 413 reconfiguraciones automáticas realizadas por el controlador SDN durante los 10 segundos de ejecución, así como 3973 decisiones DBA procesadas por el plano de control. Asimismo, la interfaz registra una utilización de grants

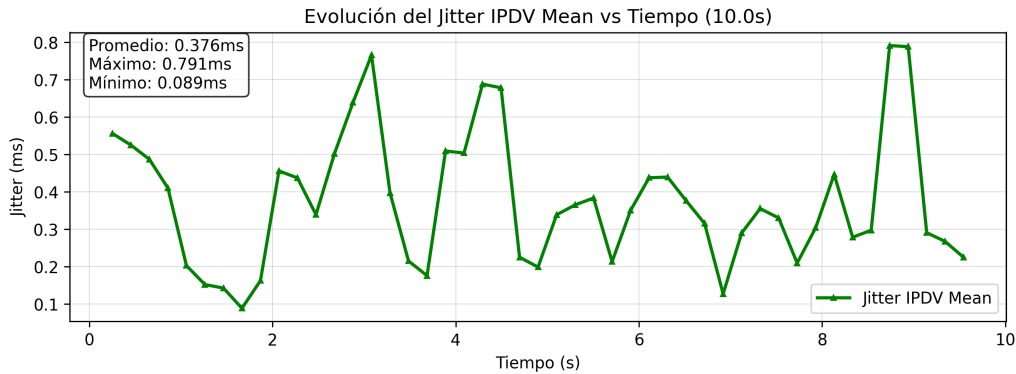


Figura 4.11: Comportamiento del Jitter (IPDV) upstream bajo el esquema de control centralizado.

del 100.2 %, una eficiencia espectral de 0.37 bits/Hz y un total de 396 violaciones QoS acumuladas durante el experimento.

De igual manera, el panel reporta un índice de fairness de Jain de 0.997, métrica utilizada para cuantificar el nivel de equidad en la distribución de recursos entre las ONUs. Dado que este índice se encuentra definido en un rango entre 0 y 1, el valor obtenido indica que la asignación de ancho de banda se mantuvo prácticamente uniforme entre los nodos participantes durante la simulación.



Figura 4.12: Métricas consolidadas en el panel de resumen global de la arquitectura SDN.

Complementariamente, la Figura 4.13 muestra el panel operativo asociado al controlador SDN, donde se presentan métricas internas vinculadas al comportamiento dinámico del plano de control durante la simulación. Entre los indicadores visibles se registra un tiempo promedio de respuesta

de 2.50 ms y una latencia de decisión de 2.00 ms para el procesamiento de solicitudes DBA.

Además, la interfaz reporta una tasa de reasignación de 3657 eventos de redistribución de recursos ejecutados automáticamente por el controlador a lo largo de los 10 segundos de simulación. Esta métrica representa la cantidad total de ajustes dinámicos realizados sobre la asignación de ancho de banda entre las ONUs durante la operación del sistema. Finalmente, el panel registra una utilización promedio de ancho de banda del 8.5 % en el entorno evaluado.



Figura 4.13: Interfaz de monitoreo y estado operacional del componente controlador SDN.

Finalmente, la Figura 4.14 detalla la distribución de ancho de banda y tráfico por clase de servicio administrada por el entorno SDN. La tabla resume el porcentaje de recursos asignados, la cantidad de paquetes procesados y la latencia promedio asociada a cada categoría de prioridad configurada en el sistema.

Los resultados muestran que la clase de servicio Crítico concentra el mayor porcentaje de ancho de banda asignado con 292.30 Mbps, equivalente al 64.1 % del total, seguida por la categoría Alto con 106.63 Mbps (23.4 %). Por otra parte, las clases Medio, Bajo y Best Effort registran asignaciones de 42.54 Mbps (9.3 %), 11.91 Mbps (2.6 %) y 2.69 Mbps (0.6 %), respectivamente. Asimismo, la tabla presenta las latencias promedio observadas para cada nivel de prioridad, cuyos valores varían entre 0.68 ms y 3.45 ms según la clase de servicio procesada.

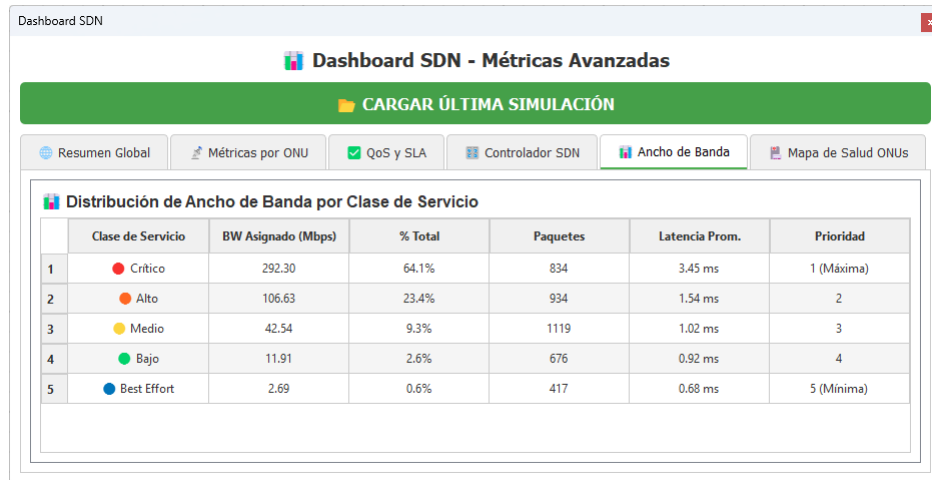


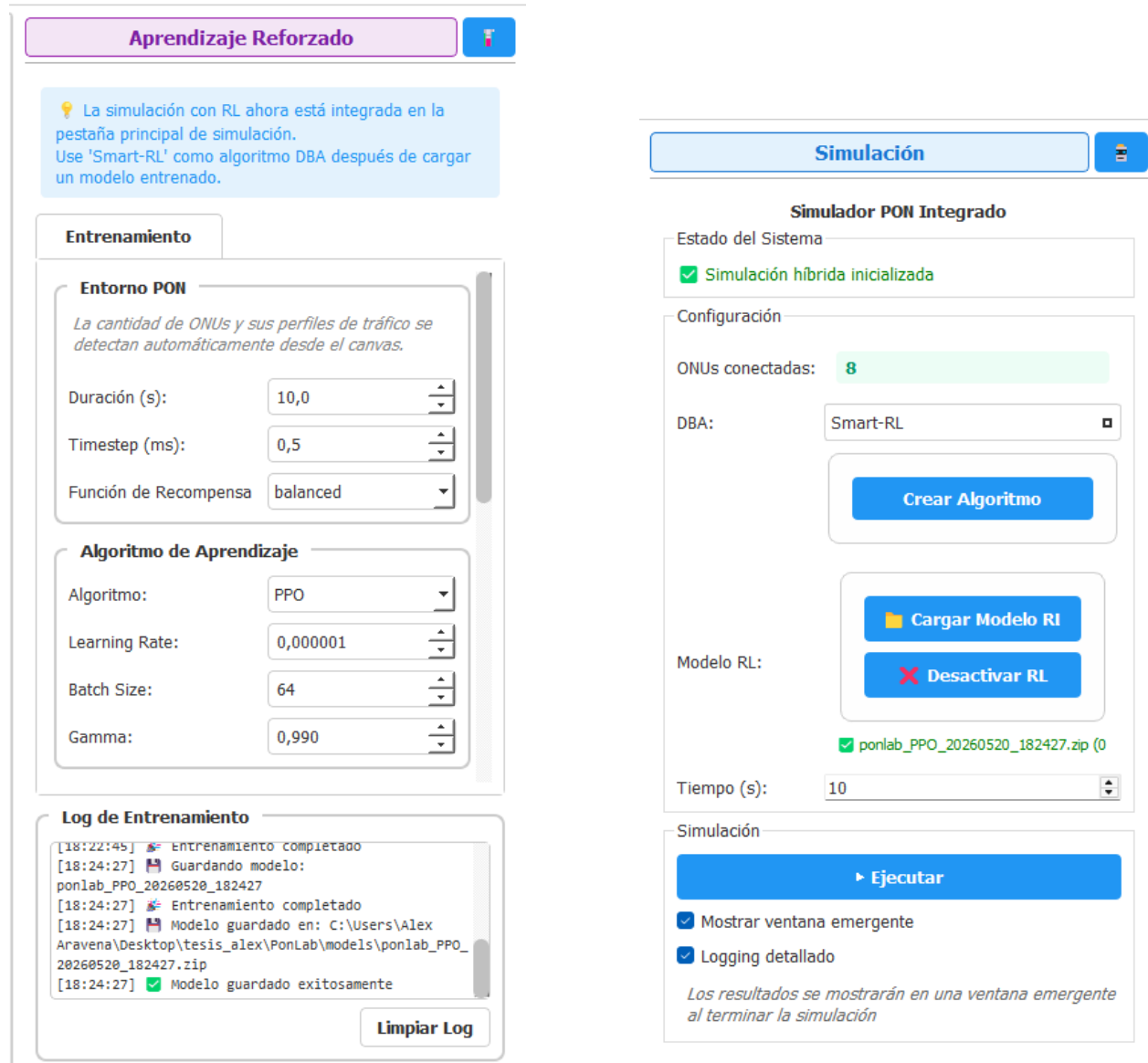
Figura 4.14: Distribución de ancho de banda y tráfico por clase de servicio en el entorno SDN.

4.4 Resultados Preliminares del Módulo de Aprendizaje por Refuerzo (DRL)

El último bloque operacional evaluado corresponde al plano de control inteligente guiado por Aprendizaje por Refuerzo Profundo. Para esta prueba, el entorno gráfico de *PonLab* permitió emplear la topología PON base con 8 ONUs conectadas, incorporando el módulo Smart-RL como mecanismo de asignación dinámica de recursos. A diferencia del escenario SDN, este caso se ejecuta desde el panel principal de simulación del entorno PON integrado, utilizando un modelo previamente entrenado y cargado en el simulador.

Antes de ejecutar la simulación, se accedió al panel de entrenamiento del módulo de Aprendizaje por Refuerzo, mostrado en la Figura 4.15(a). En esta interfaz se configuró una duración de simulación de 10.0 s, un *timestep* de 0.5 ms y una función de recompensa de tipo balanced. En el bloque correspondiente al algoritmo de aprendizaje se seleccionó PPO, con un *learning rate* de 0.000001, un tamaño de lote (*batch size*) de 64 y un factor de descuento γ de 0.990. El registro del panel indica la finalización del entrenamiento y el guardado exitoso del modelo generado.

Posteriormente, se retornó al panel principal de simulación para cargar el modelo entrenado en el entorno *PonLab*. Como se observa en la Figura 4.15(b), la configuración del escenario mantiene 8 ONUs conectadas, selecciona Smart-RL como algoritmo DBA y establece una duración total de 10s. En la misma interfaz se visualiza el archivo del modelo RL cargado, el cual es empleado por el simulador para ejecutar la asignación de recursos durante la prueba.



(a) Panel de configuración de entrenamiento.

(b) Carga del modelo en Smart-RL.

Figura 4.15: Entorno gráfico de PonLab para el plano de control inteligente: (a) interfaz de ajuste de hiperparámetros y entrenamiento con PPO, y (b) configuración de la prueba utilizando el modelo RL generado.

La Figura 4.16 presenta la evolución temporal del *throughput* promedio registrado en el canal ascendente durante los 10 segundos de simulación con el algoritmo Smart-RL. La gráfica corresponde a una serie temporal expresada en MB/s, donde la curva muestra variaciones continuas entre muestras consecutivas durante todo el intervalo evaluado. Los registros reportan un valor mínimo de 10.079 MB/s y un valor máximo de 31.829 MB/s, con un promedio general de 20.061 MB/s. Además, se observan picos puntuales de mayor magnitud alrededor de los segundos 6.5 y 8.5, así

como descensos localizados cercanos a los segundos 6.1 y 9.7.

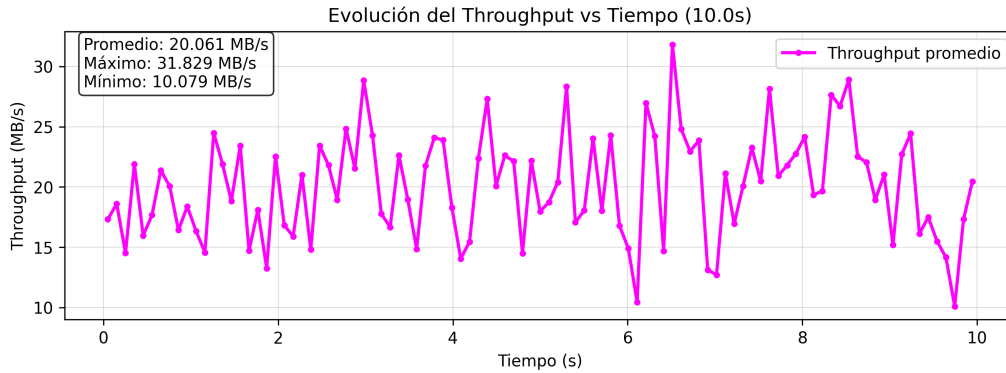


Figura 4.16: Evolución temporal del Throughput en el canal upstream bajo la gestión del agente DRL.

Por otra parte, la Figura 4.17 muestra el comportamiento temporal del retardo promedio de paquetes, denominado *Mean Delay*. La métrica se encuentra expresada en milisegundos y presenta una curva con oscilaciones frecuentes a lo largo de los 10 segundos de simulación. Los valores registrados se ubican entre un mínimo de 0.383 ms y un máximo de 0.807 ms, mientras que el promedio general corresponde a 0.593 ms. La gráfica incluye incrementos puntuales cercanos a los segundos 1.6, 5.3, 5.7 y 8.8, donde se alcanzan algunos de los valores más altos de retardo promedio.

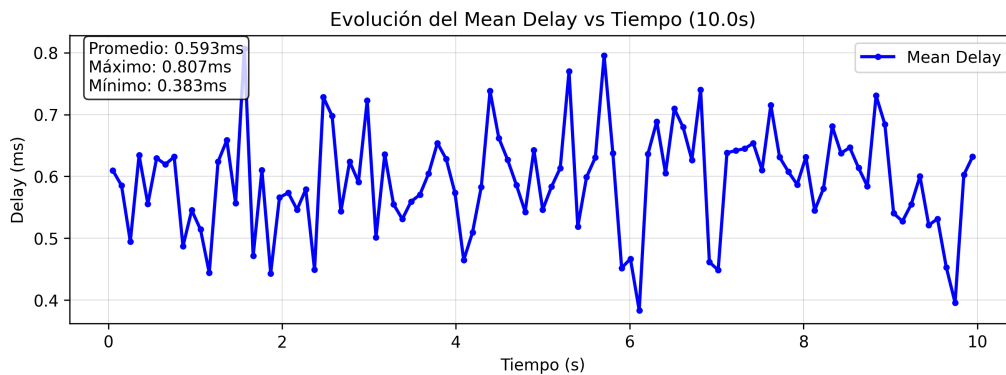


Figura 4.17: Retardo promedio registrado para las 8 ONUs bajo el plano de control inteligente.

Finalmente, la Figura 4.18 representa la evolución temporal del *Jitter IPDV Mean* medido en el enlace ascendente durante la simulación. La gráfica está expresada en milisegundos y describe la variación instantánea del retardo promedio a lo largo del periodo de 10 segundos. Los registros

muestran un valor mínimo de 0.015 ms, un máximo de 0.150 ms y un promedio general de 0.069 ms. La curva presenta fluctuaciones intermitentes, con incrementos destacados alrededor de los segundos 1.6, 5.9 y 6.0, además de una reducción marcada cercana al segundo 7.3.

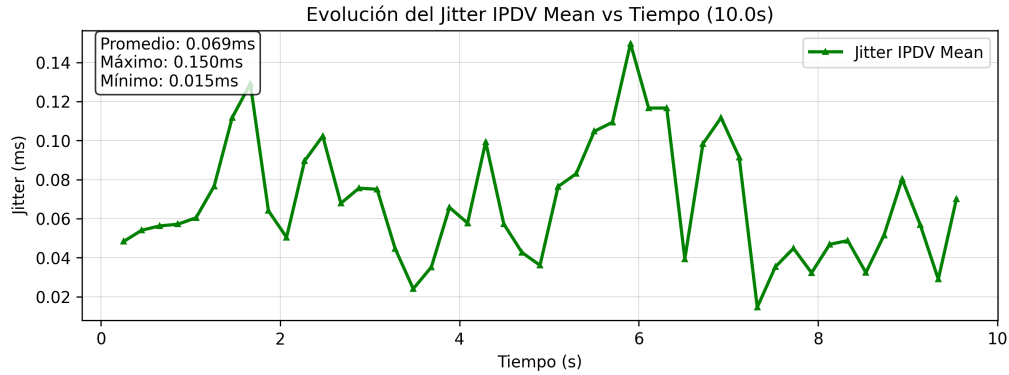


Figura 4.18: Comportamiento del Jitter IPDV Mean upstream operando con el algoritmo Smart-RL.

Según los resultados obtenidos, los bajos niveles de latencia y jitter mantenidos por el agente DRL sugieren una gestión eficiente de los recursos en el canal ascendente. No obstante, atribuir este comportamiento únicamente a una reducción de la congestión en las colas constituye, en rigor, una hipótesis preliminar, dado que no se dispone de mediciones directas de la evolución de la ocupación de los búferes ni de la tasa de pérdida de paquetes durante esta ejecución. Dicha hipótesis encuentra respaldo en el diseño de la función de recompensa *balanced* empleada durante el entrenamiento, la cual penaliza explícitamente los retardos excesivos y premia el caudal efectivo, incentivando al agente a evitar estados de saturación. Sin embargo, la verificación concluyente de esta interpretación requeriría un análisis complementario que correlacione las decisiones de asignación del agente con las dinámicas internas de las colas, aspecto que se propone como una línea de trabajo futura dentro de este entorno de validación.

4.5 Discusión de Resultados

La etapa experimental permitió validar que *PonLab Simulator* cumple con el objetivo central de esta memoria: disponer de un entorno configurable para modelar, ejecutar y comparar estrategias de asignación dinámica de ancho de banda en redes ópticas pasivas. Esta validación se relaciona directamente con la problemática descrita en el marco teórico, donde la gestión del canal ascen-

dente en arquitecturas PON depende de mecanismos DBA capaces de coordinar las ventanas de transmisión de múltiples ONUs sobre un medio compartido [11], [12].

En términos funcionales, el framework permitió construir una topología residencial G-PON compuesta por una OLT y ocho ONUs, configurar sus parámetros de operación, ejecutar simulaciones de 10 segundos y visualizar métricas de rendimiento asociadas al enlace ascendente. Además, la plataforma permitió alternar entre tres paradigmas de control con distinta naturaleza: un esquema heurístico tradicional basado en GIANT, una arquitectura con controlador centralizado SDN y un agente inteligente Smart-RL entrenado mediante Aprendizaje por Refuerzo Profundo. Esta capacidad de intercambio entre lógicas de control resulta clave para evaluar políticas DBA sin depender de implementaciones rígidas de hardware.

La validación también confirma la pertinencia del enfoque modular descrito en el diseño del sistema. El paradigma SDN permite desacoplar el plano de control del plano de datos, habilitando una gestión centralizada y programable de la red [5]. En la literatura, este tipo de control se ha explorado mediante modelos jerárquicos YANG/NETCONF y agentes SDN basados en OpenFlow para arquitecturas PON [4], [8]. Por su parte, la incorporación de DRL responde a la tendencia de formular la asignación de recursos como un problema de decisión secuencial, donde el agente observa estados de red, ejecuta acciones de asignación y recibe recompensas asociadas al desempeño obtenido [15], [18].

Por lo tanto, los resultados no deben interpretarse únicamente como la búsqueda del mayor valor en una métrica aislada, sino como una comparación del compromiso operacional entre caudal, latencia y estabilidad temporal. Este enfoque es relevante porque las soluciones basadas en aprendizaje pueden priorizar objetivos de calidad de servicio, como baja latencia o baja variabilidad, aun cuando ello implique sacrificar parte de la tasa de transferencia agregada.

4.6 Análisis comparativo de rendimiento en el enlace ascendente

La Tabla 16 consolida los valores mínimos, máximos y promedios obtenidos para *Throughput*, *Mean Delay* y *Jitter IPDV* en los tres escenarios evaluados. La tabla se ubica al inicio del análisis para entregar una visión global antes de examinar cada métrica por separado. Luego, cada figura se presenta una sola vez dentro de la subsección correspondiente, evitando duplicaciones y

manteniendo una relación directa entre la evidencia gráfica y su interpretación.

Tabla 16: Resumen comparativo de métricas de rendimiento por paradigma de control.

Métrica	Parámetro	Heurístico (GIANT)	Controlador SDN	Agente DRL (Smart-RL)
Throughput (MB/s)	Mínimo	19.429	25.684	10.079
	Máximo	78.324	68.472	31.829
	Promedio	42.900	45.607	20.061
Mean Delay (ms)	Mínimo	0.690	0.841	0.383
	Máximo	3.873	3.238	0.807
	Promedio	1.509	1.589	0.593
Jitter IPDV (ms)	Mínimo	0.139	0.089	0.015
	Máximo	0.928	0.791	0.150
	Promedio	0.422	0.376	0.069

4.6.1 Análisis de Throughput

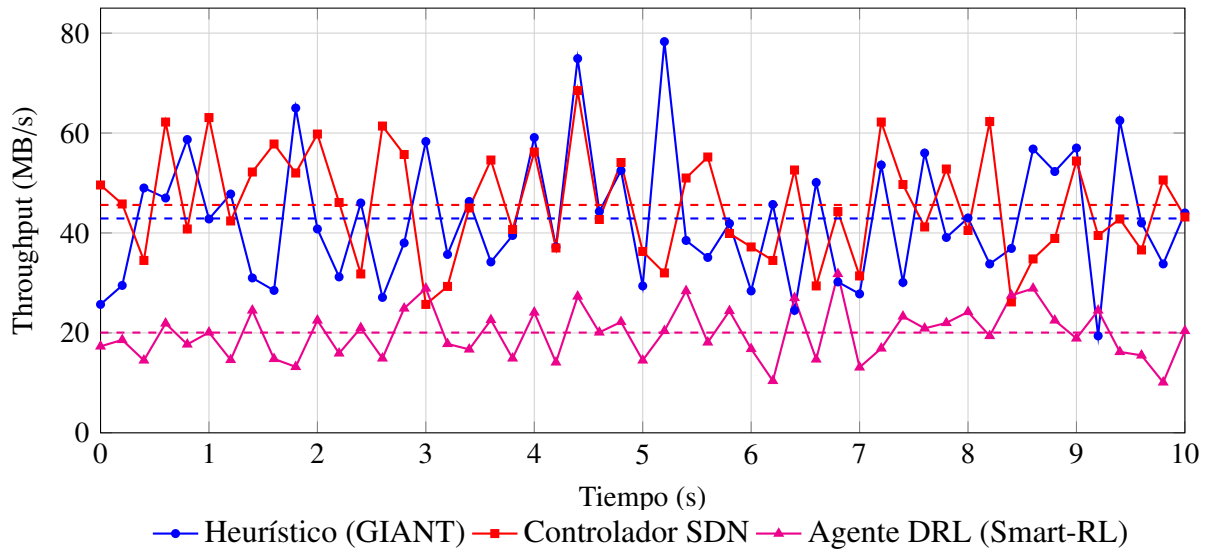


Figura 4.19: Comparación temporal del throughput en el enlace ascendente para los tres paradigmas evaluados. Las líneas segmentadas representan el valor promedio de cada enfoque durante la simulación de 10 segundos.

La Figura 4.19 muestra que el controlador SDN obtuvo el mayor *throughput* promedio del experimento, alcanzando 45.607 MB/s, seguido por el esquema heurístico GIANT con 42.900 MB/s. Esta diferencia sugiere que la capa de control centralizado permitió mejorar la utilización del canal ascendente respecto de la línea base, manteniendo un nivel de caudal elevado durante una proporción importante del intervalo de simulación. Este comportamiento es coherente con el propósito de

SDN en redes ópticas: entregar una visión centralizada del estado de la red y facilitar decisiones de asignación más flexibles que las reglas heurísticas estáticas [4], [5].

El caso GIANT presenta un comportamiento oscilatorio, con valores entre 19.429 MB/s y 78.324 MB/s. Esta variabilidad es esperable en un esquema DBA heurístico sometido a tráfico residencial con múltiples ONUs concurrentes, ya que la asignación se ajusta a solicitudes dinámicas de buffers y a ciclos de sondeo periódicos. Aun así, su promedio confirma que el algoritmo funciona como una línea base válida para evaluar el desempeño del motor de simulación, en concordancia con el rol que cumplen los algoritmos DBA tradicionales en redes GPON [12].

Por otro lado, el agente DRL alcanzó un *throughput* promedio de 20.061 MB/s, claramente inferior a GIANT y SDN. Este resultado no debe interpretarse de forma aislada como una falla del enfoque inteligente, sino como una consecuencia del objetivo aprendido por el agente y de la configuración preliminar del módulo Smart-RL. En trabajos de asignación de ancho de banda basados en aprendizaje por refuerzo, la función de recompensa puede orientar el comportamiento hacia la reducción de latencia, la estabilidad o la equidad, además de la maximización del caudal [7], [18]. En este caso, los resultados posteriores muestran que el agente adopta una política más conservadora en términos de utilización del canal, favoreciendo métricas temporales de QoS por sobre el caudal máximo.

4.6.2 Análisis de Latencia (Mean Delay)

La Figura 4.20 evidencia una de las diferencias más relevantes entre los paradigmas evaluados. Mientras GIANT y SDN mantienen promedios similares, de 1.509 ms y 1.589 ms respectivamente, el agente DRL reduce el retardo promedio a 0.593 ms. Además, el máximo registrado por Smart-RL fue de 0.807 ms, muy por debajo de los máximos observados en GIANT y SDN, equivalentes a 3.873 ms y 3.238 ms. Estos valores se mantienen ampliamente bajo el umbral de 10 ms utilizado en la literatura para evaluar comunicaciones de baja latencia en esquemas de asignación adaptativa [18].

La similitud entre GIANT y SDN en latencia promedio indica que el aumento de *throughput* obtenido por el controlador centralizado no se tradujo necesariamente en una reducción del retardo medio. En otras palabras, el plano SDN mejoró la utilización del canal, pero sus decisiones de

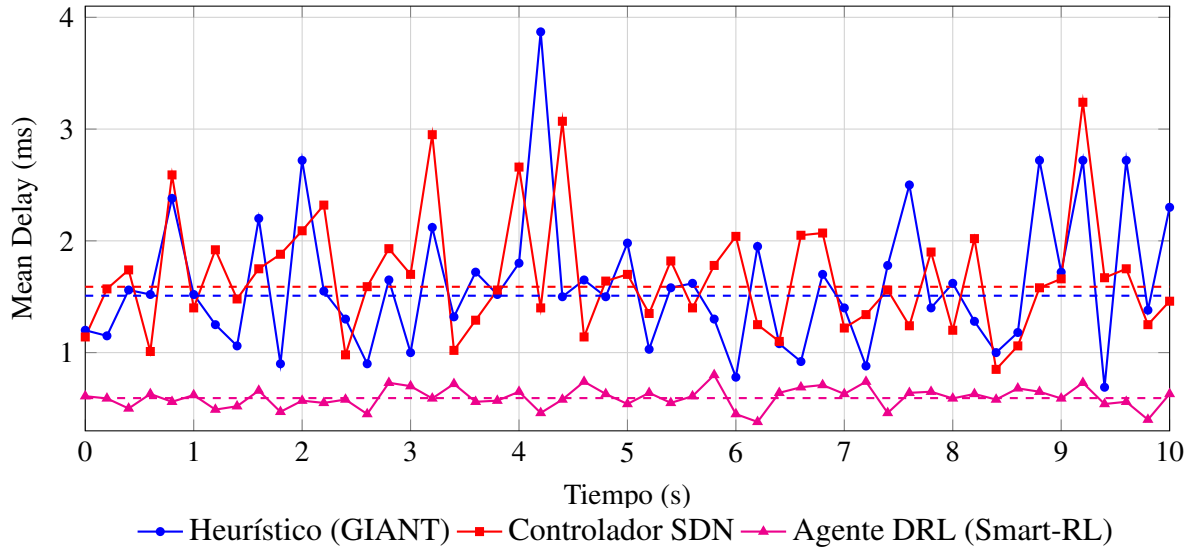


Figura 4.20: Comparación temporal del Mean Delay en el enlace ascendente para los tres paradigmas evaluados. Las líneas segmentadas representan el valor promedio de cada enfoque durante la simulación de 10 segundos.

reasignación no eliminaron completamente la acumulación temporal de paquetes en las colas. Este resultado es consistente con la idea de que el control centralizado entrega mayor programabilidad y monitoreo, pero su desempeño final depende de la política concreta de asignación implementada sobre el controlador [5], [8].

En contraste, el agente DRL mantuvo una curva de retardo considerablemente más baja y acotada durante toda la simulación. Esto sugiere que la política aprendida priorizó la reducción de tiempos de espera en los buffers, anticipando o mitigando condiciones de congestión antes de que se transformaran en picos de latencia. Este comportamiento coincide con el enfoque reportado en estudios de DRL aplicado a redes PON, donde el agente aprende a ajustar dinámicamente la asignación de ancho de banda en función del estado de la red y de recompensas asociadas al retardo [15], [18].

Desde el punto de vista operacional, el resultado refuerza la existencia de un compromiso entre caudal y latencia. SDN logra el mayor volumen promedio transmitido, pero DRL consigue el mejor desempeño temporal. Por ello, Smart-RL se posiciona como una alternativa especialmente atractiva para servicios sensibles al retardo, mientras que SDN resulta más favorable cuando la prioridad principal es maximizar la utilización del canal ascendente.

4.6.3 Análisis de Jitter (IPDV)

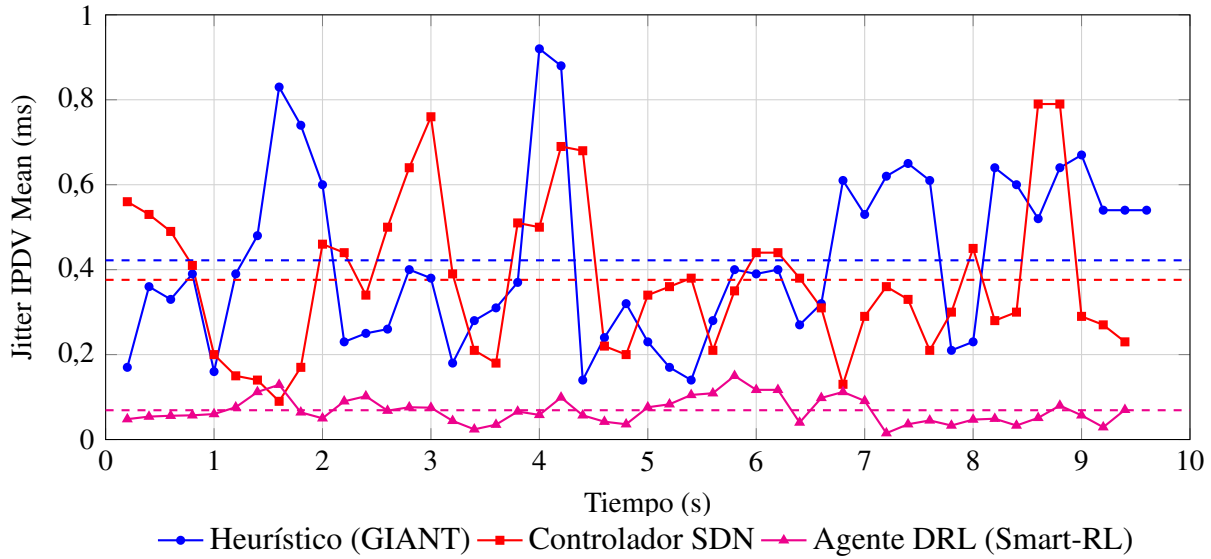


Figura 4.21: Comparación temporal del Jitter IPDV Mean en el enlace ascendente para los tres paradigmas evaluados. Las líneas segmentadas representan el valor promedio de cada enfoque durante la simulación de 10 segundos.

La Figura 4.21 confirma la tendencia observada en el análisis de latencia. El esquema SDN reduce moderadamente el *Jitter IPDV Mean* respecto de GIANT, pasando de 0.422 ms a 0.376 ms. Sin embargo, ambos enfoques presentan oscilaciones visibles y picos de variabilidad temporal, con máximos de 0.928 ms para GIANT y 0.791 ms para SDN. Esto indica que la centralización del control mejora parcialmente la estabilidad de la transmisión, pero no elimina por completo la variabilidad producida por ráfagas de tráfico, reasignaciones dinámicas y acumulación momentánea de paquetes en las colas.

El agente DRL, en cambio, presenta un comportamiento notoriamente más estable. Su jitter promedio fue de 0.069 ms y su máximo alcanzó solo 0.150 ms, valores sustancialmente menores que los de los otros dos paradigmas. Esta reducción es coherente con una política de control orientada a suavizar la entrega de paquetes y evitar variaciones bruscas del retardo. La minimización del jitter es especialmente importante para flujos sensibles a variaciones temporales, como voz, video interactivo y otros servicios con requerimientos estrictos de QoS, donde la regularidad de llegada de paquetes puede ser tan relevante como la latencia promedio [11], [18].

En conjunto, la baja latencia y el bajo jitter alcanzados por Smart-RL muestran que el agente no

solo reduce el retardo medio, sino que también estabiliza la dinámica temporal del tráfico ascendente. Esta observación respalda la utilidad del aprendizaje por refuerzo como mecanismo de control para escenarios donde la calidad temporal del servicio es prioritaria. No obstante, debido a que el *throughput* del agente fue menor, futuros ajustes de la función de recompensa deberían incorporar explícitamente un balance entre caudal, latencia y jitter, tal como proponen enfoques recientes de optimización multiobjetivo en redes PON inteligentes [7].

4.7 Interpretación global del compromiso entre caudal y QoS

Los resultados evidencian que no existe un único paradigma dominante para todas las métricas evaluadas. El controlador SDN obtuvo el mejor desempeño en *throughput*, validando la utilidad de un plano de control centralizado para mejorar la utilización del canal y sostener una distribución eficiente de recursos. Además, las métricas internas del dashboard SDN, como las 413 reconfiguraciones automáticas, las 3973 decisiones DBA procesadas y el índice de equidad de Jain de 0.997, indican que el controlador fue capaz de operar dinámicamente manteniendo una distribución prácticamente uniforme de recursos entre las ONUs. Esta capacidad se alinea con los beneficios de programabilidad, telemetría y orquestación flexible descritos para SDN en redes ópticas [4], [5].

Sin embargo, el mejor desempeño temporal fue obtenido por el agente DRL. Sus valores de *Mean Delay* y *Jitter IPDV* fueron sustancialmente menores que los de GIANT y SDN, lo que sugiere una política de asignación enfocada en prevenir congestión y estabilizar el retardo. Este comportamiento es consistente con los fundamentos del aprendizaje por refuerzo aplicados a redes PON, donde el agente ajusta sus decisiones a partir de la interacción con el entorno y de señales de recompensa asociadas al desempeño observado [15], [18].

De esta manera, GIANT cumple el rol de línea base para validar el funcionamiento del motor de simulación; SDN representa una mejora en control, monitoreo y eficiencia de caudal; y DRL demuestra potencial para escenarios donde las restricciones de baja latencia y baja variabilidad son prioritarias. Esta comparación confirma que PonLab permite estudiar no solo resultados absolutos, sino también criterios de operación diferenciados según el tipo de servicio que se desea priorizar.

4.8 Alcances y limitaciones de la evaluación

La interpretación de los resultados debe considerar las condiciones experimentales establecidas en el diseño de validación. En primer lugar, las simulaciones se ejecutaron sobre una topología residencial de una OLT y ocho ONUs durante 10 segundos, por lo que los valores obtenidos constituyen una validación operacional inicial del framework y no una caracterización exhaustiva de todos los escenarios PON posibles. En segundo lugar, el modelo físico asumió condiciones ideales de transmisión, despreciando atenuación, pérdidas por inserción y otros efectos ópticos, con el objetivo de aislar el impacto de las políticas de asignación de recursos sobre las métricas de red.

Por otra parte, una limitación de carácter práctico está asociada directamente a la capacidad de cómputo disponible durante la fase de pruebas. Si bien resultaría metodológicamente deseable ejecutar simulaciones de larga duración (del orden de una hora, es decir, 3600 segundos) que aproximen condiciones operativas reales y permitan observar el comportamiento en régimen estacionario, al extender el tiempo de ejecución a dicho intervalo se evidenció un problema crítico: el elevado volumen de eventos encolados y la acumulación progresiva de estados en los procesos internos del simulador saturaban los búferes del sistema, provocando que el software quedara bloqueado e impidiera completar la simulación. En consecuencia, no fue posible verificar exhaustivamente la estabilidad ni el rendimiento del entorno *PonLab* bajo cargas sostenidas prolongadas, lo que acota la validación funcional realizada a ventanas temporales reducidas y constituye una limitación relevante para la generalización de los resultados.

Adicionalmente, el módulo DRL fue evaluado como una integración preliminar. En consecuencia, sus resultados deben entenderse como una demostración de la capacidad de *PonLab* para incorporar agentes inteligentes, más que como una política definitiva de optimización. La literatura muestra que los enfoques basados en aprendizaje pueden mejorar el desempeño de redes PON cuando se diseñan recompensas adecuadas y se entrenan bajo escenarios representativos [6], [7], [15]. Por ello, futuras iteraciones deberían explorar múltiples semillas de simulación, perfiles de tráfico heterogéneos, tiempos de ejecución más extensos y funciones de recompensa multiobjetivo que equilibren *throughput*, latencia, jitter y equidad.

5. Conclusiones

La presente memoria abordó la necesidad de contar con un entorno flexible y programable para la gestión del canal ascendente en Redes Ópticas Pasivas (PON) mediante el diseño, desarrollo y validación de **PonLab Simulator**. El trabajo dio respuesta satisfactoria al **Objetivo General** al entregar un *framework* modular e integrado que supera la rigidez de las herramientas tradicionales, permitiendo modelar topologías, configurar parámetros operacionales, simular la asignación de recursos y evaluar comparativamente múltiples políticas de control bajo un mismo entorno experimental.

Desde la perspectiva funcional y de arquitectura (**OE1** y **OE3**), la plataforma demostró capacidad para abstraer la complejidad de la capa física y centrarse en la capa MAC. La implementación de una Interfaz Gráfica de Usuario (GUI) en Python/PyQt5, acoplada a un motor de eventos discretos con ciclo de sondeo de $125\mu\text{s}$ y a herramientas de exportación estructurada (.json, .png, .txt), eliminó la dependencia de configuraciones por consola, facilitando el diseño interactivo de topologías (nodos OLT y ONU), el ajuste de perfiles de tráfico y la visualización de métricas cuantitativas en tiempo real.

En relación con la integración y evaluación comparativa de paradigmas de control (**OE2**), la herramienta permitió contrastar tres enfoques heterogéneos: la heurística tradicional (ej. *GIANT*) como línea base, una arquitectura de control centralizado *SDN* y el agente inteligente *Smart-RL* basado en Aprendizaje por Refuerzo Profundo (DRL). La validación experimental sobre una topología residencial G-PON de 1 OLT y 8 ONUs (2,5 Gbps DS / 1,25 Gbps US) en ventanas de 10 s (**OE4**) evidenció un claro compromiso operacional (*trade-off*):

- **Control Centralizado (SDN):** Maximizó la eficiencia de transmisión al lograr el mayor *throughput* promedio (45,607 MB/s frente a 42,900 MB/s de *GIANT* y 20,061 MB/s de DRL), respaldado por 413 reconfiguraciones automáticas, 3973 decisiones DBA procesadas y un índice de equidad de Jain de 0,997.
- **Control Inteligente (Smart-RL):** Destacó en las métricas temporales de Calidad de Servicio (QoS), alcanzando un retardo medio promedio de 0,593 ms y un *jitter* IPDV de 0,069 ms, valores sustancialmente menores a los de *GIANT* y *SDN*, aunque a costa de una reducción

en el caudal agregado.

A nivel metodológico, la investigación declaró abiertamente sus limitaciones prácticas: las simulaciones se acotaron a ventanas de 10s debido a que ejecuciones de larga duración (3600s) provocaron la saturación de memoria y el bloqueo del sistema por acumulación de eventos encolados; se asumieron condiciones ópticas ideales sin degradación física; la integración de DRL actuó como una prueba de concepto con recompensa orientada al retardo; y la evaluación se basó en ejecuciones únicas.

Trabajo Futuro y Proyecciones

A partir de los hallazgos y limitaciones identificadas, se proyectan las siguientes líneas de trabajo futuro: (1) diseñar funciones de recompensa multiobjetivo para el agente DRL que equilibren simultáneamente *throughput*, latencia, *jitter* y equidad; (2) optimizar la gestión de memoria y el motor de eventos para soportar simulaciones de larga duración en régimen estacionario bajo perfiles de tráfico heterogéneos (residencial, empresarial, IoT y 5G/6G); (3) incorporar modelos de la capa física que consideren atenuación, dispersión y pérdidas por inserción; y (4) implementar un protocolo de validación estadística basado en múltiples réplicas independientes (30 o más) con diferentes semillas aleatorias, reportando intervalos de confianza (95 % o 99 %) y aplicando pruebas de hipótesis (ANOVA y prueba *t*).

Referencias

- [1] D. Roy, A. S. Rao, T. Alpcan, G. Das y M. Palaniswami, “Achieving QoS for bursty uRLLC applications over passive optical networks,” *Journal of Optical Communications and Networking*, vol. 14, n.º 5, págs. 411-425, 2022.
- [2] C. M. Assi, Y. Ye, S. Dixit y M. A. Ali, “Dynamic bandwidth allocation for quality-of-service over Ethernet PONs,” *IEEE Journal on selected Areas in Communications*, vol. 21, n.º 9, págs. 1467-1477, 2003.
- [3] K. A. Memon, S. S. Jaffer, M. A. Qureshi y K. K. Qureshi, “Dynamic bandwidth allocation in time division multiplexed passive optical networks: a dual-standard analysis of ITU-T and IEEE standard algorithms,” *PeerJ Computer Science*, vol. 11, e2863, 2025.
- [4] Z. Liang, Z. Yang y J. Tang, “A SDN Enabled PON Controller based on Hierarchical Models,” en *Optoelectronics Global Conference (OGC)*, IEEE, 2022.
- [5] K. Nsafoa-Yeboah et al., “Software-Defined Networks for Optical Networks Using Flexible Orchestration: Advances, Challenges, and Opportunities,” *Journal of Computer Networks and Communications*, vol. 2022, 2022. doi: 10.1155/2022/5037702.
- [6] J. Hatem, A. Dhaini y S. Elbassuoni, “Deep Learning-Based Dynamic Bandwidth Allocation for Future Optical Access Networks,” *IEEE Access*, vol. PP, págs. 1-1, jul. de 2019. doi: 10.1109/ACCESS.2019.2929480.
- [7] B. Cao, Q. Zhao, Y. Hong y X. Fan, “Network performance evaluation criterion model based on large connections for low latency in industrial 50G-PON network,” *Optical Fiber Technology*, vol. 90, págs. 104 107, 2025, issn: 1068-5200. doi: <https://doi.org/10.1016/j.yofte.2024.104107>. dirección: <https://www.sciencedirect.com/science/article/pii/S1068520024004528>.
- [8] D. de Pintos, N. Merayo, C. Sangrador, J. C. Aguado, I. de Miguel y R. J. D. Barroso, “Software defined networking agent demonstration to enable configuration and management of XGS-PON architectures,” *Journal of Optical Communications and Networking*, vol. 15, n.º 9, págs. 620-637, 2023. doi: 10.1364/JOCN.494694.

- [9] C. Park, H. S. Jung, D. H. Han e Y. Lee, “Performance analysis of DBA scheme with interleaved polling algorithm in an Ethernet PON,” en *Proceedings. ISCC 2004. Ninth International Symposium on Computers And Communications (IEEE Cat. No.04TH8769)*, vol. 2, jul. de 2004, 792-797 Vol.2. doi: 10.1109/ISCC.2004.1358637.
- [10] W. Xiong y X. Zhang, “Research on the Multi Sub-Cycle dynamic bandwidth allocation algorithm,” en *Proceedings of 2011 International Conference on Computer Science and Network Technology*, vol. 4, dic. de 2011, págs. 2298-2302. doi: 10.1109/ICCSNT.2011.6182433.
- [11] Y. Weng, D. Zhang, J. Jin, D. Shu y M. Wu, “The Impact of the Equalization Delay on Latency Performance in SR-DBA Scheme for TDM-PON,” en *2024 Asia Communications and Photonics Conference (ACP) and International Conference on Information Photonics and Optical Communications (IPOC)*, nov. de 2024, págs. 1-3. doi: 10.1109/ACP/IPOC63121.2024.10809525.
- [12] T. Thangappan, E. Manimaran, A. Arasu, R. Arulprakash y J. Harish Ganapathi, “Review of Dynamic Bandwidth Allocation in GPON,” en *2020 International Conference on Communication and Signal Processing (ICCSP)*, 2020, págs. 884-888. doi: 10.1109/ICCSP48568.2020.9182151.
- [13] G. Simon et al., “200 Gb/s Coherent Point-to-Multipoint Coexistence With 50G-PON for Next-Generation Optical Access,” *IEEE Photonics Technology Letters*, vol. 36, n.º 10, págs. 665-668, 2024. doi: 10.1109/LPT.2024.3385681.
- [14] G. Kramer, B. Mukherjee y G. Pesavento, “Interleaved polling with adaptive cycle time (IPACT): a dynamic bandwidth distribution scheme in an optical access network,” *Photonic Network Communications*, vol. 4, n.º 1, págs. 89-107, 2002.
- [15] Q. Zhou, J. Zhu, J. Zhang, Z. Jia, B. Huberman y G.-K. Chang, *Intelligent Bandwidth Allocation for Latency Management in NG-EPON using Reinforcement Learning Methods*, ene. de 2020. doi: 10.48550/arXiv.2001.07698.
- [16] R. S. Sutton y A. G. Barto, “Reinforcement learning: An introduction second edition,” *Adaptive computation and machine learning: The MIT Press, Cambridge, MA and London*, 2018.

- [17] J. F. Kurose et al., *Computer networking: a top-down approach*, 2021.
- [18] L. Ruan, M. P. I. Dias y E. Wong, "Towards self-adaptive bandwidth allocation for low-latency communications with reinforcement learning," *Optical Switching and Networking*, vol. 37, pág. 100 567, 2020, ISSN: 1573-4277. DOI: <https://doi.org/10.1016/j.osn.2020.100567>. dirección: <https://www.sciencedirect.com/science/article/pii/S1573427719301031>.
- [19] A. Dixit, B. Lannoo, G. Das, D. Colle, M. Pickavet y P. Demeester, "Dynamic bandwidth allocation with SLA awareness for QoS in ethernet passive optical networks," *Journal of Optical Communications and Networking*, vol. 5, n.º 3, págs. 240-253, 2013.
- [20] T. Orphanoudakis, H.-C. Leligou, E. Kosmatos y J. D. Angelopoulos, "Performance evaluation of GPON vs EPON for multi-service access," *International Journal of Communication Systems*, vol. 22, n.º 2, págs. 187-202, 2009.
- [21] P. Morales et al., "Multi-band environments for optical reinforcement learning gym for resource allocation in elastic optical networks," en *2021 International Conference on Optical Network Design and Modeling (ONDM)*, IEEE, 2021, págs. 1-6.
- [22] X. Wu et al., "An XG-PON module for the NS-3 network simulator," 2013.
- [23] B. Lantz, A. A. Díaz-Montiel, J. Yu, C. Rios, M. Ruffini y D. Kilper, "Demonstration of software-defined packet-optical network emulation with mininet-optical and onos," en *Optical Fiber Communication Conference*, Optica Publishing Group, 2020, M3Z-9.

Anexos

A. Validación del Cumplimiento de Objetivos Específicos

En el presente anexo se detalla la correspondencia explícita entre los Objetivos Específicos (OE) planteados al inicio de esta memoria y los desarrollos, implementaciones y resultados expuestos a lo largo del documento. La Tabla 17 presenta una síntesis del cumplimiento de cada objetivo junto con las secciones principales donde se evidencia su ejecución.

Tabla 17: Matriz de trazabilidad y cumplimiento de Objetivos Específicos.

Objetivo	Descripción	Capítulo/Sección	Estado
OE1	Diseñar una arquitectura para asignación de recursos en redes PON mediante un entorno de simulación.	Cap. 3 (Sec. 3.3, 3.5)	Cumplido
OE2	Integrar técnicas heurísticas, SDN e IA (DRL) para la optimización de recursos.	Cap. 3 (Sec. 3.4) y Cap. 4	Cumplido
OE3	Integrar herramientas gráficas y textuales para visualización de topologías y resultados con bibliotecas de Python.	Cap. 3 (Sec. 3.1, 3.5) y Cap. 4	Cumplido
OE4	Validar el rendimiento del simulador en escenarios de prueba reales contrastando con la literatura.	Cap. 3 (Sec. 3.6), Cap. 4 y Cap. 5	Cumplido

A.1 Validación del OE1: Diseño de la arquitectura para asignación de recursos

Objetivo: *Diseñar una arquitectura que influya en la asignación de recursos en redes PON, con el fin de representar adecuadamente escenarios realistas, mediante el desarrollo de un entorno de simulación que las integre de forma precisa.*

Evidencia de cumplimiento:

- **Modelado de Capa MAC:** En el Capítulo 3 (Sección 3.4.4) se detalla la arquitectura de eventos discretos del motor de simulación de *PonLab*, incorporando la técnica de verificación temporal automática (*event-less polling*) basada en ciclos estándar de $125\ \mu\text{s}$ para simular de forma precisa el ciclo de reporte y concesión (*Report/Gate*).
- **Estructura Modular:** En la Sección 3.4 se especifican los 7 módulos funcionales de la plataforma (M1 a M7), permitiendo desacoplar la topología física, la generación de tráfico por contenedores de tráfico (T-CONTs) y la lógica de asignación dinámica de ancho de banda (DBA).
- **Escenarios Realistas:** Se implementó el soporte para redes G-PON con velocidades de línea de 2.5 Gbps descendente y 1.25 Gbps ascendente, junto con perfiles de tráfico residencial y empresarial dinámicos.

A.2 Validación del OE2: Integración de técnicas heurísticas, SDN e Inteligencia Artificial

Objetivo: *Integrar diferentes técnicas, incluyendo heurísticas, agentes de redes definidas por “software” e inteligencia artificial, para optimizar la asignación de recursos comparando con metodologías ya existentes.*

Evidencia de cumplimiento:

- **Enfoque Heurístico:** Se incorporó e implementó el algoritmo tradicional GIANT (*GigaPON Access Network*) como línea base de comparación en el canal ascendente.
- **Control Centralizado SDN:** Se desacopló el plano de control del plano de datos mediante la integración de una OLT basada en SDN (Sección 3.4.5), habilitando reconfiguraciones automáticas y orquestación dinámica del ancho de banda.
- **Inteligencia Artificial (DRL):** Se desarrolló e integró el módulo Smart-RL (Sección 3.4.7), el cual utiliza el algoritmo de Aprendizaje por Refuerzo Profundo PPO (*Proximal Policy Optimization*) a través de las bibliotecas Gymnasium y Stable-Baselines3.

- **Evaluación Comparativa:** En el Capítulo 4 se ejecutaron pruebas bajo las mismas condiciones para los tres paradigmas, comparando directamente métricas de *throughput*, latencia y *jitter*.

A.3 Validación del OE3: Integración de herramientas gráficas y textuales

Objetivo: *Integrar herramientas gráficas y textuales para visualizar topologías, parámetros de red y resultados de asignación mediante el uso de bibliotecas gráficas de Python.*

Evidencia de cumplimiento:

- **Interfaz Gráfica de Usuario (GUI):** Implementada íntegramente en Python utilizando la biblioteca PyQt5 (3.5), la cual incluye un canvas interactivo para el diseño de topologías (*drag-and-drop*), paneles de propiedades de dispositivos (OLT/ONU) y paneles específicos para la configuración de simulación y entrenamiento de modelos RL.
- **Visualización de Resultados:** Integración de bibliotecas de análisis de datos de Python (Matplotlib, Seaborn, Pandas y NumPy) para la generación y despliegue automático de series temporales de *throughput*, *mean delay*, niveles de buffer y *jitter* IPDV.
- **Formatos Textuales y Estructurados:** Generación de logs en consola en tiempo real, exportación del estado de red en archivos JSON (`metadata.json`, `network_state.json`), datos comprimidos de simulación (`.json.gz`) y reportes ejecutivos en texto plano (`RESUMEN.txt`).

A.4 Validación del OE4: Rendimiento del simulador en escenarios realistas y contraste con la literatura

Objetivo: *Validar el rendimiento del “simulador” para escenarios de prueba que reflejen condiciones reales de uso y carga contrastando con la literatura.*

Evidencia de cumplimiento:

- **Escenario Experimental:** Se configuró y ejecutó exitosamente una topología residencial G-PON de 1 OLT y 8 ONUs concurrentes durante un intervalo continuo de 10 segundos (Capítulo 3, Sección 3.6).

- **Métricas de QoS Evaluadas:** Se cuantificó cuantitativamente el comportamiento de la red a través de la Tasa de Transferencia (*Throughput*), Latencia Promedio (*Mean Delay*), *Jitter* (*IPDV*) e Índice de Equidad de Jain (alcanzando 0.997 bajo control SDN).
- **Contraste con la Literatura y Discusión:** En la Sección 4.5 y Capítulo 5 se contrastan los resultados obtenidos (latencias inferiores a 1 ms para el agente DRL y *throughputs* superiores a 45 MB/s para SDN) con los umbrales críticos de baja latencia (< 10 ms) documentados en los trabajos de Ruan et al., Zhou et al. y Cao et al., analizándose además de forma transparente las fortalezas y limitaciones computacionales del entorno de simulación.