

**ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA  
CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD  
DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING**



**Cristian David Rocha Fernández**

**Universidad del Cauca**

**Facultad de Ingeniería Electrónica y Telecomunicaciones Departamento de  
Telecomunicaciones**

**Grupo I+D Nuevas Tecnologías en Telecomunicaciones - GNTT**

**Popayán, 2022**

**ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA  
CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD  
DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING**



**Cristian David Rocha Fernández**

**Trabajo de Grado presentado como requisito para obtener el título de  
Ingeniero en Electrónica y Telecomunicaciones**

**Director: PhD. Ing. José Giovanni López Perafán**

**Universidad del Cauca**

**Facultad de Ingeniería Electrónica y Telecomunicaciones Departamento de  
Telecomunicaciones**

**Grupo I+D Nuevas Tecnologías en Telecomunicaciones - GNTT**

**Popayán, 2022**



**ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING**

---

**CONTENIDO**

1. INTRODUCCIÓN.....	1
1.1. PLANTEAMIENTO DEL PROBLEMA .....	1
1.2. ESCENARIO DE MOTIVACIÓN.....	2
1.3. OBJETIVOS.....	3
1.4. ENFOQUE .....	3
1.5. ALCANCE .....	6
1.6. APORTES.....	6
1.7. ORGANIZACIÓN DEL TRABAJO DE GRADO .....	6
2. GENERALIDADES DE REDES ÓPTICAS ELÁSTICAS DEFINIDAS POR SOFTWARE Y <i>DEEP LEARNING</i> .....	8
2.1. REDES ÓPTICAS ELÁSTICAS DEFINIDAS POR SOFTWARE .....	8
2.2. GENERALIDADES DE <i>DEEP LEARNING</i> .....	21
3. CONMUTACIÓN ÓPTICA DE RÁFAGAS Y ENRUTAMIENTO Y ASIGNACIÓN DE ESPECTRO .....	28
3.1. CONMUTACIÓN ÓPTICA DE RÁFAGAS (OBS) .....	28
3.2. ENRUTAMIENTO Y ASIGNACIÓN DE ESPECTRO.....	33
4. DISEÑO E IMPLEMENTACIÓN .....	47
4.1. METODOLOGÍA DE TRABAJO .....	47
4.4. SELECCIÓN DE HERRAMIENTA DE SIMULACIÓN.....	50
4.5. IMPLEMENTACIÓN DE RED.....	54
4.6. DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS.....	60
5. ESCENARIOS DE SIMULACIÓN Y PRUEBAS .....	75
5.1. INTRODUCCIÓN .....	75
5.2. CASOS DE SIMULACIÓN .....	76
5.3. PARÁMETROS DE CONFIGURACIÓN PARA ESCENARIOS DE SIMULACIÓN .....	77



**ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING**

---

5.4.	EJECUCIÓN DE LAS SIMULACIONES .....	79
5.5.	ANÁLISIS DE RESULTADOS .....	80
6.	CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS .....	96
6.1.	CONCLUSIONES .....	96
6.2.	RECOMENDACIONES .....	97
6.3.	TRABAJOS FUTUROS.....	98



**ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING**

---

**LISTA DE FIGURAS**

Figura 1.1. Diagrama de flujo del enfoque del trabajo de grado. ....	5
Figura 2.1. Esquema general de un sistema WDM. Tomada de [9].....	10
Figura 2.2. Uso del espectro a) WDM b) EON. Tomada de [16].....	14
Figura 2.3. Propiedades de segmentación y agregación en EON. Tomada de [17]. .....	15
Figura 2.4. Arquitectura de SDN. Tomada de [2].....	20
Figura 2.5. Arquitectura general de SDON. Tomada de [25] .....	21
Figura 2.6. Diseño básico de un sistema de RL. Tomada de [30] .....	24
Figura 2.7. Estructura de una red neuronal profunda con tres capas ocultas. Tomada de [33].....	25
Figura 3.1. Arquitectura de una Red por OBS. Tomada de [38] .....	29
Figura 3.2. Arquitectura del Nodo de Ingreso en OBS. Tomada de [37].....	30
Figura 3.3. Efectos de la Carga en Esquema de Ensamble de Ráfaga basado en tiempo y longitud de ráfaga. Tomada de [18] .....	33
Figura 3.4. Ejemplo de Uso de Espectro para una EON. Tomada de G.694.1. ....	35
Figura 3.5. Establecimiento de conexión en una red FlexGrid considerando las restricciones de continuidad y contigüidad. Tomada de [16] .....	36
Figura 3.6. Escenarios de Asignación de espectro fijo en el tiempo. a) Subuso del espectro disponible. b) Espectro Disponible Insuficiente. Tomada de [42] .....	40
Figura 3.7. Escenarios de Asignación de espectro semi-elástico en el tiempo. a) Reducción de Espectro. b) Expansión de Espectro. Tomada de [42].....	41
Figura 3.8. Escenarios de Asignación de Espectro Elástico en el tiempo. a) Expansión/Contracción de Espectro. b) Reasignación de Espectro. Tomada de [41].....	42
Figura 3.9. Ejemplo de uso de un segmento del espectro de una red. Tomada de [46].....	46
Figura 4.1. Diagrama de flujo de la metodología empleada. Adaptada de [47].....	47
Figura 4.2. Diagrama de Flujo de Metodología de Simulación. ....	49
Figura 4.3. Topología NSFNeT con distancias aproximadas entre nodos. Tomada de [53].....	50
Figura 4.4. Módulo Fuente/Destino implementado en OMNeT++.....	55
Figura 4.5. Módulo de Nodo Frontera implementado en OMNeT++.....	56
Figura 4.6. Módulo de Nodo Frontera de Ingreso implementado en OMNeT++... ..	56
Figura 4.7. Módulo de Nodo de control implementado en OMNeT++.....	58



**ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING**

---

Figura 4.8. Módulo Dynamic_WRON_Control implementado en OMNeT++. .....	59
Figura 4.9. Red OBS/SDEON basada en la topología NSFNeT implementada en OMNeT++. .....	60
Figura 4.10. Diagrama de flujo de algoritmo Dijkstra. ....	62
Figura 4.11. Pseudocódigo para el algoritmo Dijkstra. Tomada de [65].....	63
Figura 4.12. Código para Algoritmo Dijkstra en lenguaje C++. ....	64
Figura 4.13. Diagrama de flujo de algoritmo First-Fit. Adaptada de [67].....	66
Figura 4.14. Porción de código en Python para la definición de la red y el entorno. ....	68
Figura 4.15. Porción de código en Python para la creación de la red neuronal profunda.....	69
Figura 4.16. Porción de código en Python para la creación del agente DRL y sus métodos. ....	70
Figura 4.17. Porción de código en Python para la implementación del método para la evaluación del agente. ....	71
Figura 4.18. Diagrama de flujo para el desarrollo de algoritmo cognitivo basado en DRL. ....	72
Figura 4.19. Diagrama de Ciclo Cognitivo. Tomada de [37] .....	73
Figura 4.20. Modelo del Ciclo Cognitivo OODA. Tomada de [37].....	74
Figura 5.1. Probabilidad de bloqueo de red vs Tráfico con carga media. a) 1.25 Gbps b) 2.5 Gbps.....	82
Figura 5.2. Probabilidad de bloqueo de red vs Tráfico con carga alta. a) 1.25 Gbps b) 2.5 Gbps .....	83
Figura 5.3. Probabilidad de bloqueo de red (con algoritmo clásico basado en SCC) vs Tráfico, Velocidades de transmisión de 1.25 y 2.5 Gbps. a) Carga media b) Carga alta. ....	85
Figura 5.4. Probabilidad de bloqueo de red (con algoritmo cognitivo basado en DL) vs Tráfico, Velocidades de transmisión de 1.25 y 2.5 Gbps. a) Carga media b) Carga alta. ....	86
Figura 5.5. Retardo extremo a extremo de ráfaga vs Tráfico. Velocidades de 1.25 Gbps y 2.5 Gbps. a) Carga baja b) Carga media c) Carga alta .....	88
Figura 5.6. Probabilidad de bloqueo vs Tráfico. a) Carga media b) Carga alta.....	91
Figura 5.7. Retardo extremo a extremo vs Tráfico. a) Carga baja b) Carga media c) Carga alta .....	93



**ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING**

---

**LISTA DE TABLAS**

Tabla 3.1. Resumen Comparativo de Tipos de Enrutamiento. Adaptada de [16]..	38
Tabla 4.1. Comparación de herramientas de simulación. Adaptada de [46] .....	53
Tabla 5.1. Escenarios de simulación para pruebas. ....	76
Tabla 5.2. Parámetros de configuración de la red NSFNeT OBS/SDEON .....	78
Tabla 5.3. Especificaciones técnicas de los equipos utilizados .....	79
Tabla 5.4. Parámetros configurados para el subcaso 1.....	80
Tabla 5.5. Parámetros configurados para el subcaso 2.....	89
Tabla 5.6. Síntesis de los resultados de simulación .....	94



ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

## ACRÓNIMOS

<b>AE</b>	<i>Auto Encoder</i> , Codificador Automático
<b>AI</b>	<i>Artificial Intelligence</i> , Inteligencia Artificial
<b>AR</b>	<i>Adaptive Routing</i> , Enrutamiento Adaptativo
<b>AST</b>	<i>Average Setup Time</i> , Tiempo Medio de Configuración
<b>BCP</b>	<i>Burst Control Packet</i> , Paquete de Control de Ráfaga
<b>BP</b>	<i>Blocking Probability</i> , Probabilidad de bloqueo
<b>BVT</b>	<i>Bandwidth Variable Transponder</i> , Transpondedor de Ancho de Banda Variable
<b>BV-WXC</b>	<i>Bandwidth Variable-Wavelength Cross Connect</i> , Conmutador Cruzado de Ancho de Banda Variable
<b>CF</b>	<i>Central Frequency</i> , Frecuencia Central
<b>CNN</b>	<i>Convolutional Neural Network</i> , Red Neuronal Convolutiva
<b>DBM</b>	<i>Deep Boltzmann Machine</i> , Máquina de Boltzmann Profunda
<b>DBN</b>	<i>Deep Belief Network</i> , Red de Creencia Profunda
<b>DDQN</b>	<i>Double Deep Q-Network</i> , Red Q Profunda Doble
<b>DL</b>	<i>Deep Learning</i> , Aprendizaje Profundo
<b>DQN</b>	<i>Deep Q-Network</i> , Red Q Profunda
<b>DRL</b>	<i>Deep Reinforcement Learning</i> , Aprendizaje por Refuerzo Profundo
<b>DW</b>	<i>Distance Weight</i> , Algoritmo de menor distancia
<b>EF</b>	<i>Exact Fit</i> , Algoritmo de ajuste exacto
<b>EON</b>	<i>Elastic Optical Network</i> , Red Elástica Óptica
<b>ER</b>	<i>Experience Replay</i> , Repetición de Experiencia
<b>ESA</b>	<i>Elastic Spectrum Allocation</i> , Asignación de espectro elástico en el tiempo
<b>FAR</b>	<i>Fixed Alternate Routing</i> , Enrutamiento Fijo Alternado
<b>FF</b>	<i>First Fit</i> , Algoritmo de primer ajuste
<b>FLF</b>	<i>First-Last Fit</i> , Algoritmo de primer y último ajuste
<b>FR</b>	<i>Fixed Routing</i> , Enrutamiento Fijo
<b>FSA</b>	<i>Fixed Spectrum Allocation</i> , Asignación de espectro fijo en el tiempo
<b>GAN</b>	<i>Generative Adversarial Network</i> , Red Generativa Adversaria
<b>GNN</b>	<i>Graph Neural Network</i> , Red Neuronal de Grafos
<b>HW</b>	<i>Hop Weight</i> , Algoritmo de peso por salto
<b>IDE</b>	<i>Integrated Development Environment</i> , Entorno de Desarrollo Integrado
<b>IETF</b>	<i>Internet Engineering Task Force</i> , Grupo de Trabajo de Ingeniería de Internet



**ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING**

---

<b>ILP</b>	<i>Integer Lineal Programming, Programación Lineal Entera</i>
<b>IP</b>	<i>Internet Protocol, Protocolo de Internet</i>
<b>ITU</b>	<i>International Telecommunication Union, Unión Internacional de Telecomunicaciones</i>
<b>KSP</b>	<i>K-Shortest Paths, Algoritmo de K caminos más cortos</i>
<b>LCR</b>	<i>Least Congested Routing, Enrutamiento por Menos Congestión</i>
<b>LF</b>	<i>Last Fit, Algoritmo de último ajuste</i>
<b>LU</b>	<i>Least Used, Algoritmo del menos usado</i>
<b>ML</b>	<i>Machine Learning, Aprendizaje Automático</i>
<b>MNH</b>	<i>Minimum Number of Hops, Algoritmo de mínimo número de saltos</i>
<b>MSP</b>	<i>Modified Dijkstra's Shortest Path, Algoritmo Dijkstra modificado de camino más corto</i>
<b>MU</b>	<i>Most Used, Algoritmo del más usado</i>
<b>NBI</b>	<i>Northbound Interfaces, Interfaces hacia el Norte</i>
<b>NC-TUNS</b>	<i>National Chiao Tung University Network Simulator, Simulador de Red de la Universidad Nacional Chiao Tung</i>
<b>NFV</b>	<i>Network Function Virtualization, Virtualización de Funciones de Red</i>
<b>NOS</b>	<i>Network Operating System, Sistema Operativo de Red</i>
<b>NS-3</b>	<i>Network Simulator 3, Simulador de Red 3</i>
<b>NSFNeT</b>	<i>National Science Foundation Network, Fundación Nacional de Ciencias</i>
<b>N-WDM</b>	<i>Nyquist Wavelength Division Multiplexing, Multiplexación Nyquist por División de Longitud de Onda</i>
<b>OADM</b>	<i>Optical Add-Drop Multiplexer, Multiplexor Óptico de Adición-Sustracción</i>
<b>OBS</b>	<i>Optical Burst Switching, Conmutación Óptica de Ráfagas</i>
<b>OCS</b>	<i>Optical Circuit Switching, Conmutación Óptica de Circuitos</i>
<b>OFDM</b>	<i>Orthogonal Frequency-Division Multiplexing, Multiplexación por División de Frecuencias Ortogonales</i>
<b>OFF</b>	<i>OpenFlow Protocol, Protocolo OpenFlow</i>
<b>OPS</b>	<i>Optical Packet Switching, Conmutación Óptica de Paquetes</i>
<b>OXC</b>	<i>Optical Cross Connect, Conector Óptico Cruzado</i>
<b>QoS</b>	<i>Quality of Service, Calidad de Servicio</i>
<b>QoT</b>	<i>Quality of Transmission, Calidad de Transmisión</i>
<b>RBM</b>	<i>Restricted Boltzmann Machine, Máquina de Boltzmann Restringida</i>
<b>RF</b>	<i>Random Fit, Algoritmo de ajuste aleatorio</i>
<b>RL</b>	<i>Reinforcement Learning, Aprendizaje por Refuerzo</i>
<b>RNN</b>	<i>Recurrent Neural Network, Red Neuronal Recurrente</i>



**ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING**

---

<b>ROADM</b>	<i>Reconfigurable Optical Add-Drop Multiplexer</i> , Multiplexor Óptico Reconfigurable de Adición-Sustracción
<b>RSA</b>	<i>Routing and Spectrum Allocation</i> , Enrutamiento y Asignación de Espectro
<b>RWA</b>	<i>Routing and Wavelength Assignment</i> , Enrutamiento y Asignación de Longitud de Onda
<b>SBI</b>	<i>Southbound Interfaces</i> , Interfaces hacia el sur
<b>SCC</b>	<i>Spectrum Continuity Constraint</i> , Restricción de Continuidad de Espectro
<b>SCVPS</b>	<i>Spectrum-Constraint Path Vector Searching</i> , Algoritmo de búsqueda de vector de ruta de restricción de espectro
<b>SDEON</b>	<i>Software-Defined Elastic Optical Networking</i> , Red Óptica Elástica Definida por Software
<b>SDN</b>	<i>Software-Defined Networking</i> , Redes Definidas por Software
<b>SL</b>	<i>Supervised Learning</i> , Aprendizaje Supervisado
<b>SP</b>	<i>Shortest Path</i> , Algoritmo de camino más corto
<b>SSA</b>	<i>Semi-elastic Spectrum Allocation</i> , Asignación de espectro semi-elástico en el tiempo
<b>SSL</b>	<i>Semi-Supervised Learning</i> , Aprendizaje Semi-supervisado
<b>SW</b>	<i>Slot Width</i> , Ancho de Slot
<b>TN</b>	<i>Target Network</i> , Red de Destino
<b>UL</b>	<i>Unsupervised Learning</i> , Aprendizaje no Supervisado
<b>WAN</b>	<i>Wide Area Network</i> , Red de Área Amplia
<b>WDM</b>	<i>Wavelength Division Multiplexing</i> , Multiplexación por División de Longitud de Onda
<b>WRON</b>	<i>Wavelength-Routed Optical Network</i> , Red Óptica con Encaminamiento por Longitud de Onda



# 1.INTRODUCCIÓN

## 1.1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad, existe una tendencia a la integración y convergencia de las tecnologías y redes de telecomunicaciones existentes dentro una misma infraestructura de red de protocolo de internet (IP, *Internet Protocol*) orientada a paquetes. Dicha integración, trae consigo, un incremento exponencial en las demandas de tráfico<sup>1</sup> por parte de los usuarios y, en consecuencia, surge la evidente necesidad de una nueva red capaz de satisfacer estos nuevos requerimientos.

En este contexto, han surgido nuevas propuestas como es el caso de las redes ópticas elásticas (EON, *Elastic Optical Network*), un tipo de red creada con el objetivo de obtener un uso más eficiente de los recursos al adaptarse a los requerimientos del ancho de banda<sup>2</sup> de un enlace de red [1]. El diseño de estas nuevas redes está siendo orientado a sistemas centralizados, en donde se separa el plano de control del plano de datos con el fin de conseguir un control y gestión centralizados de la red, esto a partir de un único elemento controlador desde donde se obtiene una visión holística de toda la red. La integración de estas dos tecnologías es lo que se conoce como redes ópticas elásticas definidas por *software* (SDEON, *Software-Defined Elastic Optical Networking*) [2].

Uno de los problemas principales en las EON es el enrutamiento y la asignación de espectro de frecuencia (RSA, *Routing and Spectrum Allocation*), el cual involucra dos procedimientos. En primer lugar, se realiza el enrutamiento, en donde se debe seleccionar la ruta óptima entre una fuente y el destino, y, segundo, la asignación de espectro, definiendo el mejor conjunto de recursos espectrales disponibles de la ruta seleccionada [3] [4].

Actualmente, la influencia del aprendizaje automático (ML, *Machine Learning*) se ha incrementado significativamente dentro del campo de las redes ópticas llegando a convertirse en una herramienta prometedora para abordar varios de los problemas existentes. La gran cantidad de datos que se pueden manejar y los parámetros que requieren ser optimizados, son razones de peso para considerar su uso, puesto que no existen otras herramientas con resultados comparables [5]. El aprendizaje profundo (DL, *Deep Learning*) constituye una rama más avanzada del ML, y se caracteriza por estar construido a partir del principio de redes neuronales<sup>3</sup> y por no requerir reglas programadas previamente ni intervención humana constante, ya que el propio sistema es capaz de aprender por sí mismo mediante una fase de

---

<sup>1</sup> cantidad de datos que se mueven a través de una red en un momento dado.

<sup>2</sup> rango de frecuencias para transmitir datos a través de una conexión de dos puntos de una red en un tiempo específico.

<sup>3</sup> conjunto de unidades, llamadas neuronas artificiales, conectadas entre sí para transmitirse señales.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

entrenamiento previa, siendo de esta manera apropiado para el análisis predictivo y la toma de decisiones de forma automática [6].

Se plantea entonces, para este proyecto de grado, la propuesta de un algoritmo cognitivo<sup>4</sup> para la solución de RSA en una SDEON basado en DL. En aras de medir su desempeño, se realiza un análisis comparativo entre una SDEON con el algoritmo mencionado incluido y una SDEON con un algoritmo clásico basado en restricción de continuidad de espectro (SCC, *Spectrum Continuity Constraint*), todo esto mediante simulación.

Lo anterior conlleva a realizar la siguiente pregunta de investigación:

¿Cómo varía el desempeño de una SDEON al emplear un algoritmo cognitivo basado en DL y un algoritmo clásico basado en restricción de continuidad de espectro?

### 1.2. ESCENARIO DE MOTIVACIÓN

El constante despliegue de nuevos servicios y el rápido incremento en el tráfico y flujo de datos dentro de las redes, ha generado grandes exigencias dentro de la industria de telecomunicaciones promoviendo, por tal motivo, la búsqueda y desarrollo de nuevas soluciones dentro de este campo.

De esta manera, surgen nuevas tecnologías dentro de las redes ópticas como respuesta a estos nuevos requerimientos, como es el caso de las redes basadas en Multiplexación por División de Longitud de Onda (WDM, *Wavelength Division Multiplexing*), las cuales son redes de mayor capacidad, desarrolladas con el fin de permitir el transporte de señales de información a través de diferentes longitudes de onda<sup>5</sup> dentro de una misma fibra óptica<sup>6</sup>. En este tipo de redes el espectro de frecuencia es dividido en canales fijos de 50 o 100 GHz, lo que ocasiona que cuando se tienen conexiones con una baja tasa de datos<sup>7</sup> (menor a la capacidad del canal), el espectro disponible no es aprovechado en su totalidad.

Con el propósito de mejorar el uso del espectro dentro de las redes mencionadas, aparecen las EON, caracterizadas principalmente por dividir el espectro en canales más pequeños de 6.25 y 12.5 GHz y proporcionar un ancho de banda variable para cada conexión, permitiendo adaptarse con mayor flexibilidad a los requerimientos de nuevas conexiones y aprovechando de mejor manera los recursos disponibles. En la actualidad, existe una tendencia de estas redes hacia sistemas centralizados con unas características específicas, lo que se conoce como redes definidas por

---

<sup>4</sup> algoritmo que sigue los pasos del ciclo cognitivo.

<sup>5</sup> distancia física entre dos puntos a partir de los cuales una onda de luz se repite.

<sup>6</sup> tecnología usada para transmitir información en forma de pulsos de luz.

<sup>7</sup> velocidad de transferencia de datos a través de una red.



# ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

*software* (SDN, *Software-Defined Networking*), la combinación de SDN y EON es conocida en la actualidad como SDEON [7].

Han surgido diversas propuestas para dar solución a RSA, uno de los principales problemas que se encuentran dentro de las EON, sin embargo, el uso de algoritmos basados en DL para la resolución de esta problemática es un campo relativamente nuevo y no ha sido completamente explorado.

Por esta razón, en el presente proyecto de grado se propone un algoritmo cognitivo basado en DL como solución al problema de RSA en SDEON. Para la evaluación de este algoritmo, se consideran principalmente los parámetros de desempeño de probabilidad de bloqueo y el retardo extremo a extremo y se comparan los resultados al incluir en la red otro algoritmo clásico basado en SCC.

## 1.3. OBJETIVOS

### 1.3.1. OBJETIVO GENERAL

Analizar comparativamente el desempeño<sup>8</sup> de una SDEON para la solución del problema RSA al incluir un algoritmo clásico basado en restricción de continuidad de espectro y un algoritmo cognitivo basado en DL.

### 1.3.2. OBJETIVOS ESPECÍFICOS

- Diseñar e implementar un algoritmo cognitivo basado en DL para solucionar el problema de RSA en SDEON.
- Evaluar mediante simulación, el desempeño de la SDEON definida al incluir el algoritmo clásico basado en restricción de continuidad de espectro y el algoritmo cognitivo diseñado.

## 1.4. ENFOQUE

Para el desarrollo del presente trabajo de grado, se realizó una amplia investigación bibliográfica en donde se recopilaron trabajos relacionados con las principales temáticas a tratar: RSA, OBS, DL y SDEON.

Posteriormente, se realizó una caracterización de una red OBS/SDEON con ayuda de una herramienta de simulación, utilizando una topología<sup>9</sup> irregular de red de la Fundación Nacional de Ciencias (NSFNeT, *National Science Foundation Network*), una red de área amplia (WAN, *Wide Area Network*) de Estados Unidos de América

---

<sup>8</sup> desempeño medido en términos de la probabilidad de bloqueo y el retardo extremo a extremo de la red.

<sup>9</sup> disposición de una red incluyendo sus nodos y líneas de conexión



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

la cual consta de 14 nodos y 21 enlaces bidireccionales utilizada ampliamente en trabajos de investigación.

A continuación, se realizó el diseño e implementación del algoritmo clásico basado en SCC y del algoritmo cognitivo basado en DL. Se incluyó cada uno de ellos por separado en la red NSFNeT OBS/SDEON caracterizada con el fin de realizar las pruebas pertinentes y evaluar y comparar el desempeño de cada uno.

Finalmente, a partir de los resultados obtenidos de las simulaciones y considerando los parámetros de desempeño de probabilidad de bloqueo y retardo extremo a extremo, se realizó un análisis comparativo de desempeño de la red con cada uno de los algoritmos independientemente. La Figura 1.1, muestra un diagrama de flujo en donde se describe de manera general el proceso seguido para cumplir el objetivo general del presente trabajo de grado.



# ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

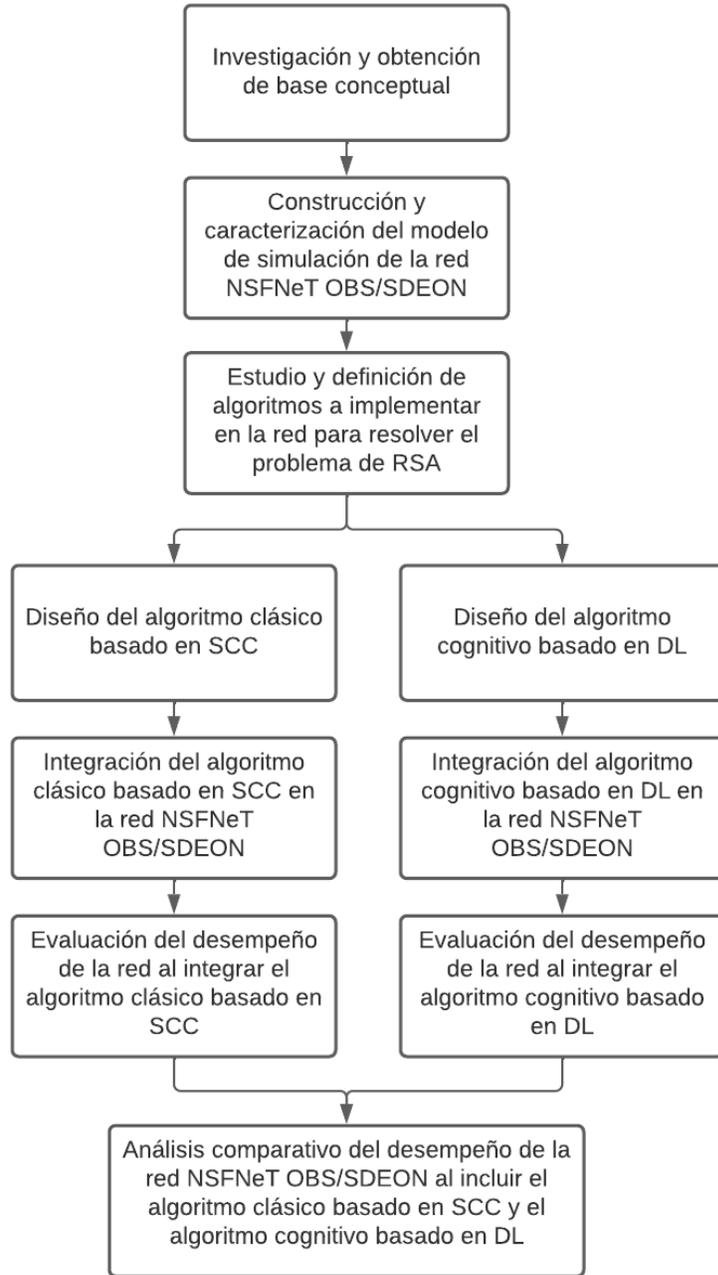


Figura 1.1 Diagrama de flujo del enfoque del trabajo de grado.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

### 1.5. ALCANCE

En el presente documento se describe el proceso para el diseño y caracterización del modelo de la red OBS/SDEON usando la topología NSFNeT en la herramienta de simulación seleccionada, y también, el diseño de cada uno de los dos algoritmos.

Para realizar la evaluación del desempeño de cada uno de los algoritmos definidos en los objetivos del trabajo de grado, se plantean principalmente dos escenarios. El primero consta de la red NSFNeT OBS/SDEON en donde se incluye el algoritmo clásico basado en SCC. En el segundo se tiene la misma red NSFNeT OBS/SDEON, incluyendo en este caso, el algoritmo cognitivo basado en DL. Para ambos escenarios, con el fin de observar el comportamiento y desempeño de la red con cada algoritmo, se utiliza velocidades de transmisión de enlace de 1 y 2,5 Gbps y anchos de espectro de 100 y 200 GHz, realizando variaciones de tráfico y de longitud de paquete en la red para cada configuración.

### 1.6. APORTES

El desarrollo de este trabajo de grado contribuye a la línea investigativa del departamento de Telecomunicaciones de la Universidad del Cauca, en el área específica de sistemas de comunicaciones ópticas. Los aportes del presente trabajo de grado son:

- Caracterización de una red OBS/SDEON sobre la topología NSFNeT en la herramienta de simulación IDE OMNeT++.
- Diseño e implementación de un algoritmo clásico basado en SCC.
- Diseño e implementación de un algoritmo cognitivo basado en DL.
- Análisis comparativo del desempeño de una red NSFNeT OBS/SDEON al incluir un algoritmo clásico basado en SCC y un algoritmo cognitivo basado en DL.

### 1.7. ORGANIZACIÓN DEL TRABAJO DE GRADO

En el Capítulo 2 y 3, se describen los conceptos básicos relacionados con EON, SDN, y DL. Asimismo, se incluye definiciones sobre redes ópticas, técnicas de multiplexación y técnicas de conmutación ópticas y se profundiza en algunos conceptos importantes como la técnica de conmutación óptica de ráfagas (OBS, *Optical Burst Switching*) y sus características. Se incluyen también, conceptos sobre RSA y los tipos de algoritmos RSA encontrados en la literatura. Esto con el fin de formar una base de conocimiento sólida para el desarrollo del trabajo de grado.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

El Capítulo 4, trata sobre las metodologías aplicadas para el desarrollo del trabajo de grado, el diseño e implementación del modelo de la red NSFNeT OBS/SDEON en el entorno de desarrollo integrado (IDE, *Integrated Development Environment*) de OMNeT++, describiendo en detalle cada uno de los módulos que la conforman, y también, la descripción de los algoritmos diseñados.

En el Capítulo 5, se presentan los escenarios de simulación y pruebas, para la evaluación y análisis de desempeño de la red NSFNeT OBS/SDEON al incluir los algoritmos diseñados. Las pruebas fueron realizadas en el entorno de simulación IDE OMNeT++.

Finalmente, el Capítulo 6 contiene las conclusiones, recomendaciones y propuestas sobre posibles trabajos futuros relacionados con este trabajo de grado.



## 2. GENERALIDADES DE REDES ÓPTICAS ELÁSTICAS DEFINIDAS POR SOFTWARE Y *DEEP LEARNING*

Desde la antigüedad, las personas han considerado el uso de métodos ópticos para comunicarse a largas distancias. En los años 1000 a.C., los griegos y romanos utilizaban la transmisión óptica empleando señales de humo y fuego para enviar mensajes, anuncios o llamadas de ayuda. Las mejoras a lo largo del tiempo de estos sistemas se deben gracias a los grandes avances de la tecnología, iniciando por la invención del láser en la década de los 60 [8].

Hoy en día, se cuenta con numerosas tecnologías, metodologías y técnicas que han hecho posible las comunicaciones a través de fibra óptica y que han permitido incrementar enormemente las capacidades de los sistemas de comunicaciones actuales siendo capaces de afrontar los nuevos requerimientos y las grandes demandas, dichos sistemas son conocidos como redes ópticas.

### 2.1. REDES ÓPTICAS ELÁSTICAS DEFINIDAS POR SOFTWARE

Las SDEON se constituyen por tres conceptos principales los cuales son tratados en esta sección: redes ópticas, redes ópticas elásticas (EON) y redes definidas por *software* (SDN).

#### 2.1.1. REDES ÓPTICAS

Se describen a continuación, algunos conceptos de importancia dentro de las redes ópticas como lo son las técnicas de multiplexación y las técnicas de conmutación.

##### 2.1.1.1. TÉCNICAS DE MULTIPLEXACIÓN

Mediante la multiplexación es posible la transmisión de información de múltiples usuarios a través de un solo canal, permitiendo incrementar la utilización de éste y la capacidad de transmisión. Ésta se compone principalmente de dos dispositivos, el multiplexor que se encarga de combinar las diferentes señales en una sola y el demultiplexor el cual obtiene nuevamente cada una de las señales por separado.

El constante mejoramiento de los componentes ópticos y su mayor confiabilidad y precisión, han permitido que sea posible el empleo de casi todas las técnicas de



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

multiplexación existentes en sistemas inalámbricos dentro de los sistemas ópticos. Se discuten a continuación, algunas de las técnicas más empleadas [9].

- **MULTIPLEXACIÓN POR DIVISIÓN DE TIEMPO (TDM)**

Probablemente, el esquema más usado en sistemas eléctricos e inalámbricos no tiene un uso tan amplio en redes ópticas, debido a que la fibra óptica ya cuenta con un gran ancho de banda disponible, sin embargo, aún es usado en algunas aplicaciones. Este tipo de multiplexación se realiza por medio del intercalado de pulsos que permite que diferentes flujos de información óptica usen y compartan la capacidad total de la fibra durante un tiempo limitado, mejorando la velocidad de datos general de la fibra [9].

- **MULTIPLEXACIÓN POR DIVISIÓN DE ESPACIO (SDM)**

La idea básica en la multiplexación por división de espacio es la separación de diferentes canales en el espacio. Esto puede realizarse de diferentes maneras, siendo la más sencilla el uso de múltiples fibras. Sin embargo, esto requiere que la infraestructura del canal sea duplicada para cada fibra lo que incrementa significativamente los costos de implementación. Existen diferentes técnicas que posibilitan la reducción de estos costos, como el uso de un láser de alta potencia con divisores para cada fibra, el uso de múltiples fibras para formar un solo cable, el uso de una matriz emisora de láser en la fuente, el uso de fibras multinúcleo, entre otros [9].

- **MULTIPLEXACIÓN POR DIVISIÓN DE LONGITUD DE ONDA (WDM)**

Es la técnica más conocida y usada en la actualidad dentro de las redes ópticas. Consiste en múltiples longitudes de onda donde cada una se encuentra codificada por separado con información, éstas se multiplexan en una sola fibra monomodo, se transmiten a cierta distancia y finalmente se demultiplexan en longitudes de onda en canales separados cuya información se detecta y se recibe. El uso de esta técnica permite un aumento considerable en la capacidad de una fibra óptica comparado con los enlaces originales punto a punto desplegados en la década de los 80 que transportaban únicamente una longitud de onda [8].



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

La implementación de estas redes WDM sofisticadas requiere una variedad de dispositivos para realizar funciones de combinación, distribución, aislamiento y amplificación de la señal óptica en diferentes longitudes de onda.

La Figura 2.1 muestra un esquema de un sistema de comunicaciones WDM. Se tienen múltiples transmisores, cada uno funcionando en su propia longitud de onda dedicada. Estos flujos de datos individuales se multiplexan y se transmiten a través de una misma fibra óptica, posteriormente, se demultiplexan y llegan a múltiples receptores.

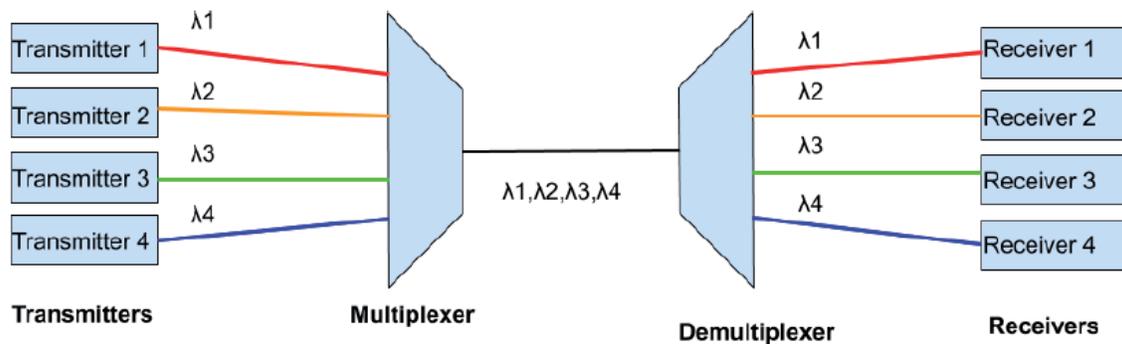


Figura 2.1. Esquema general de un sistema WDM. Tomada de [9]

Un enlace típico WDM consta de transmisores, multiplexores de agregación y eliminación (OADM, *Optical Add-Drop Multiplexer*) y detectores para el enlace de comunicación. También, puede incluir, de acuerdo con los requerimientos, pre-amplificadores, amplificadores de línea y post-amplificadores [9].

### 2.1.1.2. TÉCNICAS DE CONMUTACIÓN

La conmutación se define como el proceso que permite el establecimiento de una ruta desde un nodo de origen hasta un nodo de destino dentro de una red haciendo un uso eficiente de los recursos disponibles. A nivel eléctrico, esto representa mayores retardos debido al procesamiento requerido para realizar el envío de la información. Es por esta razón, que surge la necesidad de llevar este proceso al dominio óptico con el objetivo de reducir estos retardos y obtener un diseño de red totalmente óptico [10].

A continuación, se describen las técnicas de conmutación óptica más empleadas dentro de las redes ópticas, a saber, la Conmutación Óptica de Circuitos (OCS, *Optical Circuit Switching*), la Conmutación Óptica de Paquetes (OPS, *Optical Packet*



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

*Switching*) y la Conmutación Óptica de Ráfagas (OBS), siendo esta última la empleada en este trabajo de grado.

### • CONMUTACIÓN ÓPTICA DE CIRCUITOS (OCS)

En este tipo de conmutación orientada a conexión, se establecen canales de comunicación temporales entre los nodos extremos antes de iniciar la transmisión, asignando una longitud de onda a cada conexión entre el nodo origen y destino. Una vez se establece el camino, se transmite la información sin la necesidad de almacenar y reenviar paquetes en cada nodo de la ruta. No se requiere conversión óptico-eléctrica por lo que se tienen altas velocidades de transmisión. Es posible también, el uso de diferentes longitudes de onda durante el trayecto mediante matrices de conmutación (OXC, *Optical Cross Connect*).

Esta técnica es apropiada cuando se tiene tráfico constante y se requiere gran ancho de banda, ya que el hecho de requerir un establecimiento previo de la conexión puede generar retardos excesivos si se envían paquetes de pequeño tamaño [11].

### • CONMUTACIÓN ÓPTICA DE PAQUETES (OPS)

Consiste en el enrutamiento y transmisión de los datos de usuario a través de la red en forma de paquetes ópticos. Estos datos llevan consigo información de control la cual está encapsulada en un encabezado con un formato especial. En un sistema OPS, las funciones de enrutamiento y procesamiento del encabezado se realizan de forma electrónica, mientras que la conmutación de la carga útil<sup>10</sup> se realiza en el dominio óptico para cada uno de los paquetes.

La principal limitación en este tipo de conmutación radica en la necesidad de uso de búferes para el almacenamiento temporal de los paquetes ópticos mientras estos establecen una ruta de salida en los conmutadores intermedios. Esta limitación tecnológica es eliminada mediante el empleo de la conmutación óptica de ráfagas descrita en la siguiente sección [8].

---

<sup>10</sup> conjunto de datos de usuario que se transmiten, excluyendo información de control.



## • CONMUTACIÓN ÓPTICA DE RÁFAGAS (OBS)

En un sistema OBS los paquetes se ensamblan dentro de unidades de datos más grandes las cuales son conocidas como ráfagas de datos. Para cada una de estas ráfagas se cuenta con un paquete de cabecera creado con el fin de llevar a cabo las funciones de control. El ensamble de la ráfaga y la generación del paquete de cabecera tienen lugar en los nodos de frontera de la red OBS, los cuales son nodos que están en los extremos de la red y pueden comunicarse e interactuar con redes de otro tipo. La agrupación de los paquetes se realiza mediante ciertos criterios como la dirección de destino, la calidad de servicio<sup>11</sup> (QoS, *Quality of Service*), entre otros. La ráfaga de datos y su correspondiente paquete de cabecera son enrutados a través de los nodos centrales de la red por diferentes canales. La ráfaga es enviada desde el nodo frontera a través de un canal de datos, mientras que el paquete de cabecera se envía a través de un canal de control, llegando ambos a los mismos nodos. De acuerdo con el protocolo implementado, existe un tiempo de diferencia entre el envío del paquete de control, que se realiza en primer lugar, y el envío de la ráfaga de datos [12].

Este tipo de conmutación será empleada en este trabajo de grado en el diseño de la red y se describe con mayor detalle en el siguiente capítulo.

### 2.1.2. REDES ÓPTICAS ELÁSTICAS (EON)

Hoy en día, la infraestructura de las redes de telecomunicaciones afronta grandes desafíos debido al constante crecimiento del volumen de tráfico en las redes y al despliegue de nuevos servicios como computación en la nube<sup>12</sup>, video bajo demanda, transmisión de video en alta definición, entre otros. De acuerdo con datos estadísticos, se espera que para el 2023 el 66% de la población mundial esté usando Internet comparado con el 51% que se tenía en el 2018, evidenciándose un notable aumento del 15% en un periodo de cinco años [13].

Por tal motivo, han surgido nuevos paradigmas y tecnologías con el objetivo de satisfacer estas necesidades. Es así, como aparece el concepto de las redes ópticas elásticas o EON [14], mediante las cuales se mejora considerablemente la eficiencia y desempeño de las redes actuales.

---

<sup>11</sup> rendimiento promedio de una red desde el punto de vista de los usuarios de esa red

<sup>12</sup> uso de una red de servidores remotos conectados a internet para almacenar, administrar y procesar datos, servidores, bases de datos, redes y software.



### 2.1.2.1. DEFINICIÓN DE EON

Una EON es un tipo de red capaz de adaptarse a los requerimientos de las conexiones ópticas con el propósito de optimizar la eficiencia del uso del espectro disponible. Estas redes nacen como una evolución de los sistemas ópticos WDM actuales. En un sistema WDM se realiza la transmisión de múltiples canales de información a través de una misma fibra, teniendo cada canal una longitud de onda determinada. Como componente dentro de estos sistemas, se tiene un multiplexor reconfigurable óptico (ROADM, *Reconfigurable Optical Add and Drop Multiplexer*) que es el dispositivo encargado de unir las señales ópticas provenientes de diferentes transmisores y enviarlas a través de un hilo de fibra. Este dispositivo también se encarga de las funciones de demultiplexación. La inserción o extracción de longitudes de onda se realiza por medio de un conmutador interno de selección de longitud de onda (WSS, *Wavelength Selective Switch*) [15].

Estos sistemas operan mediante rejillas fijas, de acuerdo al estándar G.694.1 de la Unión Internacional de Telecomunicaciones (ITU, *International Telecommunication Union*), con una separación entre canales desde 25 GHz hasta 100 GHz y con bandas de guarda para evitar interferencia con canales adyacentes, lo que hace que no se tenga un uso eficiente del espectro principalmente cuando se tienen conexiones con baja tasa de datos.

En contraste, las redes EON trabajan empleando una rejilla flexible o *Flexgrid*, proporcionando un ancho de banda variable para cada conexión. Estas redes cuentan con una granularidad más fina que en WDM donde cada canal se divide por ranuras o slots de frecuencia con anchos de 6.25 o 12.5 GHz, de esta manera se logra mejorar el uso del espectro y maximizar el *throughput*<sup>13</sup> de la red.

En la Figura 2.2 (a) y (b), se observa un ejemplo en donde se puede evidenciar las principales diferencias en el uso del espectro por parte de los dos sistemas. WDM asigna un canal de longitud fija para cada una de las conexiones de 10, 40 y 100 Gbps, mientras que EON se adapta a los requerimientos de cada conexión obteniendo al final un ahorro significativo de espectro y sin utilizar bandas de guarda innecesarias [16].

---

<sup>13</sup> capacidad de un enlace de transportar información útil, representa la cantidad de información válida que puede transmitirse por unidad de tiempo.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

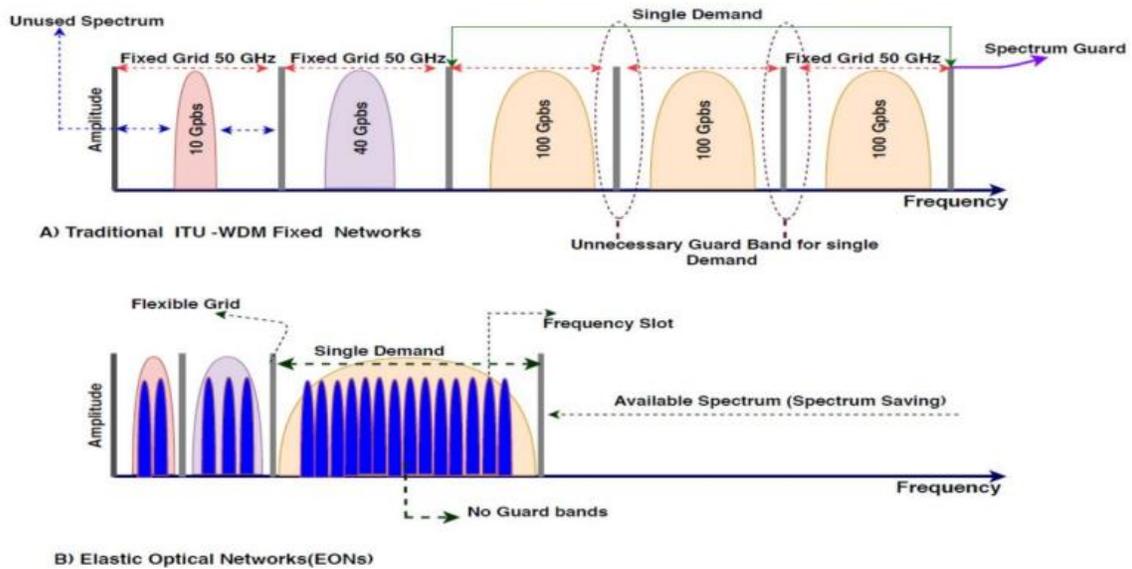


Figura 2.2. Uso del espectro a) WDM b) EON. Tomada de [16]

### 2.1.2.2. CARACTERÍSTICAS DE EON

En las redes EON, la frecuencia central de cada canal tiene una granularidad más fina lo que permite la expansión y contracción de los caminos ópticos. Por esta razón, estas redes proporcionan varias características únicas descritas a continuación.

- **Ancho de Banda Flexible:**

Las redes ópticas tradicionales requieren de una asignación completa de la capacidad de la longitud de onda para una ruta óptica. En cambio, en las redes ópticas elásticas, se realizan operaciones de segmentación y agregación de ancho de banda. La segmentación consiste en asignar el ancho de banda suficiente (sublongitud de onda) cuando se tienen conexiones que solo requieren una parte de este. Por otra parte, la agregación permite crear una ruta óptica o súper longitud de onda combinada, garantizando una alta utilización de los recursos espectrales disponibles.

Un ejemplo de aplicación de estas propiedades se observa en la Figura 2.3, donde un ancho de banda de 40 Gbps es segmentado en tres sublongitudes de onda de 5, 15 y 20 Gbps respectivamente. En cuanto a la agregación, se observa como tres anchos de banda de 40 Gbps son multiplexados proporcionando un supercanal de 120 Gbps [16].



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

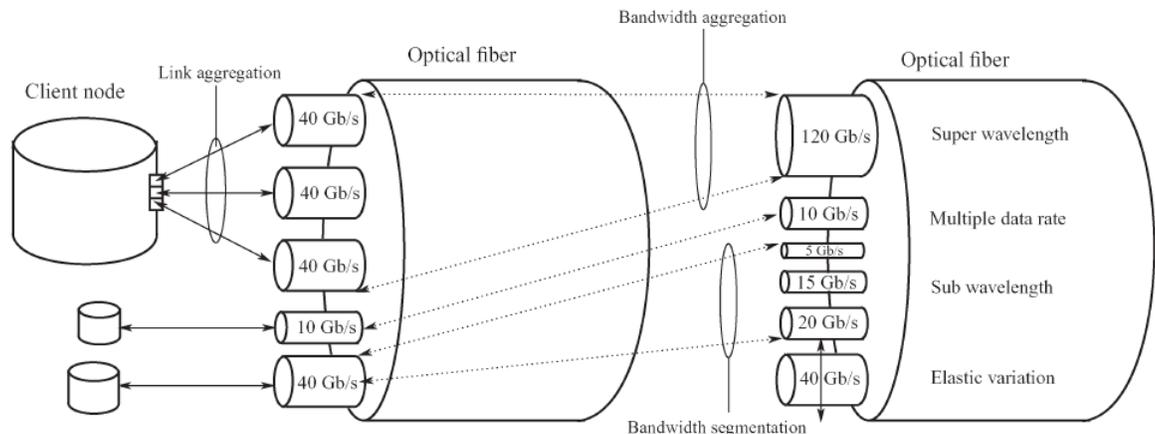


Figura 2.3. Propiedades de segmentación y agregación en EON. Tomada de [17].

- **Acomodación múltiple de tasa de datos:**

Estas redes tienen la capacidad de soportar tasas de datos múltiples y proporcionar alta eficiencia en el uso de espectro a través de la asignación flexible de este. Las redes ópticas tradicionales que operan con rejillas fijas ocasionan un malgasto del ancho de banda debido a la excesiva separación de frecuencias cuando se tienen señales con baja tasa de bits [16] [17].

- **Virtualización de red:**

Permite la visualización de la red óptica mediante enlaces virtuales soportados por las subportadoras de multiplexación por división de frecuencias ortogonales (OFDM, *Orthogonal Frequency-Division Multiplexing*) [17].

- **Ahorro de energía**

Con el objetivo de ahorrar el consumo de potencia, estas redes tienen una propiedad única que consta en la desactivación de las señales subportadoras que no están siendo usadas mientras la demanda de tráfico en la red es baja [17].

### 2.1.2.3. TECNOLOGÍAS PARA EON

- **MULTIPLEXACIÓN POR DIVISIÓN DE FRECUENCIAS ORTOGONALES (OFDM)**



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

Es un tipo de multiplexación prometedora para transmisiones de alta velocidad en redes ópticas. Es un tipo especial de esquema de modulación<sup>14</sup> multiportadora el cual transmite señales a alta velocidad divididas en un número de canales ortogonales o subportadoras de menor capacidad. Gracias a esta división es posible el ajuste del ancho de banda de la señal a transmitir, seleccionando solamente las subportadoras que generan dicha señal [15].

La propiedad de ortogonalidad permite que las señales subportadoras puedan concentrarse y traslaparse en un ancho espectral menor que en WDM, eliminando de esta manera las bandas de guarda entre éstas [18].

- **NYQUIST WDM**

Otra alternativa para la transmisión de canales es conocida como Nyquist WDM (N-WDM, *Nyquist Wavelength Division Multiplexing*) la cual se basa en el uso de pulsos ópticos a partir de un espectro rectangular y con un ancho de banda igual a la tasa de símbolos. Esta técnica ha dado resultados similares sin requerir tanta complejidad en los equipos como en el caso de OFDM, el cual emplea receptores de mayor ancho de banda y conversores análogo-digital de mayor velocidad.

Sin embargo, esta técnica no es muy viable para redes ópticas elásticas ya que para su implementación requiere una señal completamente rectangular en el dominio de la frecuencia [15].

### 2.1.2.4. ARQUITECTURA DE EON

Para satisfacer las crecientes demandas de ancho de banda, se propone una arquitectura para redes ópticas elásticas la cual consiste principalmente de dos elementos básicos: un transpondedor de ancho de banda variable (BVT, *Bandwidth Variable Transponder*) y un conmutador cruzado de ancho de banda variable (BV-WXC, *Bandwidth Variable-Wavelength Cross Connect*). Se describen a continuación en más detalle estos elementos [16].

---

<sup>14</sup> proceso mediante el que modifica un parámetro de una señal portadora para que lleve información.



- **TRANSPONDEDOR DE ANCHO DE BANDA VARIABLE (BVT)**

Este elemento es usado para sintonizar el ancho de banda ajustando la tasa de bits de transmisión o el formato de modulación. Está diseñado para operar con transmisiones de alta velocidad usando formatos complejos de modulación espectralmente eficientes [16].

- **CONMUTADOR CRUZADO DE ANCHO DE BANDA VARIABLE (BV-WXC)**

El conmutador cruzado de ancho de banda variable se encarga de asignar una conexión cruzada con un ancho de banda apropiado para dar soporte a un camino de luz óptico elástico. Para esto, es necesario configurar su ventana de conmutación de forma flexible de acuerdo con el ancho espectral de la señal óptica de entrada [16].

### **2.1.2.5. ENRUTAMIENTO Y ASIGNACIÓN DE ESPECTRO EN EON**

El enrutamiento y asignación de espectro es considerado una de las funcionalidades claves y a su vez uno de los principales desafíos en las redes ópticas elásticas. En los sistemas WDM, cuando se desea proporcionar un camino de luz para una nueva conexión entre dos nodos de una red es necesario establecer una ruta apropiada y asignar la longitud de onda requerida, este problema es conocido como enrutamiento y asignación de longitud de onda (RWA, *Routing and Wavelength Assignment*) [19].

El problema de RSA en las redes ópticas elásticas es equivalente al problema de RWA en las redes ópticas basadas en WDM. La principal diferencia entre estos dos problemas es que en RSA se asigna un conjunto de *slots* contiguos de frecuencia a una conexión, en lugar de asignar una longitud de onda como sucede en RWA. Se hablará más a detalle de este concepto en el siguiente capítulo [16].

### **2.1.3. REDES DEFINIDAS POR SOFTWARE**

Con la aparición constante de nuevos servicios de red y el creciente tráfico de todo tipo de información circulando por las redes, se convierte en un tema esencial la inteligencia de éstas, en donde se debe satisfacer las nuevas demandas de ancho de banda, transmisión de alta velocidad a largas distancias y las grandes capacidades requeridas [20].



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

Es por esta razón, que desde hace un tiempo se reconoció la necesidad de desarrollar nuevas capas de abstracción en la red para las funciones de control y gestión de ésta. De aquí nace el concepto de redes definidas por *software* o SDN que será discutido en esta sección [21].

### 2.1.3.1. DEFINICIÓN DE SDN

Las SDN es el principio de diseño que surgió con el fin de estructurar el desarrollo de las nuevas capas de abstracción de red. La principal idea detrás de este nuevo paradigma es proporcionar un nivel más alto de gestión de red con un dispositivo central programable que controla los nodos de la red [22].

Tal y como es definido por el grupo de trabajo de ingeniería de Internet (IETF, *Internet Engineering Task Force*), SDN abstrae y separa el plano de datos del plano de control, lo que permite un desarrollo tecnológico más rápido en estos dos planos.

Dentro de las redes ópticas, este principio ofrece una programación y control unificados para diferentes tipos de recursos de capa óptica de acuerdo con los requisitos de los usuarios u operadores. A partir de *software* programable y personalización dinámica, se resuelve el problema de extensión de funciones, al lograr una respuesta más rápida a las peticiones, un uso eficiente de los recursos y un aprovisionamiento de servicios flexible [20].

### 2.1.3.2. PRINCIPIOS DE SDN

SDN es definida como una arquitectura de red con cuatro principios fundamentales:

- Los planos de control y datos están desacoplados. La funcionalidad de control es eliminada de los dispositivos de red, los cuales se convierten en dispositivos simples para el reenvío de paquetes.
- La toma de decisiones para el envío de paquetes se basa principalmente en el flujo y no en el destino. Un flujo se refiere a una secuencia de paquetes entre un origen y un destino, estos paquetes tienen políticas de servicio idénticas.
- La lógica de control es trasladada a una entidad externa, conocida como controlador SDN o sistema operativo de red (NOS, *Network Operating System*). El NOS es una plataforma *software* que proporciona los recursos para la programación de los dispositivos de reenvío basados en un sistema centralizado.
- Este tipo de red es programable mediante aplicaciones de *software* que son ejecutadas en NOS, el cual se comunica con los dispositivos del plano de datos. Esta es considerada la característica fundamental de SDN [23].



### 2.1.3.3. ARQUITECTURA DE SDN

Fundamentalmente, SDN realiza la separación de los planos de control y datos de la red. El plano de control está supervisado por una entidad centralizada denominada controlador SDN, por su parte, el plano de datos se compone de dispositivos de transmisión simplificados denominados conmutadores SDN. Este enfoque permite un despliegue de políticas de enrutamiento de alcance global en la red, a diferencia del enfoque poco eficiente de salto a salto que se tiene en redes distribuidas.

En el plano de control, el dispositivo de control se encarga de decidir a qué interfaz se deben enviar los paquetes en el plano de datos. Este concepto es especialmente útil en redes multicapa ayudando a mejorar la eficiencia del enrutamiento y la asignación de espectro en el caso de las redes ópticas elásticas. La arquitectura de SDN requiere un canal de comunicación entre los planos, para este propósito se ha diseñado el protocolo OpenFlow (OFP, *OpenFlow Protocol*). A través de este, el controlador puede enviar diferente tipo de información, como, por ejemplo, las entradas a ser colocadas en las tablas de los conmutadores para la implementación de políticas de enrutamiento. Estas políticas, gracias a la naturaleza centralizada de SDN, pueden ser gestionadas por un administrador de red o programadas de forma automática por aplicaciones externas que se integran al controlador [22].

La arquitectura de SDN se compone generalmente de tres capas:

- **Capa de Aplicación:** Está formada por programas encargados de enviar el comportamiento y los requisitos deseados de la red al controlador SDN. Las interfaces de comunicación entre las capas de aplicación y control son conocidas como interfaces hacia el norte (NBI, *Northbound Interfaces*). A partir de la recopilación de información del controlador, las aplicaciones pueden obtener una visión de la red que facilita la toma de decisiones.
- **Capa de Control:** Esta capa está constituida por el controlador SDN donde se encuentra la inteligencia de la red. Este controlador es una entidad lógica con una visión global de la red que se encarga de recibir instrucciones desde la capa de aplicación y controlar los dispositivos de la red para obtener los resultados deseados. Asimismo, este dispositivo extrae información desde los dispositivos sobre la red y el tráfico actual en esta y la envía a la capa de aplicación.
- **Capa de Infraestructura:** Se compone de dispositivos programables los cuales son comúnmente llamados elementos o dispositivos de red. Cuenta con conmutadores SDN los cuales tienen la función de consultar al controlador cada decisión que deben tomar en un momento determinado. Cuando llega un



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

nuevo flujo de datos, el conmutador le solicita al controlador realiza la acción correspondiente y posteriormente almacena esta información en sus tablas para futuros paquetes entrantes [22].

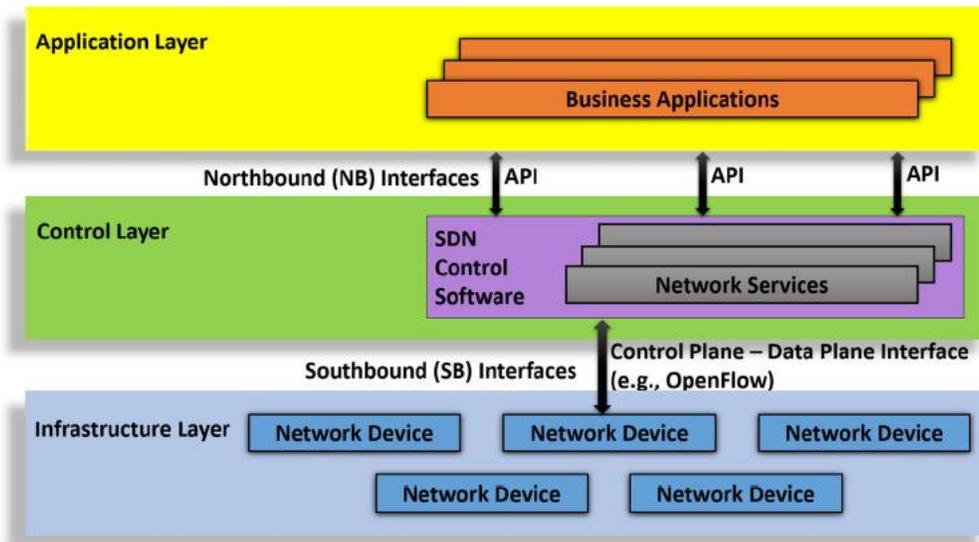


Figura 2.4. Arquitectura de SDN. Tomada de [2]

En la Figura 2.4, se muestran los aspectos generales de la arquitectura de SDN. Las interfaces de comunicación entre capas normalmente se denominan interfaces hacia el norte y hacia el sur o interfaces *Northbound* y *Southbound*. Una NBI es mediante la cual se comunican el controlador y las aplicaciones, mientras que una interfaz *Southbound* (SBI, *Southbound Interfaces*) hace referencia a la conexión entre el controlador y el hardware de la red.

### 2.1.3.4. SDON

Las SDN y la tecnología de red óptica flexible son dos conceptos clave que permiten a los operadores de red personalizar la infraestructura de acuerdo con los requerimientos de la aplicación, minimizando así, los costos de operación y de capital necesarios para alojar nuevas aplicaciones y servicios [24].

Para cumplir con las tendencias y aplicaciones actuales se debe lograr que las redes sean rápidas, flexibles y adaptables a las demandas de los usuarios. Esto se puede conseguir fusionando los beneficios de SDN y las redes ópticas, llevando así, a que estas redes sean programables y los suficientemente flexibles para satisfacer los requerimientos de los usuarios.

En la Figura 2.5, se representa la arquitectura de una SDON en donde se tiene una comunicación entre dos clientes A y B. Estos clientes se encuentran conectados a

conmutadores *OPF*, los cuales a su vez son controlados por el controlador SDN el cual realiza el reenvío de la comunicación. En cuanto a la parte óptica, se observan algunos conmutadores ROADM. La comunicación entre los conmutadores *OPF* y ROADM se hace a través de conversores eléctrico-ópticos [25].

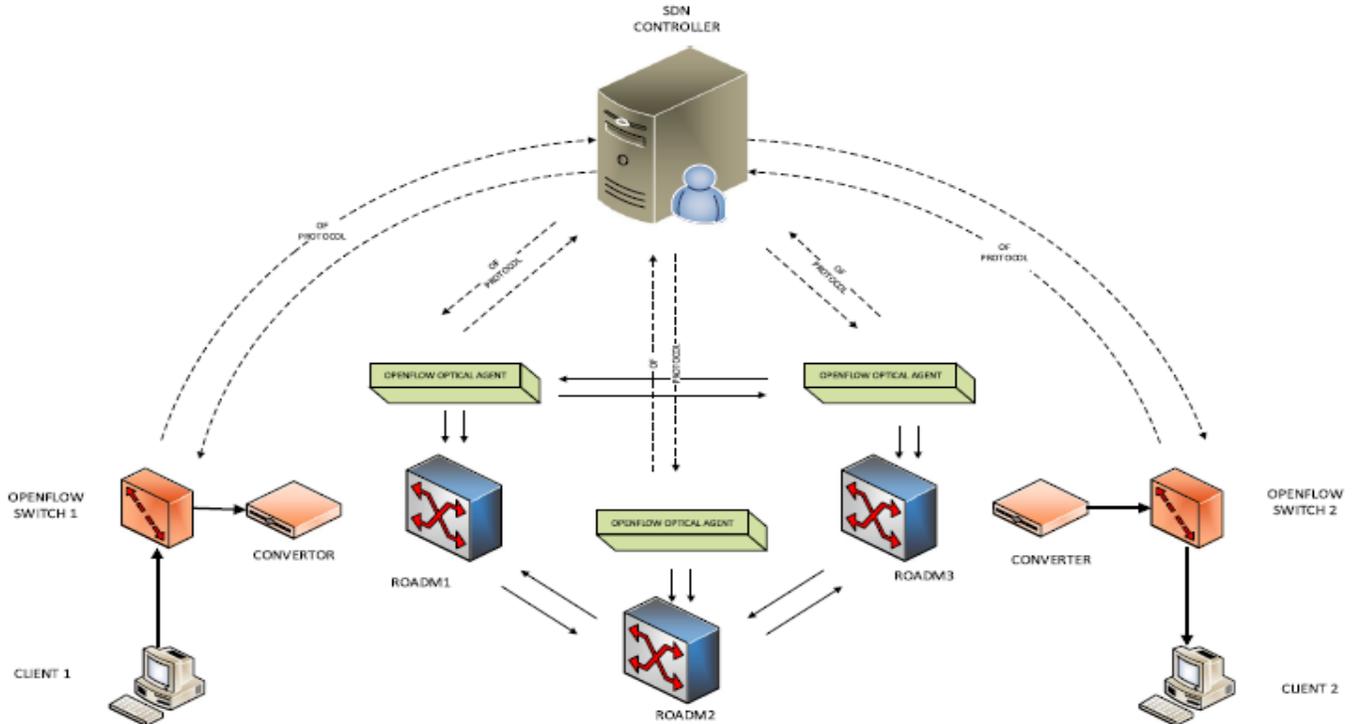


Figura 2.5. Arquitectura general de SDON. Tomada de [25]

## 2.2. GENERALIDADES DE *DEEP LEARNING*

Los sistemas de inteligencia artificial o IA (*AI*, *Artificial Intelligence*) tienen la capacidad de realizar funciones análogas al aprendizaje y la toma de decisiones imitando procesos biológicos, especialmente los procesos cognitivos de los seres humanos. Generalmente, la IA no ofrece sistemas completamente autónomos, sino que, mediante ésta, es posible agregar conocimiento y razonamiento a una gran cantidad de aplicaciones o entornos con el fin de hacerlos más inteligentes e intuitivos frente a cambios.

La mejora del rendimiento de las redes de telecomunicaciones a partir de la aplicación de técnicas basadas en IA ha sido un área de amplia investigación en los últimos tiempos, enfocándose principalmente en aspectos de transmisión, conmutación y gestión [26]. La tendencia actual está llevando a la introducción de



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

este tipo de técnicas en redes ópticas para superar las limitaciones presentes y abordar futuros problemas.

Los métodos basados en ML están siendo ampliamente usados en diferentes aplicaciones en redes ópticas. Estos métodos han demostrado una capacidad notable para analizar grandes cantidades de datos y aprender a partir de los entornos. A continuación, se discute más a detalle este aspecto.

### **2.2.1. MACHINE LEARNING**

El ML, una rama de la IA, es adecuado para hacer frente a los problemas complejos en donde es difícil dar una respuesta en tiempos razonables, teniendo una capacidad única para aprender el comportamiento de un sistema a partir de datos previos. El desarrollo de nuevas tecnologías de red como SDN y la virtualización de las funciones de red (NFV, *Network Function Virtualization*) hace que la aplicación de ML en redes ópticas sea más eficiente, siendo ampliamente usado en aspectos de gestión y control en redes ópticas como RSA, gestión de fallas, gestión de tráfico, predicción de calidad de transmisión (QoT, *Quality of Transmission*), entre otros [27].

Los algoritmos de ML son clasificados generalmente en cuatro categorías: aprendizaje supervisado (SL, *Supervised Learning*), aprendizaje no supervisado (UL, *Unsupervised Learning*), aprendizaje semi-supervisado (SSL, *Semi-Supervised Learning*) y aprendizaje por refuerzo (RL, *Reinforcement Learning*).

#### **2.2.1.1. APRENDIZAJE SUPERVISADO**

Las técnicas de aprendizaje supervisado reciben un conjunto de datos o *dataset* con vectores de las características de entrada y la salida que se espera para cada muestra, aprendiendo de dichos datos. La función puede generar valores continuos prediciendo un valor numérico al partir de la entrada, conocido esto como regresión, o una etiqueta a partir de un conjunto especificando la clase de la entrada dada, conocido como clasificación. Algunas de las técnicas aplicadas en el aprendizaje supervisado son regresión lineal, árboles de decisión, bosque aleatorio, regresión logística, k-vecinos más cercanos, máquinas de soporte vectorial, Naive Bayes, redes neuronales, entre otros [28].

#### **2.2.1.2. APRENDIZAJE NO SUPERVISADO**

A diferencia del aprendizaje supervisado, los algoritmos de aprendizaje no supervisado no reciben muestras de etiquetas o del resultado deseado del conjunto de datos. Estos esquemas deben encontrar la estructura de entrada a partir de la comprensión de los datos y la identificación de patrones, por lo tanto, no predicen



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

valores de salida o clase de una entrada. Estos mecanismos son útiles para agrupar datos con características similares, o en el caso de las redes, para agrupar flujos de tráfico similares para proporcionar diferente calidad de servicio de acuerdo con los recursos disponibles. Algunas de las técnicas más usadas son agrupación o *clustering*, redes neuronales, reducción de dimensión, entre otros [28].

### 2.2.1.3. APRENDIZAJE SEMISUPERVISADO

Estos algoritmos reciben como entrada una pequeña cantidad de datos sin etiquetar junto con una gran cantidad de datos etiquetados. Estos métodos tienen aplicaciones similares al aprendizaje supervisado, sin embargo, al hacer uso de datos sin etiquetas se puede mejorar la precisión del modelo resultante. Adicionalmente, estos consumen menos tiempo y son menos complejos, siendo eficaces en redes para la clasificación de tráfico, donde la cantidad de muestras inicial suele ser baja. Las principales técnicas empleadas son el aprendizaje transductivo y el aprendizaje inductivo [28].

### 2.2.1.4. APRENDIZAJE POR REFUERZO

El aprendizaje por refuerzo es una técnica de ML basada en la experiencia, en donde un agente autónomo crea una experiencia a partir de un proceso de ensayo y error para mejorar sus elecciones a futuro [29]. Este agente realiza acciones sobre un entorno para cambiar el estado de éste, recibiendo a partir de esto una recompensa o penalización determinada según el estado alcanzado y el objetivo del agente. A partir de los conceptos de esta definición, en apariencia sencilla, se ha logrado desarrollar muchos agentes avanzados de IA para realizar tareas muy complejas, desafiando en ocasiones el desempeño humano en algunas tareas específicas.

El agente y el entorno son los dos principales componentes del aprendizaje por refuerzo. El diagrama de la Figura 2.6 representa un diseño básico del sistema de RL con los ciclos de aprendizaje y acción. El agente interactúa con su entorno para aprender a realizar la mejor acción posible  $A_t$  en un estado dado  $S_t$  en el que se encuentra el entorno en un tiempo  $t$ . La acción del agente, a su vez, cambia el estado del entorno de  $S_t$  a  $S_{t+1}$  y genera una recompensa  $R_t$  para el agente. A partir de aquí, el agente selecciona la mejor acción posible para el nuevo estado  $S_{t+1}$  invocando una nueva recompensa  $R_{t+1}$  y así sucesivamente.

El agente influye en el entorno mediante las acciones y el entorno cambia de estado como respuesta a la acción, la función responsable de este cambio es conocida como función de transición de estados [30].



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

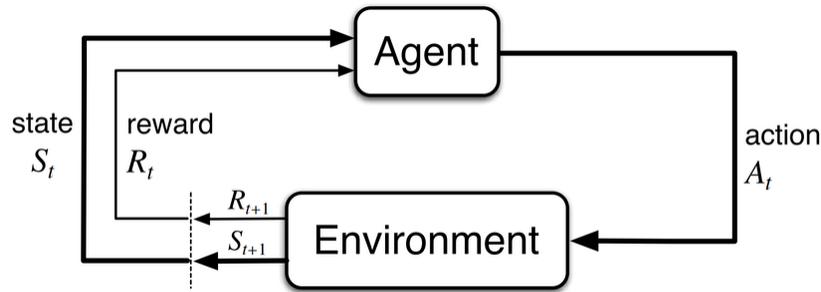


Figura 2.6. Diseño básico de un sistema de RL. Tomada de [30]

Las interacciones entre el agente y el entorno se realizan durante varios ciclos. En cada uno de estos ciclos, llamado también iteración o *time step*, el agente observa el entorno, realiza una acción y recibe un nuevo estado y recompensa. El conjunto de estado, acción, recompensa, y el nuevo estado en cada *time step* se le denomina experiencia. La secuencia de estos *time step* desde el principio hasta el final de una tarea se llama episodio. Las tareas que tienen un final natural son llamadas tareas episódicas, mientras que las tareas que no lo tienen se les llama tareas continuas.

Durante un periodo de iteraciones, el agente intenta mejorar su decisión sobre la acción a tomar, dado un estado del entorno, aprendiendo de las recompensas que recibe durante el proceso de entrenamiento. Por su parte el entorno se encarga de presentar al agente distintos estados posibles que pueden existir dentro del problema [31]. Cuando el agente dispone de una descripción completa del modelo del entorno se tiene un RL basado en modelo o *model-based*, por otra parte, cuando el agente no conoce este modelo y debe tomar decisiones con información incompleta, se le conoce como RL sin modelo o *model-free*.

### 2.2.2. DEEP LEARNING

Desde el año 2010, el éxito de las redes neuronales profundas ha sido reconocido por numerosas razones, incluyendo el desarrollo de unidades de procesamiento avanzadas, los avances en la eficiencia en algoritmos de entrenamiento como la retropropagación, el establecimiento de parámetros apropiados de inicialización, entre otros [32].

El DL puede definirse como un tipo de ML basado en redes neuronales artificiales que imita la manera en que los humanos obtienen ciertos tipos de conocimiento. Este tipo de redes pueden resolver tareas ya sea de aprendizaje supervisado, no supervisado o por refuerzo.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

Una neurona es la unidad básica de las redes neuronales profundas la cual se basa originalmente en el funcionamiento y la constitución de una neurona biológica real. Esta neurona puede tener muchas neuronas de salida de la siguiente capa o muchas neuronas de entrada de la capa anterior. Las señales provenientes de las neuronas de la capa anterior son agregadas y pasadas a través de lo que se conoce como función de activación. En DL pueden existir muchas capas entre la capa de salida y la capa de entrada en donde cada una de estas capas intermedias puede estar compuesta por ciento o incluso miles neuronas o nodos, estas capas intermedias son conocidas como capas ocultas o *hidden layers*. La Figura 2.7 muestra un ejemplo de la estructura de una red neuronal con tres capas ocultas, una capa de entrada y una capa de salida [33].

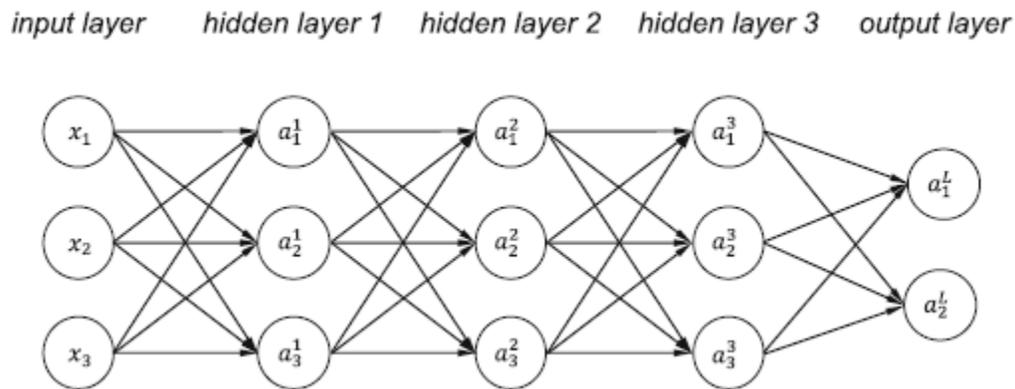


Figura 2.7. Estructura de una red neuronal profunda con tres capas ocultas.  
Tomada de [33]

Existen muchos modelos basados en DL, entre los cuales se encuentran principalmente: los codificadores automáticos (AE, *Auto Encoder*), las máquinas de Boltzmann restringidas (RBM, *Restricted Boltzmann Machine*), las redes de creencia profunda (DBN, *Deep Belief Network*), las máquinas de Boltzmann profundas (DBM, *Deep Boltzmann Machine*), las redes neuronales convolucionales (CNN, *Convolutional Neural Network*), las redes neuronales recurrentes (RNN, *Recurrent Neural Network*), las redes neuronales de grafos (GNN, *Graph Neural Network*), las redes generativas adversarias (GAN, *Generative Adversarial Network*), el aprendizaje profundo por refuerzo (DRL, *Deep Reinforcement Learning*), entre otros [34]. A continuación, se hace una descripción más detallada del DRL, método a usar para el desarrollo del algoritmo cognitivo basado en DL, el cual fue elegido ya que es uno de los modelos actuales con mayor capacidad para resolver tareas complejas de toma de decisiones secuenciales sin la necesidad de un conocimiento previo de los datos.



### 2.2.2.1. APRENDIZAJE PROFUNDO POR REFUERZO

El aprendizaje por refuerzo es una de las direcciones más importantes del ML en los últimos años. Es un proceso de aprendizaje en el cual un agente toma decisiones periódicamente, observa los resultados y ajusta su estrategia para conseguir una política óptima. No obstante, este proceso requiere de una gran cantidad de tiempo para llegar a la mejor política, ya que tiene que explorar y obtener conocimiento de un sistema completo, haciéndolo inadecuado para redes de gran escala. Por este motivo, ha surgido la idea de introducir y combinar el DL con el aprendizaje por refuerzo, obteniéndose el DRL. De esta manera, es posible superar las limitaciones mencionadas y dar paso a una nueva era de desarrollo con nuevas posibilidades para el aprendizaje por refuerzo [35].

El DRL está compuesto por dos bloques fundamentales: el RL y el DL. En cuanto a RL, normalmente se requiere una gran cantidad de memoria para el almacenamiento de las funciones de valor o funciones Q, esto debido a que, en los problemas del mundo real, los conjuntos o espacios de estado son grandes llegando en ocasiones a ser infinitos, lo que hace prácticamente imposible usar tablas para almacenar estas funciones. Aquí es donde DL entra en escena, permitiendo una aproximación de dichas funciones en un conjunto más pequeño de parámetros. Operando en conjunto, el DRL es capaz de aprovechar las ventajas de las redes neuronales profundas para mejorar el proceso de aprendizaje, aumentando la velocidad y el rendimiento de los algoritmos de RL. Gracias a lo anterior, el DRL ha sido adoptado en numerosas aplicaciones como robótica, visión por computadora, reconocimiento de voz, procesamiento de lenguaje natural, entre otros [36].

A grandes rasgos, los algoritmos básicos de DRL son clasificados generalmente en dos categorías: los métodos basados en valor o *value-based* y los métodos basados en política o *policy-based*.

- **Métodos Value-based:**

Esta clase de métodos optimizan las funciones Q o funciones de valor a partir de los estados o pares estado-acción que se tienen en la entrada de la red neuronal, se puede obtener una o múltiples neuronas de salida en esta red neuronal.

Dentro de este tipo de métodos existen principalmente dos clases, en primer lugar, están las redes Q profundas (DQN, *Deep Q-Network*), en donde se toma un estado como entrada y se retorna funciones Q aproximadas para cada acción bajo el estado de entrada. Para mejorar el proceso de entrenamiento y que sea más estable, en DQN se pueden aplicar dos técnicas conocidas como redes de destino (TN, *Target Network*) y repetición de experiencia (ER, *Experience Replay*).



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

Por otra parte, están las redes DQN dobles (DDQN, *Double Deep Q-Network*), las cuales usan dos funciones de valor para seleccionar y evaluar simultáneamente los valores de las acciones; con esto, se reduce la sobreestimación de las funciones Q que se suele presentar en DQN mejorando la estabilidad y obteniendo un mejor desempeño de los métodos [35] [36].

- **Métodos Policy-based:**

Este tipo de métodos son utilizados para aproximar directamente una función de política en lugar de una función de valor. Esto quiere decir que, en lugar de entrenar una red neuronal para generar valores de acciones o estados, se entrena para que indique cuál puede ser la siguiente acción por realizar [31].

Su clasificación se realiza de acuerdo con diferentes aspectos como la naturaleza de las funciones de política, la forma de evaluación de la política y las técnicas usadas para aprendizaje o actualización de parámetros [36].



### 3. CONMUTACIÓN ÓPTICA DE RÁFAGAS Y ENRUTAMIENTO Y ASIGNACIÓN DE ESPECTRO

En el presente capítulo se describen más a detalle algunos conceptos relacionados con los temas principales que se abordan en el trabajo de grado, a saber, la conmutación óptica de ráfagas (OBS) y el enrutamiento y asignación de espectro (RSA).

#### 3.1. CONMUTACIÓN ÓPTICA DE RÁFAGAS (OBS)

Dentro de la literatura, existen diferentes técnicas para realizar la conmutación en redes ópticas, cada una con sus características específicas y distintas aplicaciones, en el presente trabajo se estudian las más utilizadas entre éstas. En primer lugar, está la conmutación óptica de circuitos (OCS), empleada principalmente en redes de voz, en donde se establecen los recursos y enlaces previos entre los usuarios creando canales de voz dedicados, los cuales permanecen durante la comunicación y se liberan cuando ésta termina. Este enfoque orientado a conexión posee ventajas importantes al no requerir almacenamiento óptico en nodos intermedios, sin embargo, tiene como principales desventajas, los retardos excesivos que se pueden generar con transmisiones de unidades de datos pequeñas al requerir un establecimiento previo por cada conexión.

Considerando los principales inconvenientes de OCS, surge la propuesta de la conmutación óptica de paquetes (OPS), en la cual se utilizan pequeños bloques de datos llamados paquetes, cada uno compuesto por un encabezado y una carga útil que corresponde a los datos de usuario. Cuando un paquete es recibido por un nodo de la red por uno de sus puertos de entrada, este último se encarga de analizar la información almacenada en el encabezado para realizar el envío de este paquete por el puerto de salida apropiado. Este tipo de conmutación es empleado especialmente en redes de datos como las redes tipo *Ethernet* y la red de *Internet*. No obstante, se tiene un inconveniente respecto al almacenamiento de los paquetes en los nodos, ya que no es posible realizar este proceso completamente en el dominio óptico por lo que se recurre a un procesamiento electrónico del encabezado del paquete, lo que implica una alta sincronización entre dicho encabezado y la carga útil del paquete, todo esto resultando, en una mayor complejidad de la red.

Con el propósito de corregir los inconvenientes anteriormente expuestos, surge la conmutación óptica de ráfagas (OBS), la cual tiene como unidad de transporte básica la ráfaga, que está compuesta por un determinado número de paquetes y un

encabezamiento o paquete de control asignado (BCP, *Burst Control Packet*). En este tipo de conmutación se realiza el proceso de ensamble de la ráfaga teniendo en cuenta diferentes criterios, como la dirección de origen o destino, clase de servicio, tipo de datos, entre otros. De esta manera, se espera aumentar la eficiencia en las comunicaciones al reducir el número de encabezados necesarios y así brindar un servicio de transporte a nivel óptico con alto ancho de banda de forma flexible y confiable [18] [37].

### 3.1.1. Arquitectura de redes OBS

En la Figura 3.1 se muestra el modelo general de la arquitectura de una red OBS, compuesto principalmente de nodos de frontera o borde (*Edge Node*) y nodos de núcleo o centrales. Para este trabajo de grado, debido a que se desarrolla una red con enfoque centralizado (SDN), el componente del nodo de control en cada nodo es extraído y se centralizan sus funciones en un mismo nodo de control externo y separado de los demás nodos de la red; por esta razón, se considera y se describe solamente la parte del nodo de frontera con algunas modificaciones [38].

El nodo de frontera está compuesto por un nodo de ingreso y un nodo de egreso. En el nodo de ingreso se realizan las funciones de clasificación de paquetes, ensamble y transmisión de ráfaga, mientras que en el nodo de egreso es donde ocurre el desensamble de la ráfaga entrante y su transmisión al destino.

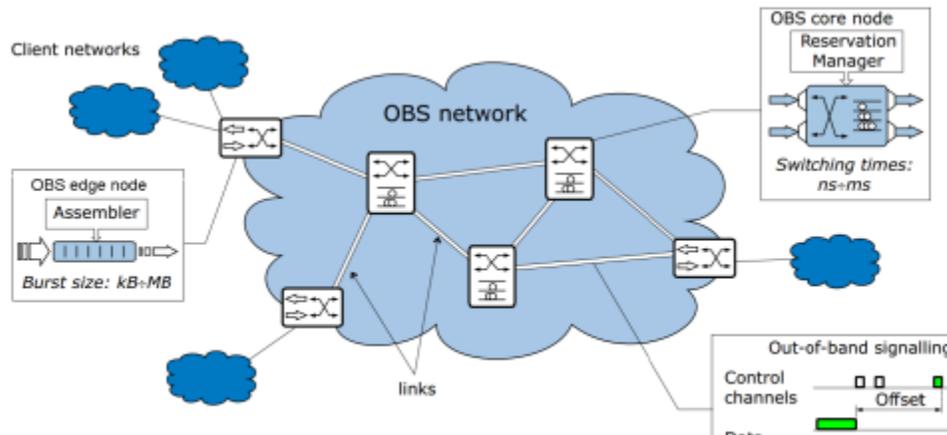


Figura 3.1. Arquitectura de una Red por OBS. Tomada de [38]

### 3.1.2. Nodo Frontera OBS

Como se mencionó anteriormente, este nodo tiene las funciones de clasificación, ensamble y transmisión de ráfagas y, además, la aplicación de los esquemas de señalización, el enrutamiento y la asignación de recursos. La arquitectura del nodo

de ingreso es presentada en la Figura 3.2, donde se especifica ordenadamente cada proceso llevado a cabo desde que un paquete ingresa al nodo.

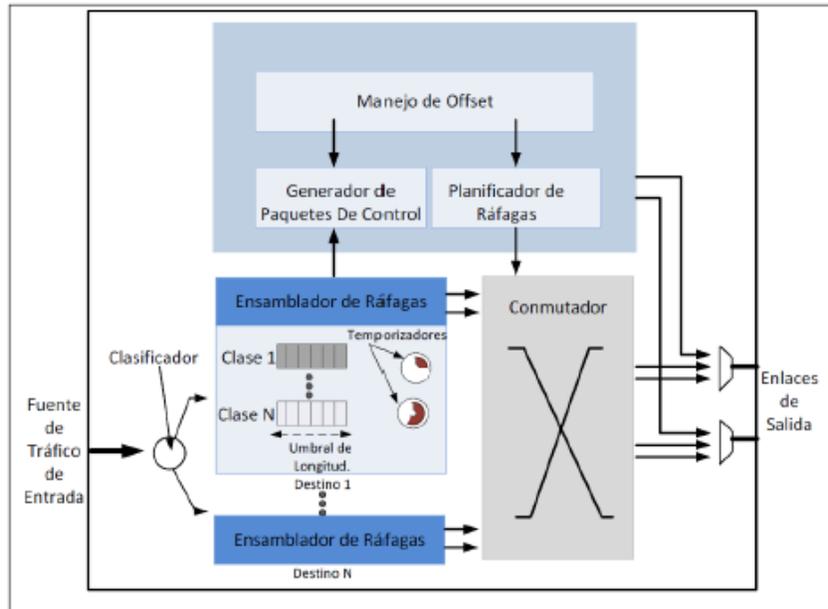


Figura 3.2. Arquitectura del Nodo de Ingreso en OBS. Tomada de [37]

En el momento en que el paquete ingresa al nodo de frontera, es clasificado siguiendo algunos criterios como la dirección de destino, el tipo de servicio, prioridad, entre otros. En segundo lugar, se ensambla la ráfaga y finalmente es transmitida a través de uno de los canales de salida disponibles hacia el nodo de frontera de destino [37].

### 3.1.2.1. Ensamble de ráfagas

El ensamble de ráfagas es el proceso en donde se agrupa los paquetes entrantes en los nodos de ingreso de la red OBS. Estos paquetes son almacenados temporalmente en colas o búferes de acuerdo con los criterios a consideración. El mecanismo de ensamble se encarga de ubicar estos paquetes en la ráfaga correspondiente basándose en una política que determina el valor límite para detener el proceso de ensamble. Estas políticas se basan al menos de un parámetro básico que determina en qué momento crear una ráfaga y en qué momento detener su creación para enviarla por el canal correspondiente. Los esquemas comúnmente usados para este proceso se basan de parámetros como el tiempo de ensamble, la



longitud de ráfaga o también, se basan en más de un parámetro los cuales son conocidos como esquemas híbridos o mixtos.

- **Esquema de ensamble de ráfaga basado en tiempo**

En este esquema se define un valor máximo de tiempo para el proceso de ensamble de la ráfaga. Después de la llegada del primer paquete para ensamblar, se inicia un temporizador con una duración determinada durante la cual se realiza el proceso de ensamble de la ráfaga. Esto implica que la longitud de las ráfagas varíe de acuerdo con la carga de entrada y que existan intervalos de tiempo uniformes entre ráfagas sucesivas. Los valores de tiempo pueden ser seleccionados a partir de los requerimientos de retardo extremo a extremo de paquetes de la red [11].

- **Esquema de ensamble de ráfaga basado en longitud**

Este esquema determina un número límite de paquetes que puede contener una ráfaga, en el momento en que se alcanza ese umbral la ráfaga es formada y enviada. Por este motivo, las ráfagas son de tamaño fijo y no existen intervalos de tiempo periódicos entre el envío de éstas [11] [39].

- **Esquema de ensamble de ráfaga híbrido**

Los dos esquemas presentados anteriormente presentan algunos inconvenientes. Por un lado, en el esquema basado en tiempo, los niveles altos de carga en la red pueden provocar la creación de ráfagas muy grandes, mientras que, en el esquema basado en longitud, si se tiene una carga baja, puede ocasionar grandes retardos en la red debido a que se debe esperar a que se llegue al límite de paquetes.

Para la solución de estos inconvenientes, se modela un esquema de ensamble de ráfaga híbrido, el cual consiste en una combinación de los dos esquemas anteriores, es decir, se establece tanto un límite de paquetes por ráfaga como un tiempo límite de ensamble y la ráfaga es enviada en el momento en que se alcanza alguno de estos dos valores. De esta manera, se logra reducir el retardo al tener un tiempo límite para el ensamble de la ráfaga y también, se asegura que la ráfaga no sea demasiado grande al tener un valor umbral para el número de paquetes por ráfaga [18].

De acuerdo con lo anterior, se considera que el esquema de ensamble de ráfaga híbrido es el adecuado en el presente trabajo de grado y se selecciona para ser implementado en la red NSFNeT OBS/SDEON.



### 3.1.2.2. Señalización

La señalización es otra de las funciones del nodo frontera con la cual se realiza la configuración de los mecanismos de conexión. En este proceso previo a la transmisión, se realiza el establecimiento de la conexión y se reservan los recursos necesarios, todo esto, a través de un canal dedicado diferente a los usados para la transmisión de los datos.

En el caso de una red distribuida, el proceso de señalización se realiza enviando mensajes para la reserva de recursos entre los nodos de la red, en cambio, en una red centralizada (SDN) para este proceso se envía un mensaje de petición de establecimiento de conexión desde el nodo origen hasta el nodo de control el cual tiene conocimiento del estado de los recursos de la red, y este último, regresa un mensaje con la información de los recursos asignados para el establecimiento de esa conexión. Para este trabajo de grado, se emplea la señalización centralizada con reserva extremo a extremo por tratarse de una red con enfoque SDN [18].

### 3.1.3. Selección de temporizador y umbral de longitud de onda

Uno de los principales aspectos a considerar en el esquema de ensamble de ráfagas híbrido es la selección de los valores adecuados para el tiempo máximo de ensamble y la longitud máxima de ráfaga de modo que se logre minimizar la probabilidad de bloqueo y pérdida de paquetes de la red. Si el umbral de longitud asignado es muy corto se obtiene un número elevado de ráfagas que debes ser procesadas en la red y se aumentará el número de contenciones. Por el contrario, si este umbral es muy alto, se reducirá el número de ráfagas en la red, pero aumentará el número promedio de paquetes perdidos por contención debido al gran tamaño de las ráfagas. Es por esto, que el desempeño de la red puede verse mejorado al seleccionar valores óptimos para el tamaño máximo de ráfaga y el tiempo máximo de ensamble. En la Figura 3.3 se muestra una gráfica con los efectos de la carga en el esquema de ensamble de ráfaga basado en tiempo de ensamble y longitud de ráfaga [18].



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

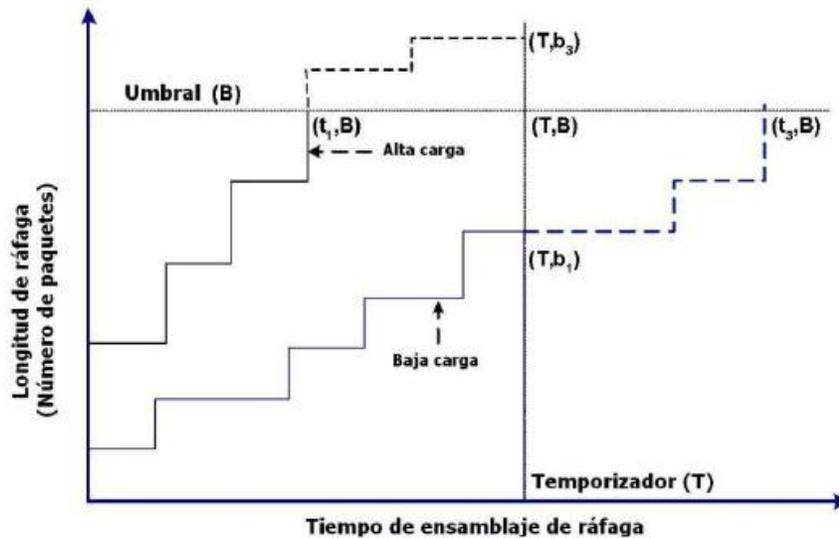


Figura 3.3. Efectos de la Carga en Esquema de Ensamble de Ráfaga basado en tiempo y longitud de ráfaga. Tomada de [18]

### 3.2. ENRUTAMIENTO Y ASIGNACIÓN DE ESPECTRO

Una red de telecomunicaciones óptica está conformada por dispositivos eléctricos, enlaces de fibra óptica y otro tipo de elementos ópticos permitiendo crear conexiones desde un origen a un destino para la transmisión de información a través de haces de luz. Para el establecimiento de una conexión óptica, se debe realizar previamente una petición de conexión desde el nodo origen hasta el nodo de control en el caso de las redes centralizadas, esta petición debe contener información acerca de los nodos de origen y de destino, así como también, el tamaño de ancho de banda que se necesita transportar. El nodo de control, para dar respuesta a la petición, debe resolver uno de los principales problemas en el caso de las redes ópticas elásticas, el enrutamiento y la asignación de espectro (RSA), el cual consiste en encontrar una ruta apropiada y asignar una porción de espectro disponible en esa ruta de acuerdo con los requerimientos de la petición, permitiendo el establecimiento de la conexión si se cuenta con los recursos requeridos o en caso contrario, rechazando la petición [40].

Para dar solución al problema de RSA y teniendo en cuenta los altos requerimientos de ancho de banda de los servicios y aplicaciones actuales, se han propuesto en la literatura diversos algoritmos que permiten la ejecución de las operaciones, ya sea de manera independiente (R+SA) o de manera simultánea (*Joint RSA*). En las siguientes secciones se discutirán estos conceptos con mayor detalle.



### 3.2.1. Funcionamiento RSA

En las redes ópticas elásticas, se reduce la granularidad para un uso más eficiente del espectro óptico, proponiendo modelos con *slots* de ancho de 6.25 GHz y 12.5 GHz, siendo la primera opción no operable hasta el momento.

De acuerdo con la recomendación G.694.1 de la ITU-T, se sugiere *slots* con frecuencia nominal central de 6.25 GHz y ancho de 12.5 GHz. De aquí se definen los conceptos de canal, *slot* y *slice*. Un canal se caracteriza por una frecuencia central (CF, *Central Frequency*) y un número determinado de *slots*, cada uno de estos *slots* tiene un ancho de 12,5 GHz y está conformado por una frecuencia central y dos *slices* consecutivos, y, a su vez, cada *slice* tiene un ancho de 6.25 GHz.

Partiendo de lo anterior, las frecuencias permitidas para el espaciamiento de canales en una fibra, tomando como referencia la frecuencia de 193.1 THz (1552.52 nm), están dadas por la ecuación 1 [18].

$$Frecuencias\ Permitidas = 193.1 + n * 0.00625 \quad (1)$$

Donde  $n$  es un entero positivo o negativo, con valor mínimo de -246, correspondiente al índice de frecuencia de 1565 nm, y máximo de 455, correspondiente al índice de frecuencia de 1530 nm.

Para el cálculo del ancho de *slot* (SW, *Slot Width*) el cual hace referencia al número de *slots* asignados a una conexión, se define la ecuación 2, donde  $m$  es un número entero positivo que representa el número de *slots* para el ancho de banda requerido [18].

$$Slot\ Width = 12.5\ GHz * m \quad (2)$$

Otra ecuación de importancia definida en la ecuación 3, es utilizada para calcular la frecuencia central de un canal asociado a una conexión óptica.

$$Frecuencia\ Central = f_i + 0.00625 * m \quad (3)$$



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

En la Figura 3.4, se observa un ejemplo de uso de espectro en una EON donde se tienen dos canales consecutivos de 50GHz y otros dos canales consecutivos de 75GHz, cada uno con sus respectivos valores para  $n$  y  $m$ . El rango de frecuencias desde 193.125 THz hasta 193.18125 THz no es asignado, este puede servir como banda de guarda entre los dos conjuntos de canales o puede ser asignado, por ejemplo, para un canal adicional de 50 GHz ( $n=8, m=4$ ), dejando 6.25 GHz sin asignar.

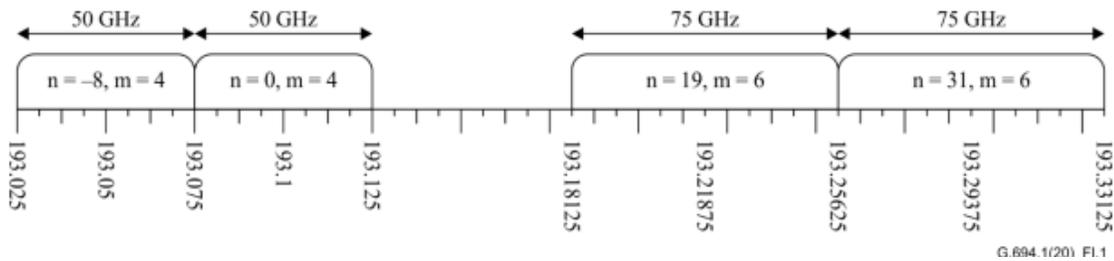


Figura 3.4. Ejemplo de Uso de Espectro para una EON. Tomada de G.694.1.

El principal objetivo de RSA es encontrar una ruta con suficiente espectro disponible para asignar en ésta el ancho de banda requerido por la conexión entrante. En cuanto al proceso de enrutamiento, las rutas posibles pueden ser seleccionada mediante algoritmos basados en diferentes criterios, como la distancia hacia la ruta, cantidad de nodos o saltos en la ruta, entre otros. Por ejemplo, un algoritmo ampliamente conocido llamado Dijkstra o algoritmo de caminos mínimos, se basa en la cantidad de número de saltos de una ruta para determinar los caminos más cortos que se pueden tomar y realizar la selección de las rutas apropiadas.

Adicionalmente, para el proceso de asignación de espectro, deben tenerse en cuenta algunas condiciones importantes descritas a continuación.

- **Contigüidad del espectro:** Los *slots* de frecuencia que conforman un canal asignado a una conexión óptica deben ser adyacentes (contiguos) en el espectro.
- **Continuidad del espectro:** El canal asignado a una conexión óptica debe ser el mismo (misma frecuencia central y misma banda de frecuencias) a lo largo de todos los enlaces de la ruta.
- **No superposición del espectro:** Cada *slot* de frecuencia del canal asignado a una conexión solo puede ser utilizado por esa única conexión al mismo tiempo.
- **Espectro Limitado:** Debe ser asignado a una conexión un número finito de *slots*.



ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

- **Bandas de Guarda:** Las conexiones adyacentes en el espectro deben estar separadas por al menos un *slot* de frecuencia sin asignar para evitar interferencias.

En la Figura 3.5, se muestra un ejemplo donde se cumplen las condiciones anteriormente descritas. Para este ejemplo, se necesita el establecimiento de una conexión que ocupa dos *slots* desde el nodo 1 hasta el nodo 4 a partir del espectro disponible en la red en ese momento. En la gráfica superior se asigna una ruta más corta conformada por los enlaces 1 y 4 (*Link 1, 4*), sin embargo, la conexión es rechazada debido a que, en el espectro disponible, existe un conjunto de dos *slots* por cada enlace que son contiguos, pero no cumplen con la condición de continuidad. Por otra parte, en la gráfica inferior, la conexión es establecida ya que en la ruta seleccionada conformada por los enlaces 1, 2 y 3 (*Link 1, 2, 3*) sí existen dos *slots* disponibles que satisfacen al mismo tiempo las condiciones de contigüidad y continuidad [16].

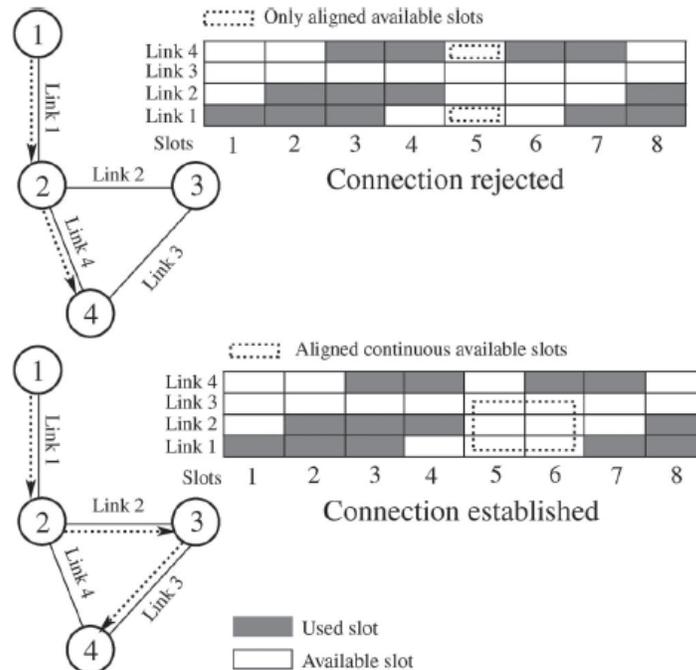


Figura 3.5. Establecimiento de conexión en una red FlexGrid considerando las restricciones de continuidad y contigüidad. Tomada de [16]



### 3.2.2. Tipos de Enrutamiento

Los algoritmos de enrutamiento existentes dentro de la literatura se clasifican generalmente en cuatro categorías: algoritmos de enrutamiento fijo (FR, *Fixed Routing*), algoritmos de enrutamiento fijo alternado (FAR, *Fixed Alternate Routing*), algoritmos de enrutamiento por menos congestión (LCR, *Least Congested Routing*) y algoritmos de enrutamiento adaptativo (AR, *Adaptive Routing*). Estos tipos de enrutamientos están destinados principalmente para encontrar las rutas adecuadas entre pares origen-destino de la red [16].

- **Enrutamiento Fijo:**

En este tipo de enrutamiento se calcula únicamente una sola ruta fija para cada par origen-destino usando algoritmos de camino más corto como el algoritmo Dijkstra. Cuando se recibe una petición de conexión, el algoritmo intenta establecer un *lightpath* a lo largo de la ruta fija predeterminada. Si alguno de los enlaces de la ruta no dispone de los *slots* requeridos la petición de conexión es bloqueada. En caso de que existan varias alternativas para la cantidad de *slots* requeridos, se hace uso de una política de asignación de espectro para elegir la opción más apropiada. Con este tipo de enrutamiento se busca reducir el tiempo de establecimiento de la conexión, pero es propenso a altas probabilidades de bloqueo. Los algoritmos que más se destacan son: algoritmo de camino más corto (SP, *Shortest Path*), algoritmo de peso por salto (HW, *Hop Weight*) y algoritmo de menor distancia (DW, *Distance Weight*) [16].

- **Enrutamiento Fijo Alternado:**

Este tipo de enrutamiento es una versión actualizada de FR. En cada uno de los nodos de la red se tiene una tabla de enrutamiento la cual contiene una lista ordenada de rutas fijas hacia los demás nodos de la red, estas rutas son calculadas con anticipación. Cuando se recibe una petición de conexión de un par origen-destino, el nodo origen intenta establecer un *lightpath* a través de cada una de las rutas de la tabla de enrutamiento hasta que se encuentra una ruta con los *slots* requeridos. Si no existe una ruta en la lista con los *slots* requeridos o la ruta está ocupada, la petición de conexión es bloqueada. En caso de que existan varias opciones para la cantidad de *slots* requeridos en la ruta seleccionada, se hace uso de una política de asignación de espectro para elegir la opción más apropiada. Este tipo de algoritmos tienen mayor complejidad, pero proporcionan una menor probabilidad de bloqueo que los algoritmos de FR. Un ejemplo de este tipo de técnica es el algoritmo de mínimo número de saltos (MNH, *Minimum Number of Hops*) [16].



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

- **Enrutamiento por menos congestión:**

En los algoritmos basados en enrutamiento por menos congestión, se predetermina una secuencia de rutas para cada par origen-destino como en FAR. Según el tiempo de llegada de las peticiones de conexión, se seleccionan entre las rutas predeterminadas, las rutas de menor congestión, es decir, las rutas con mayor disponibilidad de *slots*. Este tipo de técnica posee una alta complejidad de cálculo y una probabilidad de bloqueo similar a FAR.

- **Enrutamiento Adaptativo:**

En el enrutamiento adaptativo, las rutas para cada par origen-destino son seleccionadas de forma dinámica de acuerdo con la información de los estados de los enlaces de la red, la cual se actualiza constantemente cuando se establece o se libera alguna conexión. Este tipo de enrutamiento considera todas las rutas posibles entre el par origen-destino, proporcionando de este modo, menor probabilidad de bloqueo, sin embargo, requiere protocolos de soporte de control y gestión para actualizar constantemente las tablas de enrutamiento de los nodos y requiere también un mayor tiempo para su configuración. El AR es más adecuado para implementaciones de red centralizadas que para las implementaciones distribuidas.

En la Tabla 3.1, se presenta un resumen de los tipos de enrutamiento anteriormente descritos en donde se compara su desempeño en términos de la probabilidad de bloqueo (BP, *Blocking Probability*) y el tiempo promedio de configuración (AST, *Average Setup Time*). La probabilidad de bloqueo se define como la relación del número de conexiones bloqueadas sobre la cantidad de peticiones de conexión en la red. Por su parte, el tiempo promedio de configuración se define como el tiempo total de ejecución requerido para establecer todas las conexiones en la red [16].

Tabla 3.1. Resumen Comparativo de Tipos de Enrutamiento. Adaptada de [16].

Tipo de Enrutamiento		BP	AST	ON/OFF LINE
Estático	FR	Alta	Corto	Offline
	FAR	Media	Medio	Offline
	LCR	Media	Medio	Offline
Dinámico	AR	Baja	Largo	Online

A partir de la Tabla 3.1 se observa que FR tiene el más bajo tiempo promedio de configuración de todos los algoritmos, sin embargo, su probabilidad de bloqueo es la más alta. En contraste, AR proporciona el mejor desempeño en términos de probabilidad de bloqueo, pero tiene el mayor tiempo promedio de configuración y



mayor complejidad. Por lo anterior, para el presente trabajo de grado se selecciona el enrutamiento fijo para el algoritmo clásico basado en SCC y el enrutamiento adaptativo para el algoritmo cognitivo basado en DL, que pese a tener mayor complejidad este último para su implementación, proporciona una baja probabilidad de bloqueo y es ideal para redes centralizadas.

### 3.2.3. Esquemas de asignación de Espectro

Una de las principales ventajas de las redes elásticas o *Flexgrid* es la capacidad de asignar de forma flexible los recursos espectrales de acuerdo a las demandas de tráfico. El espectro puede ser usado eficientemente, primero, gracias a la alta granularidad en este tipo de redes que permite ajustar estrechamente el espectro asignado con el ancho de banda requerido por la conexión óptica, y, segundo, gracias a la capacidad de asignación de espectro de forma adaptativa en respuesta a las variaciones en el tráfico.

Considerando lo anterior, existen tres esquemas principales para la asignación de espectro empleando rejilla flexible o *Flexgrid*: asignación de espectro fijo en el tiempo (FSA, *Fixed Spectrum Allocation*), asignación de espectro semi-elástico en el tiempo (SSA, *Semi-elastic Spectrum Allocation*) y asignación de espectro elástico en el tiempo (ESA, *Elastic Spectrum Allocation*), estos son descritos a continuación [41].

#### 3.2.3.1. Asignación de Espectro Fijo en el tiempo

Para la asignación de espectro fijo en el tiempo, tanto la frecuencia central como el ancho del espectro permanecen estáticos a lo largo del tiempo. En un intervalo de tiempo determinado, las demandas pueden hacer uso ya sea de todo este espectro o solo de una porción para transmitir el ancho de banda requerido por la conexión.

De acuerdo con lo anterior, esta política no proporciona ningún tipo de elasticidad, debido a que en la asignación de espectro no se tiene en cuenta las variaciones en los requerimientos de ancho de banda. Si el ancho de banda requerido por una conexión es inferior a la capacidad del espectro asignado, se realiza el establecimiento de la conexión, lo que puede conducir a un uso no eficiente de la capacidad de la red al dejar una porción del espectro inutilizado. Por otra parte, si el ancho de banda requerido por la conexión es mayor que la capacidad del espectro asignado, la demanda no podrá ser atendida al no haber espectro suficiente. La Figura 3.6 (a) y (b) muestra un ejemplo de cada uno de los dos casos mencionados. La Figura 3.6 (a) muestra el caso cuando se tiene un ancho de banda requerido menor a el espectro asignado, mientras que en la Figura 3.6 (b) se observa el caso cuando el ancho de banda es más alto que el espectro asignado a la conexión [42].

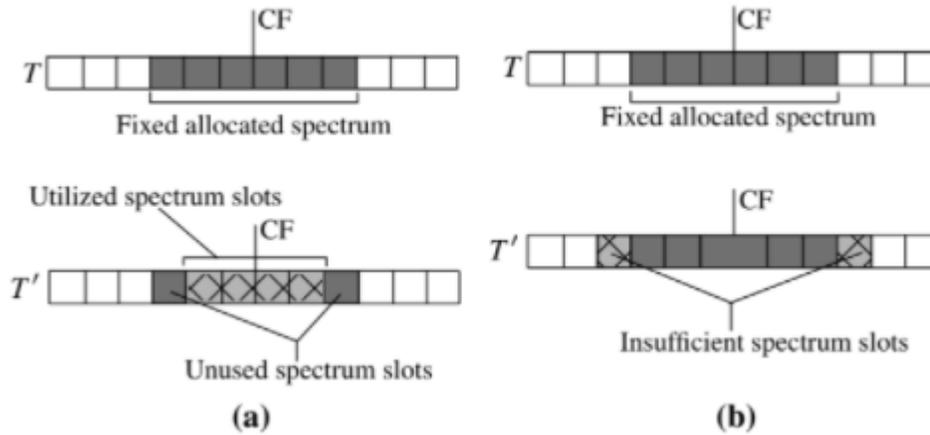


Figura 3.6. Escenarios de Asignación de espectro fijo en el tiempo. a) Subuso del espectro disponible. b) Espectro Disponible Insuficiente. Tomada de [42]

### 3.2.3.2. Asignación de Espectro Semi-Elástico en el tiempo

En este tipo de asignación de espectro, la CF permanece fija mientras que el espectro asignado a la demanda puede variar en cada intervalo de tiempo. Para adaptarse al ancho de banda requerido en un momento determinado, se asignan o se liberan *slices* de frecuencia del espectro asignado y se mantiene la CF invariante, logrando de este modo, que el espectro asignado corresponda al espectro utilizado. Los *slices* de frecuencia pueden ser compartidos entre conexiones adyacentes, sin embargo, solo pueden ser utilizados por una conexión durante un periodo de tiempo. Esta política proporciona mayor flexibilidad que en el caso de FSA [41].

Cuando se tiene el caso en que el ancho de banda requerido disminuye, la capacidad del espectro asignado puede ser reducida al liberar los *slices* innecesarios (reducción), los cuales pueden ser utilizados por otras conexiones, este escenario es representada en la Figura 3.7 (a). Por otra parte, si el ancho de banda requerido aumenta, nuevos *slices* pueden ser asignados al espectro (expansión) para poder satisfacer los requerimientos de la conexión, la Figura 3.7 (b) muestra un ejemplo de este escenario.

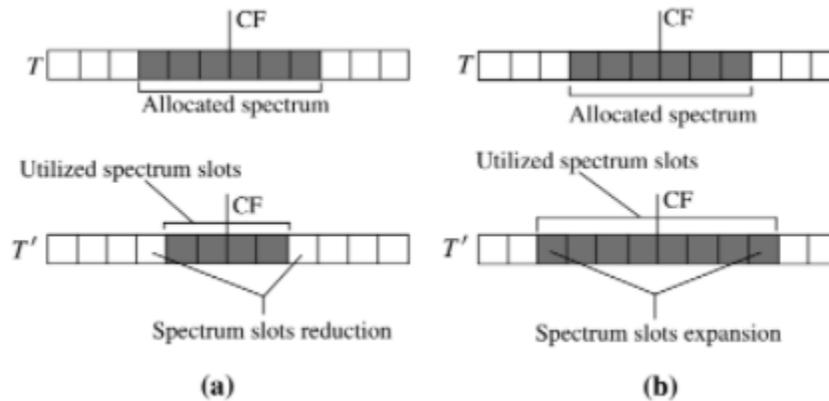


Figura 3.7. Escenarios de Asignación de espectro semi-elástico en el tiempo. a) Reducción de Espectro. b) Expansión de Espectro. Tomada de [42]

### 3.2.3.3. Asignación de Espectro Elástico en el tiempo

En la asignación de espectro elástico en el tiempo, tanto la CF como el ancho del espectro o número de *slots* pueden ser modificados en cada intervalo de tiempo. Gracias a esto, esta técnica cuenta con un grado mayor de flexibilidad que los anteriores. Se tienen principalmente dos variantes: Expansión/Contracción de espectro y Reasignación de espectro.

En el caso de la expansión/contracción de espectro, el movimiento de la CF es corto y está limitado a un cierto rango, mientras que la asignación del espectro está restringida a las CF vecinas. Para el segundo caso, es posible reasignar el espectro completamente sin ningún límite en el movimiento de la CF. En la Figura 3.8 (a) y (b) se representan el primer y segundo caso respectivamente [41].

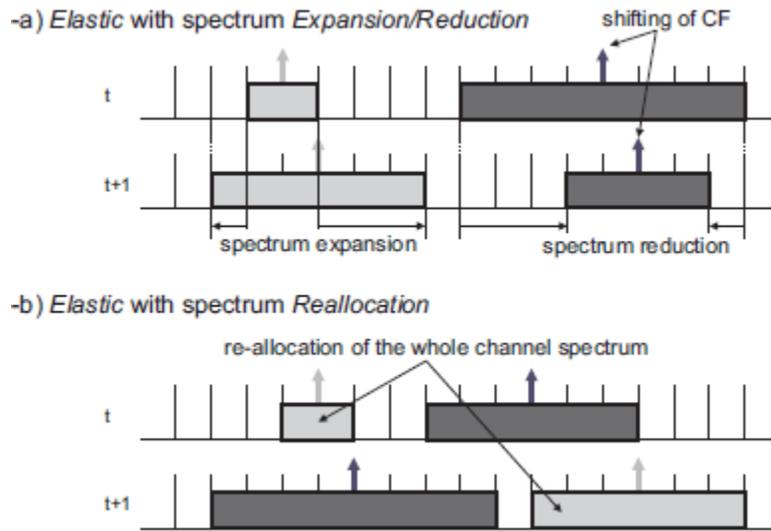


Figura 3.8. Escenarios de Asignación de Espectro Elástico en el tiempo. a) Expansión/Contracción de Espectro. b) Reasignación de Espectro. Tomada de [41]

De acuerdo con las características de los esquemas de asignación de espectro presentados, se selecciona para los dos algoritmos el esquema de asignación de espectro elástico en el tiempo debido a su mayor grado de flexibilidad para asignar el espectro de acuerdo con los requerimientos de cada conexión.

### 3.2.4. Algoritmos de Enrutamiento y Asignación de Espectro

Los algoritmos para RSA en redes ópticas elásticas pueden ser clasificados en dos categorías principales: algoritmos RSA offline o estáticos y algoritmos RSA online o dinámicos. Estas dos categorías se describen a continuación.

#### 3.2.4.1. Algoritmos RSA Offline

El objetivo de los algoritmos para resolver el problema de RSA estático, consiste en asignar una ruta física y espectro contiguo a cada demanda de forma que se minimice la cantidad de espectro asignado. Para llevar a cabo el proceso, estos algoritmos deben de conocer de antemano el comportamiento de los patrones de tráfico en la red, donde las variaciones ocurren después periodos de tiempo prolongados [43] [44].

La entrada recibida es generalmente un grupo de demandas de tráfico las cuales deben ser procesadas en conjunto, lo que implica un aumento en la complejidad



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

para el cálculo de la ruta óptima y la asignación de espectro para cada demanda. Para encontrar la solución óptima a este problema, se ha propuesto el uso de la programación lineal entera (ILP, *Integer Lineal Programming*), el cual es un método de programación matemático empleado para la resolución de problemas complejos y la obtención de soluciones enteras óptimas. Con el método de ILP se obtiene la maximización o minimización de alguna variable, que, para el caso de las redes ópticas, puede ser maximizar la cantidad de *slots* de frecuencia consecutivos liberados luego de una petición, minimizar el número de saltos de una ruta, entre otros [45].

Las formulaciones ILP no pueden ser resueltas dentro de una cantidad tiempo razonable debido a las topologías de red encontradas en la práctica, es por esto, que se ha propuesto a partir de diversos estudios, el uso de una serie de algoritmos heurísticos para obtener soluciones de manera más eficiente [43].

### 3.2.4.2. Algoritmos RSA Online

Los algoritmos para RSA *Online* son empleados cuando se tienen escenarios de tráfico dinámicos, en los cuales, los usuarios envían peticiones de conexión de manera aleatoria y éstas son servidas dinámicamente según van llegando. Para este caso, no es posible encontrar una solución óptima debido a que la red se va ocupando de acuerdo con la llegada de las peticiones [45].

Para realizar el establecimiento de una conexión entre un par origen-destino, se tiene en cuenta el estado de la red en el momento de la petición, con el fin de conocer si los recursos espectrales disponibles son suficientes para llevar a cabo el proceso, este estado de la red va actualizándose a medida que se establecen o se liberan conexiones. En el momento en que se recibe una nueva petición, se ejecuta un algoritmo para evaluar si es posible atenderla, en caso de no contar con los recursos requeridos, la petición de conexión se bloquea. Por lo anterior, una de las métricas claves para medir el rendimiento de la red en este tipo de escenario, es la probabilidad de bloqueo de las peticiones de conexión [43].

Teniendo en cuenta su menor tiempo de cálculo, se proponen algoritmos heurísticos para la solución del problema de RSA dinámico, los cuales se pueden dividir en dos categorías:

- **Algoritmos de una etapa:**

En este tipo de algoritmos, las operaciones de RSA se realizan al mismo tiempo, algunos ejemplos son: algoritmo Dijkstra modificado de camino más corto (MSP,



*Modified Dijkstra's Shortest Path*) y algoritmo de búsqueda de vector de ruta de restricción de espectro (SCVPS, *Spectrum-Constraint Path Vector Searching*) [45].

- **Algoritmos de dos etapas:**

En este enfoque, el problema de RSA es dividido en dos partes: el enrutamiento y la asignación de espectro. En primer lugar, se hace el cálculo de la ruta mediante un algoritmo de enrutamiento y posteriormente, se utiliza un algoritmo de asignación de espectro para realizar la operación faltante [45].

### 3.2.4.3. Algoritmos Heurísticos RSA

Considerando el enfoque de dos etapas descrito anteriormente, en donde se realizan de manera independiente y secuencial los procesos de enrutamiento y asignación de espectro, se describe a continuación algunos algoritmos propuestos en la literatura para resolver individualmente el problema de asignación de espectro en redes ópticas elásticas: algoritmo de primer ajuste (FF, *First Fit*), algoritmo de ajuste aleatorio (RF, *Random Fit*), algoritmo de último ajuste (LF, *Last Fit*), algoritmo de primer y último ajuste (FLF, *First-Last Fit*), algoritmo del menos usado (LU, *Least Used*), algoritmo del más usado (MU, *Most Used*) y algoritmo de ajuste exacto (EF, *Exact Fit*).

- **First Fit:**

En la política de asignación de espectro de primer ajuste, los *slots* del espectro son indexados y son añadidos a una lista de índices. A partir de esta lista, se selecciona los *slots* disponibles con índice más bajo y se asignan al *lightpath* para servir la petición de conexión. Cuando la conexión es liberada, los *slots* asignados son adicionados nuevamente a la lista de *slots* disponibles. De esta manera, es posible asignar el espectro requerido por cada conexión de forma ordenada y compacta, dejando el resto de *slots* disponibles para futuras conexiones. Esta política se considera una de las mejores debido a su baja complejidad y a la baja probabilidad de bloqueo que proporciona [46].

- **Random Fit:**

En RF, cuando llega una petición de conexión se selecciona aleatoriamente los *slots* requeridos a partir de una lista donde se almacena el estado de cada *slot* (ocupado o disponible) y se asignan a la ruta para atender la conexión. Esta lista es actualizada en el momento en que se establecen o se liberan los *slots* de una conexión. De este modo, se reduce la posibilidad de tener múltiples conexiones eligiendo el mismo espectro [46].



- **Last Fit:**

En contraste a FF, en esta política se seleccionan los *slots* requeridos con el mayor índice desde la lista de estado de *slots* y se asignan al camino de luz correspondiente para atender la petición de la conexión. La lista es actualizada en el momento en que se establece o se libera una conexión [16].

- **First-Last Fit:**

En la política de FLF, los *slots* de espectro de cada enlace son divididos en varias particiones. A partir de esto, se intenta seleccionar los *slots* con índice más bajo en la partición de números pares de la lista de *slots* disponibles, y, para la partición de números impares, se intenta seleccionar los *slots* con índice más alto de la lista de *slots* disponibles. Con este tipo de asignación de espectro, se pretende brindar más espacios de *slots* contiguos disponibles que en las políticas de FF y RF [46].

- **Least Used:**

Esta política realiza la asignación de *slots* de acuerdo con una lista en donde se almacenan los *slots* disponibles que han sido usados por la menor cantidad de enlaces de fibra de la red. Si existe un gran número de *slots* con el mismo uso mínimo, se usa la política de FF para asignar el espectro. Con la política LU, se intenta distribuir la carga uniformemente a través de todos los *slots* de espectro de la red [46].

- **Most Used:**

Mediante esta política, se asigna el espectro a una ruta desde una lista de *slots* disponibles que han sido usados por el mayor número de enlaces de fibras de la red. Al igual que en LU, si existen muchos *slots* disponibles con el mismo uso máximo, se aplica la política de FF para asignar el espectro. De este modo, se intenta conseguir la máxima reutilización de espectro en la red [46].

- **Exact Fit:**

Partiendo desde el inicio del canal de frecuencia asignado, la política EF busca el bloque de *slots* disponible que se ajuste exactamente al espectro requerido por la conexión. Si existe un bloque que coincida con los requerimientos es asignado a la conexión, y, en caso contrario, se emplea la política de FF para la asignación del espectro. De esta manera, se intenta reducir el problema de fragmentación en la red [46].



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

Para representar la funcionalidad de las políticas de asignación de espectro descritas, se plantea un segmento del espectro de una red como se muestra en la Figura 3.9. Como ejemplo, se tienen dos peticiones de conexión entrantes las cuales utilizan el enlace 1 y 2 (*Link 1*, *Link 2*) y requieren un *slot* de capacidad para establecer los caminos de luz. Adoptando la política de FF, se asignaría el *slot* 3 para la primera conexión y el *slot* 7 para la segunda. Con LF, se asignaría el *slot* 16 para la primera conexión y el 15 para la segunda. Con FLF, se seleccionaría el *slot* 3 para la primera petición de conexión y para la segunda el *slot* 16. El *slot* 16 ha sido usado dos veces y el *slot* 3 una vez, por lo que serían asignados por la política de MU. Debido a que los *slots* 10, 11 y 14 no han sido usados, LU seleccionaría dos de éstos para las dos peticiones de conexión. RF, asignaría dos *slots* entre los *slots* 3, 7, 10, 11, 14, 15 y 16 para las dos conexiones con probabilidad de asignación igual para cada uno. Finalmente, con EF, se asignaría el *slot* 3 para la primera conexión y el *slot* 7 para la segunda.

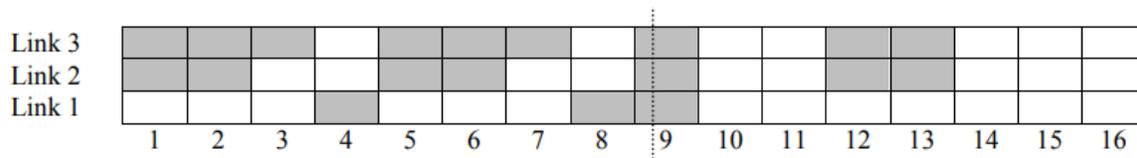


Figura 3.9. Ejemplo de uso de un segmento del espectro de una red. Tomada de [46]

Según lo explicado en este capítulo, para el presente trabajo de grado se emplean algoritmos de dos etapas. Para el problema de enrutamiento algoritmos modificados de camino más corto o Dijkstra. Por otra parte, para solucionar el problema de asignación de espectro, se emplea el algoritmo heurístico FF tanto para el primer escenario de simulación con un algoritmo clásico basado en SCC como para el segundo escenario con un algoritmo cognitivo basado en DL. Estos algoritmos para la asignación de espectro serán descritos con mayor detalle en el siguiente capítulo.



## 4. DISEÑO E IMPLEMENTACIÓN

En el presente capítulo se presentan el diseño, caracterización e implementación del modelo de la red NSFNeT OBS/SDEON en donde se describe cada módulo implementado en la herramienta de simulación seleccionada. Se describen también, la metodología aplicada para el desarrollo del trabajo de grado, la metodología para la realización de las simulaciones y el diseño de los dos algoritmos para el problema de RSA.

### 4.1. METODOLOGÍA DE TRABAJO

De acuerdo con el propósito de este trabajo de grado, se selecciona una metodología clásica en cascada dividida por fases como se muestra en la Figura 4.1 [47].

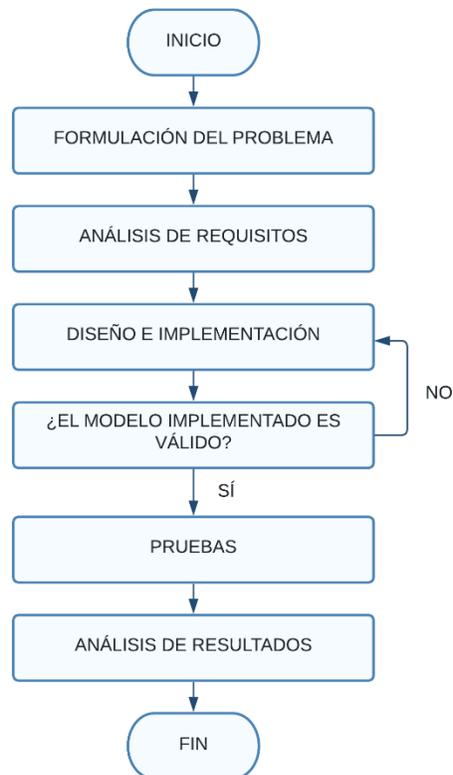


Figura 4.1. Diagrama de flujo de la metodología empleada. Adaptada de [47]



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

En primer lugar, se realiza la formulación del problema seguido de un análisis de requisitos donde se recopila información de las características requeridas en el sistema y se selecciona las herramientas a utilizar en el proyecto. En la siguiente etapa de diseño e implementación se realiza el modelado y la caracterización de la red NSFNeT OBS/SDEON en la herramienta seleccionada, como también, el diseño de los dos algoritmos y la integración individual de éstos en la red. A continuación, se evalúa el modelo resultante y se decide si es válido o no. En caso de no serlo, se regresa a la etapa de diseño y se realizan las modificaciones correspondientes, y, en caso de ser válido se avanza a la siguiente etapa de pruebas donde se realizan las simulaciones del modelo. Finalmente, se analizan los resultados obtenidos a partir de las simulaciones, considerando los parámetros de desempeño de probabilidad de bloqueo de red<sup>15</sup> y retardo extremo a extremo de ráfaga<sup>16</sup>.

### 4.2. METODOLOGÍA DE SIMULACIÓN

Para llevar a cabo las simulaciones dentro de la herramienta IDE OMNeT++ se llevan a cabo los siguientes pasos ordenadamente:

- **Definición de escenarios de simulación:** Se define la configuración de cada escenario de simulación con base en el modelo de red requerido, los algoritmos requeridos, y los parámetros a modificar para realizar las pruebas.
- **Construcción del modelo de red de prueba:** Se realiza la construcción del modelo de la red NSFNeT OBS/SDEON mediante la implementación de módulos en la herramienta de simulación seleccionada.
- **Diseño de algoritmos a implementar en el modelo de red:** Se diseña el algoritmo clásico basado en SCC y el algoritmo cognitivo basado en DL para resolver el problema de RSA para ser integrados posteriormente en la red modelada en la herramienta de simulación seleccionada.
- **Pruebas de red con algoritmo clásico basado en SCC:** Se integra el algoritmo clásico basado en SCC en el modelo de red y se realizan las pruebas considerando los escenarios de simulación planteados
- **Pruebas de red con algoritmo cognitivo basado en DL:** Se integra el algoritmo cognitivo basado en DL en el modelo de red y se realizan las pruebas considerando los escenarios de simulación planteado

---

<sup>15</sup> relación entre el número total de ráfagas perdidas y el número total de ráfagas generadas en la red.

<sup>16</sup> tiempo que tarda la ráfaga desde que es ensamblada en el nodo de origen hasta que llega al desensamblador en el nodo de destino



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

- **Análisis y comparación de resultados:** Se analizan y comparan los resultados de la red con cada algoritmo integrado en función de los parámetros de desempeño de probabilidad de bloqueo de red y retardo extremo a extremo de ráfaga.
- **Conclusiones:** Se obtienen las conclusiones con base en los resultados obtenidos a partir de las pruebas [48].

La Figura 4.2 muestra un diagrama de flujo de la metodología de simulación empleada.

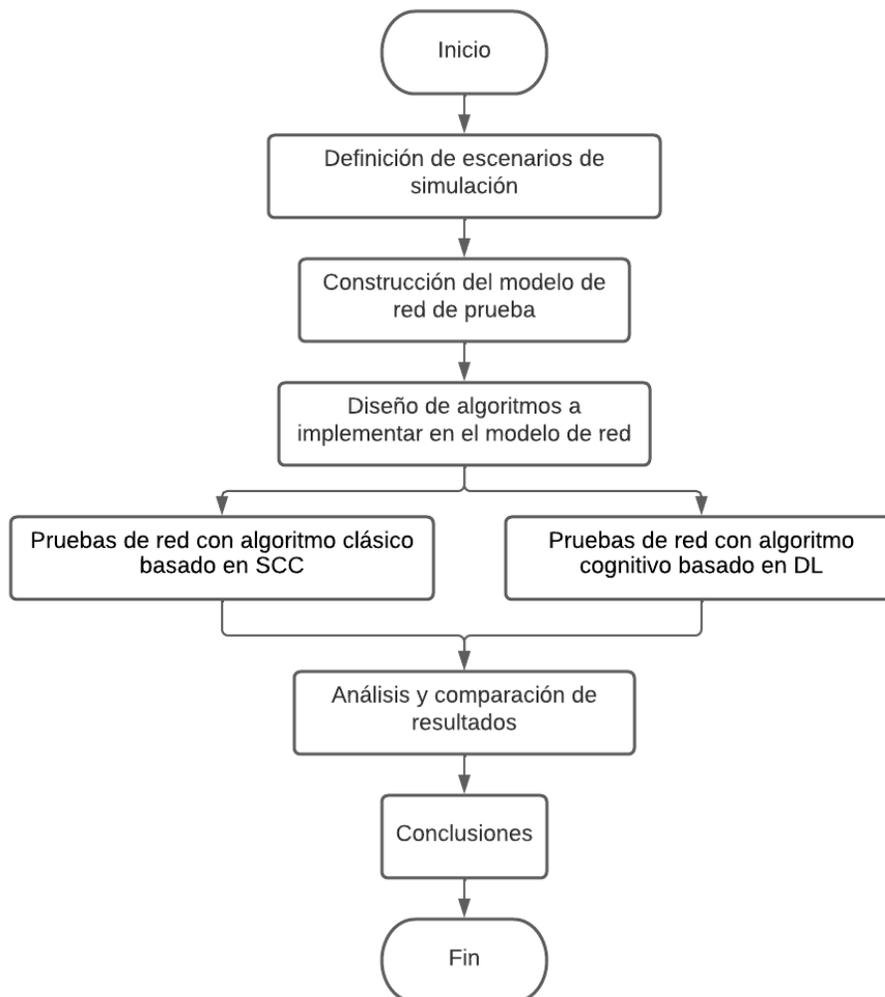


Figura 4.2. Diagrama de Flujo de Metodología de Simulación.



### 4.3. Caracterización de la red OBS

Se realiza ahora, la caracterización de la red empleando la topología NSFNeT. La red NSFNeT es una red WAN ampliamente usada para el estudio e investigación en el campo de las telecomunicaciones, la cual está distribuida a lo largo de catorce ciudades de Estados Unidos de América permitiendo una comunicación entre ellas mediante una topología irregular. Consta de 14 nodos con 21 enlaces bidireccionales con un número de conexiones variables por cada nodo. En la Figura 4.3 se muestra esta topología con cada uno de los nodos y su respectiva distancia en kilómetros entre ellos [63].

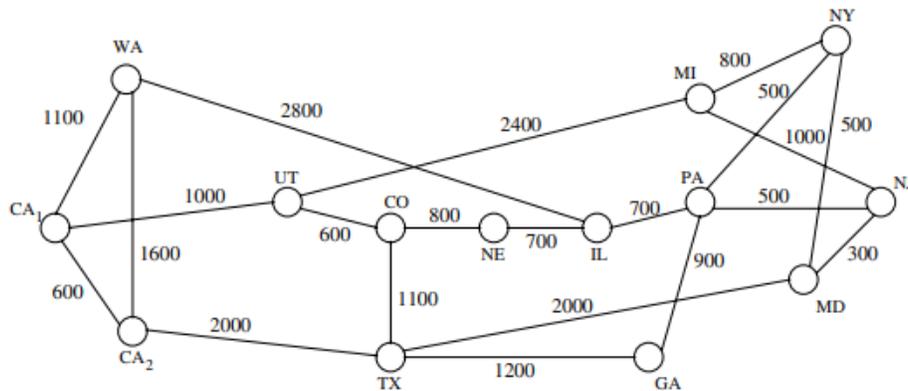


Figura 4.3. Topología NSFNeT con distancias aproximadas entre nodos. Tomada de [53]

Cada uno de los catorce nodos mostrados en la Figura 4.3 es llamado nodo de frontera y adicionalmente se diseña un nodo de control para las funciones de gestión y control centralizado de la red OBS/SDEON.

### 4.4. SELECCIÓN DE HERRAMIENTA DE SIMULACIÓN

De acuerdo con los requerimientos de la red, se realiza la selección de una herramienta *software* mediante la cual sea posible la creación y caracterización del modelo de simulación de la red SDEON basada en OBS utilizando la topología NSFNeT, permitiendo también, implementar en ésta los algoritmos desarrollados y realizar las respectivas pruebas para el análisis de desempeño en términos de la probabilidad de bloqueo y el retardo extremo a extremo.

A continuación, se presentan algunas opciones de herramientas de simulación más utilizadas dentro del área de redes de telecomunicaciones y su posterior



comparación con el fin de seleccionar la que más se adecúe para el desarrollo de este trabajo de grado.

#### **4.4.1. NS-2 (Network Simulator 2)**

Es una de las herramientas gratuitas de simulación de eventos discretos más utilizadas por la comunidad docente e investigadores. Está disponible para múltiples plataformas y brinda soporte para la simulación de todo tipo de redes. Este simulador, escrito en parte en C++, permite la configuración de una gran cantidad de parámetros como la topología de red, la pila de protocolos y parámetros específicos de cada protocolo, con el fin de evaluar diferentes tipos de tráfico de red [49]. Como limitaciones se tiene el uso de *scripts* lo cual hace que se requiera un amplio conocimiento previo para el uso de la herramienta, y también, la ausencia de una buena interfaz gráfica para la visualización de las simulaciones.

#### **4.4.2. NS-3 (Network Simulator 3)**

NS-3 es un simulador de red gratuito de eventos discretos orientado principalmente a la investigación y el uso educacional. Uno de los objetivos fundamentales en el diseño de este simulador fue permitir el desarrollo de modelos de simulación más realistas y de alto desempeño, habilitando el uso de esta herramienta como emulador. Esta versión está escrita totalmente en C++ y cuenta con una interfaz gráfica para visualizar las simulaciones [50] [51].

#### **4.4.3. NC-TUNS (National Chiao Tung University Network Simulator)**

NCTUNS es un simulador y emulador de redes de código abierto, extensible y de alta fidelidad a través del cual es posible simular protocolos usados en redes IP cableadas e inalámbricas. Cuenta con entorno gráfico integrado de fácil uso para que los usuarios puedan ejecutar las simulaciones y/o emulaciones. Es una herramienta potente y ampliamente usada en los ámbitos de investigación y educación ya que su tecnología central está basada en una novedosa metodología, consiguiendo así, grandes ventajas en comparación con simuladores de red tradicionales como NS-2 y OPNET. Su principal limitación radica en el hecho de que únicamente puede ser usado en plataformas Linux [52] [53].

#### **4.4.4. OPNET Modeler**

OPNET Modeler, actualmente llamada Riverbed, es una herramienta ampliamente usada en la industria para el modelado y la simulación de sistemas de comunicaciones. Permite diseñar y analizar redes, dispositivos, protocolos y



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

aplicaciones, ofreciendo gran escalabilidad y versatilidad. Posee una interfaz gráfica intuitiva y de alto nivel lo que hace que sea ampliamente usado a nivel académico, sin embargo, requiere de una licencia comercial y un amplio conocimiento previo de redes y del lenguaje C++ para su uso [54] [55].

### 4.4.5. OMNeT++

Es un simulador modular de eventos discretos orientado a objetos con licencia gratuita pública académica, destinado principalmente a simular redes informáticas y otros sistemas distribuidos. Es completamente programable y modular, fue diseñado desde cero lo cual permite el modelado de redes muy grandes siendo construidas a partir de modelos reutilizables. Es usado habitualmente para el modelado del tráfico en las redes de telecomunicaciones, protocolos, sistemas distribuidos, y, en general, cualquier sistema que pueda ser simulado con eventos discretos. Esta herramienta cuenta con versiones tanto para sistemas operativos basados en Unix como para Windows, MacOS, Docker, entre otros.

Un modelo de OMNeT++ está constituido de módulos que se comunican a través de mensajes y pueden tener parámetros definidos. Los módulos activos son llamados módulos simples y están escritos en C++. Estos módulos simples pueden ser agrupados en módulos más grandes llamados módulos compuestos y así sucesivamente, sin tener un límite en los niveles de jerarquía. Los mensajes pueden ser enviados directamente al módulo de destino o a través de compuertas o *gates*, las cuales son las interfaces de salida o entrada de un módulo y pueden estar vinculadas a una conexión. Estas conexiones o canales pueden tener propiedades específicas como retardos de propagación, tasas de datos, tasas de error de bits, entre otros [56].

Para el desarrollo de modelos de simulación se utilizan dos tipos de lenguaje: el primero de ellos llamado NED es utilizado para implementar la parte gráfica de OMNeT++ y describir la estructura, mientras que, mediante el lenguaje C++ se desarrolla la parte lógica del modelo [57].

Adicionalmente, OMNeT++ ofrece un IDE basado en Eclipse, cuenta con una interfaz gráfica de usuario, extensiones para simulación en tiempo real, emulación de red, integración de base de datos, entre otras funciones [58].

### 4.4.6. Comparación de herramientas

En la Tabla 4.1 se realiza una comparación de las herramientas anteriormente descritas.



**ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING**

Tabla 4.1. Comparación de herramientas de simulación. Adaptada de [46].

	<b>NS-2</b>	<b>NS-3</b>	<b>NC-TUNS</b>	<b>OPNET Modeler</b>	<b>OMNET++</b>
<b>Uso investigativo</b>	Alto	Medio	Alto	Alto	Alto
<b>Tipo de Licencia</b>	Libre	Libre	Libre	Comercial	Libre
<b>Curva de aprendizaje</b>	Alto	Alto	Alto	Alto	Alto
<b>Tipo de plataforma</b>	Windows, Mac, Unix	Windows, Mac, Unix	Linux	Windows, Unix	Windows, Unix
<b>Calidad Interfaz Gráfica</b>	Bajo	Medio	Alto	Alto	Medio
<b>Gráfica de resultados</b>	No tiene	Medio	Aceptable	Buena	Aceptable
<b>Tecnologías de nivel 2 y 3 soportadas</b>	Alto	Medio	Alto	Alto	Alto
<b>Tráfico que permite modelar</b>	Alto	Medio	Alto	Alto	Medio

De acuerdo con los datos presentados en la Tabla 4.1, se selecciona la herramienta IDE OMNeT++ para el desarrollo del trabajo de grado, al ser una de las más adecuadas gracias a la gran cantidad de características que ofrece. Como se evidencia en la Tabla 4.1, esta herramienta tiene un alto uso investigativo, es de licencia libre, es soportada tanto en Windows como en Unix, su interfaz gráfica es de nivel medio lo que facilita el uso de la herramienta por parte del usuario, posibilita la generación de datos estadísticos, permite la implementación de gran cantidad de tecnologías y protocolos de red y permite, también, el modelado y simulación de tráfico de tamaño medio.

Debido a que OMNeT++ permite la generación de datos estadísticos, pero no cuenta con la capacidad para procesarlos y presentarlos de manera adecuada, se hace necesario el uso de otra herramienta mediante la cual se pueda realizar las gráficas de los datos resultantes y así, poder analizar apropiadamente el



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

comportamiento de cada algoritmo. Se describen a continuación algunas de las herramientas más utilizadas para este propósito:

- **MATLAB:**

Es una plataforma de programación y cálculo numérico de alto rendimiento utilizada para el análisis de datos, el desarrollo de algoritmos y la creación de modelos. Permite visualizar datos e identificar patrones y tendencias a partir de las gráficas integradas en la herramienta, logrando representar estos datos de la manera más adecuada [59] [60].

- **ORIGINPRO:**

Es una herramienta *software* potente de análisis de datos y generación de gráficos de alta calidad utilizada por científicos e ingenieros. Ofrece una interfaz fácil de usar permitiendo realizar una personalización avanzada y automatización en tareas de importación, análisis, generación de gráficos e informes de datos a medida que el usuario se familiariza con la aplicación. Los gráficos y resultados pueden actualizarse automáticamente al modificar los datos o parámetros, lo cual permite realizar tareas repetitivas u operaciones por lotes desde la interfaz de usuario sin necesidad de programación [61].

Considerando las características ofrecidas por cada herramienta, se escoge OriginPro para la visualización de los resultados debido a su facilidad de uso y a la posibilidad de generar gráficos de alta calidad.

### 4.5. IMPLEMENTACIÓN DE RED

El modelo de simulación de la red OBS diseñado dentro de la herramienta IDE OMNeT++ se conforma principalmente de dos nodos: el nodo de frontera (el cual se replica 14 veces siguiendo la topología NSFNeT) y el nodo de control centralizado. Cada uno de los nodos de frontera de la red está conformado por varios módulos que permiten la conformación, transmisión, recepción de ráfagas y otras funciones adicionales. Por otra parte, en el nodo de control centralizado se crean los módulos necesarios para recibir las peticiones y devolver, de acuerdo con el algoritmo incluido, la ruta y el espectro asignado para la conexión.

Se describen, a continuación, cada uno de los módulos implementados dentro de la herramienta IDE OMNeT++ para el modelado y caracterización de la red NSFNeT OBS/SDEON.



#### 4.5.1. Módulo Fuente-Destino

Este nodo se representa a través de un módulo compuesto el cual tiene dos módulos simples: el nodo fuente (*source*) y el nodo destino (*sink*) como se muestra en la Figura 4.4. El nodo fuente es el encargado de generar y transmitir los paquetes ópticos a través de la red cada cierto tiempo predeterminado. A cada uno de los paquetes les asigna de forma aleatoria una dirección de destino y prioridad. En este módulo están definidos los parámetros de dirección de origen, dirección de destino, tamaño del paquete (en bytes) e intervalo entre generación de paquetes. Las variables para estos parámetros son declaradas en el archivo NED y se les asigna un valor en el archivo de inicialización omnet.ini del proyecto.

Por su parte, el nodo destino se encarga de recibir los paquetes correspondientes después de ser transmitidos por la red y, además, calcula los valores del número de paquetes perdidos y el retardo extremo a extremo de paquetes.

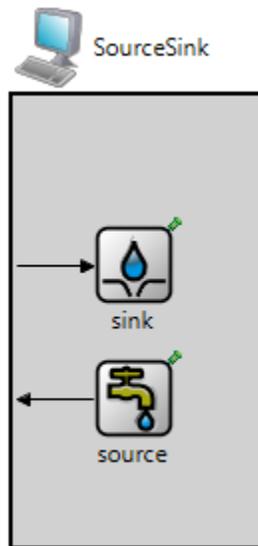


Figura 4.4. Módulo Fuente/Destino implementado en OMNeT++.

#### 4.5.2. Módulo de Nodo Frontera

En este nodo se realizan las funciones de ensamble y desensamble de las ráfagas de datos. Está compuesto por un nodo de ingreso (*Assembler*) y un nodo de egreso (*Disassembler*) como se muestra en el esquema de la Figura 4.5 el nodo de ingreso es el encargado de recibir los paquetes provenientes del nodo fuente (*source*), clasificarlos de acuerdo con ciertas condiciones, ensamblarlos en ráfagas y enviar las ráfagas por la red. Por otra parte, en el nodo de egreso se realiza el proceso de



ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

desensamble de la ráfaga y posteriormente se entregan los paquetes al nodo destino (*sink*) correspondiente.

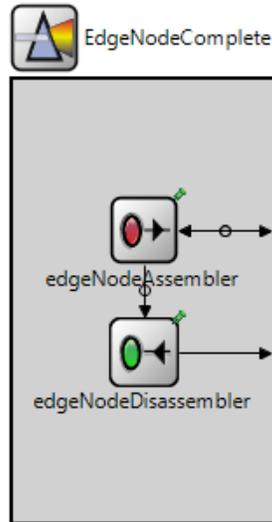


Figura 4.5. Módulo de Nodo Frontera implementado en OMNeT++.

#### 4.5.2.1. Módulo de Nodo Frontera de Ingreso

Este nodo es el encargado de realizar el ensamble y envío de la ráfaga de datos. Para llevar a cabo este proceso, cuenta con tres submódulos: un clasificador (*classifier*), un conformador de ráfagas (*burstifier*) y un planificador de envío (*sender*). El esquema de este nodo se muestra en la Figura 4.6.

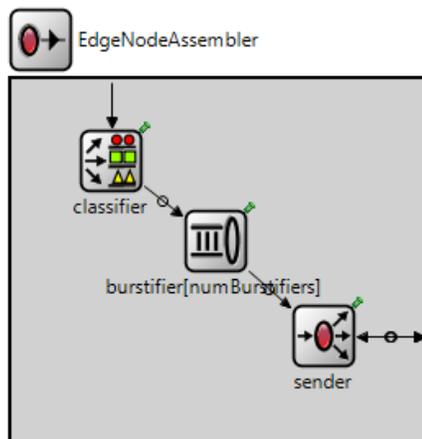


Figura 4.6. Módulo de Nodo Frontera de Ingreso implementado en OMNeT++.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

El clasificador tiene como función recibir los paquetes provenientes del nodo fuente y clasificarlos de acuerdo con unas reglas dadas, en este caso, según su dirección de destino y prioridad para transmitirlos luego al conformador de ráfagas (*burstifier*).

El conformador de ráfagas construye las ráfagas con los paquetes que tienen una misma prioridad y dirección de destino. Para realizar este proceso se tiene en cuenta el esquema de ensamble de ráfagas híbrido en donde se definen los valores establecidos para el tamaño límite de la ráfaga y el tiempo máximo de ensamble. Una vez se alcanza alguno de estos valores, se detiene el proceso, se forma la ráfaga y se envía hacia el planificador de envío (*sender*).

Después de que la ráfaga llega al planificador de envío, éste envía un mensaje de petición de *lightpath* al nodo de control con la información y requerimientos de la conexión para reservar los recursos necesarios. El nodo de control, mediante un algoritmo de RSA implementado en éste y un módulo con información de enrutamiento, responde a la solicitud con un mensaje con la ruta y el espectro asignados para el establecimiento de la conexión. El planificador recibe este mensaje y envía la ráfaga a través de una ruta dedicada hasta el siguiente nodo.

### 4.5.2.2. Módulo de Nodo Frontera de Egreso

Este nodo tiene la función de recibir las ráfagas de datos que provienen de otros nodos de la red, realiza el desensamble obteniendo los paquetes individuales y los entrega a su respectivo nodo de destino. Aquí se realiza también los cálculos del número de ráfagas perdidas y el retardo extremo a extremo de ráfagas.

### 4.5.3. Módulo de Nodo de Control

La implementación del nodo de control de la red se realiza con base en la adaptación de un módulo de un simulador centralizado de redes ópticas con encaminamiento por longitud de onda (WRON, *Wavelength-Routed Optical Network*) implementado para un trabajo de grado en [62]. Se realizan algunas modificaciones al nodo de control implementado en OMNeT++ de este simulador para que pueda operar con ranuras o *slots* de frecuencia en lugar de longitudes de onda y, además, se pueda realizar una asignación flexible del espectro disponible de acuerdo con los requerimientos de las conexiones. La Figura 4.7 muestra el esquema del nodo de control.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING



Figura 4.7. Módulo de Nodo de control implementado en OMNeT++

El nodo de control mostrado en la Figura 4.7 es el módulo que contiene toda la inteligencia de la red. En este se implementan los métodos para el enrutamiento de los caminos de luz o *lightpaths* y los métodos para la asignación del espectro demandado por los usuarios de la red. Está compuesto por un conjunto de módulos simples que se describen a continuación [62].

### 4.5.3.1. Módulo de Estado

Este módulo es el encargado de almacenar toda la información de la red. Los demás módulos de la red pueden acceder a él para conocer el estado de los recursos y actualizarlos. En este módulo se realiza también el cálculo de rutas entre un origen y un destino.

Adicionalmente, en este módulo se encuentra definida una matriz tridimensional en donde se almacena el estado de todos los enlaces de la red. Las dimensiones de esta matriz son de  $M \times N \times 2$ , donde  $M$  es el número de enlaces de la red óptica,  $N$  el número de canales ocupados en el espectro de un enlace  $M$  y el 2 representa las dos columnas de la frecuencia inicial y frecuencia final que abarca la ocupación del canal  $N$  en el espectro del enlace  $M$  [62].

### 4.5.3.2. Módulo WRON Dinámica

Dentro de este módulo se encuentra un submódulo llamado *Dynamic\_WRON\_Control* en donde se implementan todos los métodos para permitir el establecimiento, la liberación o el bloqueo de peticiones de *lightpath*. Como se muestra en la Figura 4.8, este submódulo está compuesto por otros dos módulos los cuales se describen a continuación [62].



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

- **Módulo de Control RWA Dinámico:**

En este módulo se realizan las funciones de establecimiento, liberación o bloqueo de las solicitudes de *lightpath*. Para llevar a cabo estas funciones, este módulo mantiene una comunicación continua y un intercambio de datos con los módulos *Estado* y *Metodo\_RWA\_Dinamico* [62].

- **Módulo Método RWA Dinámico:**

Este módulo es el encargado del problema de RSA, para lo cual, selecciona el camino o ruta adecuada a partir de un algoritmo implementado y ejecuta los métodos implementados para la asignación de espectro [62].

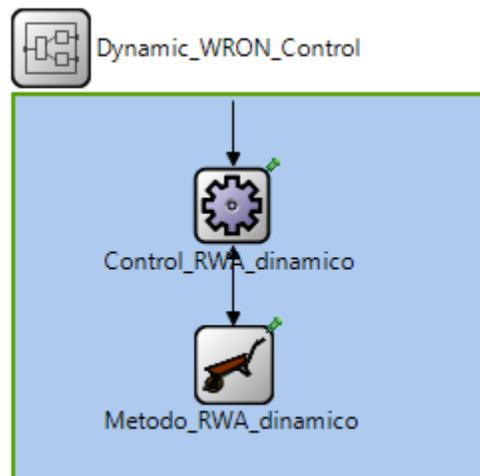


Figura 4.8. Módulo *Dynamic\_WRON\_Control* implementado en OMNeT++.

Se modela la red NSFNet con cada uno de sus componentes en la herramienta IDE OMNeT++. Cada uno de los módulos de los nodos frontera y de los nodos fuente/destino creado previamente se agrupa en un mismo módulo compuesto llamado *edgeCore*, cuyas funciones son las de generación y recepción de ráfagas, este último se replica catorce veces ubicando cada uno en la posición correspondiente de cada ciudad en el mapa de Estados Unidos, además de un único nodo de control con cada uno de sus módulos internos. De esta manera se obtiene el mapa de la red mostrado en la Figura 4.9.

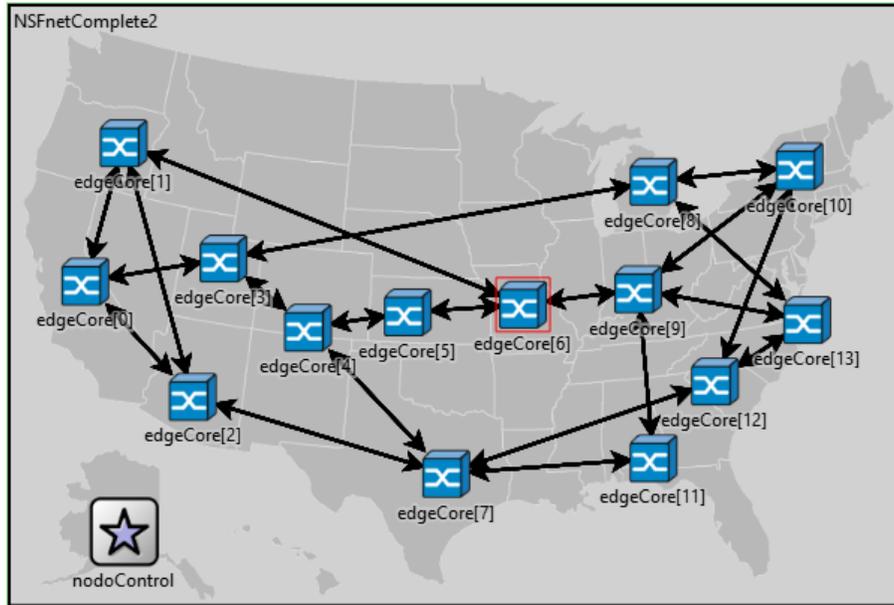


Figura 4.9. Red OBS/SDEON basada en la topología NSFNet implementada en OMNeT++.

## 4.6. DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS

Como se mencionó anteriormente, para el desarrollo del presente trabajo de grado se realiza el diseño e implementación de dos algoritmos. En primer lugar, se tiene un algoritmo clásico basado en SCC, el cual se basa en el algoritmo de Dijkstra, o algoritmo de  $k$  caminos más cortos (KSP, *K-Shortest Paths*), para resolver el problema de enrutamiento y, en lo referente a la asignación de espectro, se usa el algoritmo de primer ajuste o FF teniendo en cuenta la condición de restricción de continuidad. Asimismo, se diseña el algoritmo cognitivo basado en DL, el cual emplea igualmente el algoritmo de  $k$  caminos más cortos para el enrutamiento y se basa en los pasos del ciclo cognitivo y en los conceptos y características de DRL para resolver el problema de asignación de espectro utilizando FF. Estos dos algoritmos se integran de forma individual en el nodo de control con el fin de analizar el comportamiento de la red NSFNet OBS/SDEON con cada uno de ellos.

### 4.6.1. Algoritmo clásico basado en restricción de continuidad de espectro

El diseño del algoritmo clásico basado en SCC es realizado en dos etapas, se emplea el algoritmo de Dijkstra de camino más corto para el enrutamiento y para la asignación de espectro se emplea un algoritmo de FF modificado que tiene en



cuenta la condición de restricción de continuidad de espectro. Se describen a continuación los dos algoritmos.

#### 4.6.1.1. Algoritmo Dijkstra

Es el algoritmo de caminos mínimos o de camino más corto más conocido en la literatura. Este funciona mediante el seguimiento de la ruta más corta descubierta en cierto momento hasta un nodo específico, iniciando con una longitud de ruta con valor cero para el nodo origen. Si la ruta hacia el nodo vecino es más corta que cualquier ruta descubierta previamente a ese nodo, se actualiza entonces la ruta hacia ese vecino. Este algoritmo toma la decisión óptima en cada paso sin observar el resultado final, de esta manera se garantiza que se encontrará la solución óptima, es decir, el camino más corto desde el nodo de origen hasta el nodo de destino [64].

El algoritmo de Dijkstra, basándose en la teoría de grafos, sigue unos pasos para encontrar el camino más corto entre dos nodos de un grafo, los cuales son descritos a continuación.

Teniendo un grafo dirigido ponderado de  $N$  nodos no aislados, siendo  $s$  el nodo inicial y un vector  $D$  de tamaño  $N-1$  que almacena las distancias desde  $s$  hasta el resto de los nodos, se tienen los siguientes pasos:

1. Se inicializan todas las distancias en  $D$  con un valor infinito relativo (esas distancias son desconocidas al inicio), salvo la distancia de  $s$  al mismo  $s$  la cual es inicializada en cero.
2. Sea  $a = s$ , donde  $a$  hace referencia al nodo actual.
3. Se recorren todos los nodos adyacentes a  $a$ , excepto los nodos marcados, los cuales se representarán como  $vi$ .
4. Si la distancia desde  $s$  hasta  $vi$  almacenada en  $D$  es mayor que la distancia desde  $s$  hasta  $a$ , sumada a la distancia desde  $a$  hasta  $vi$ , esta se sustituye con la segunda nombrada, esto es:  $Di > Da + d(a, vi)$  entonces  $Di = Da + d(a, vi)$
5. Se marca como completo el nodo  $a$ .
6. Se toma como próximo nodo actual el de menor valor en  $D$  (puede hacerse almacenando los valores en una cola de prioridad) y se regresa al paso 3 mientras existan nodos no marcados.
7. Una vez termine el algoritmo, el vector  $D$  estará completamente lleno [65] [66].



# ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

En la Figura 4.10 se muestra el diagrama de flujo para el algoritmo de Dijkstra en donde se describe gráficamente el proceso paso a paso.

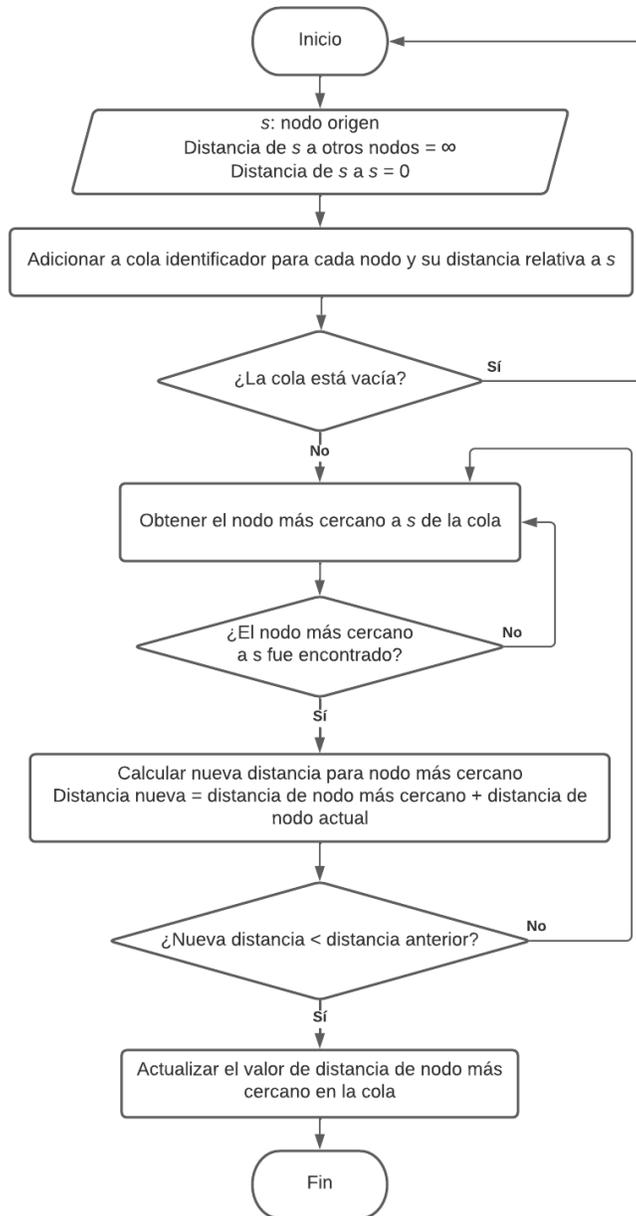


Figura 4.10. Diagrama de flujo de algoritmo Dijkstra.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

A continuación, en la Figura 4.11 se presenta el pseudocódigo del algoritmo de Dijkstra, el cual se implementa en la herramienta IDE OMNeT++. Adicionalmente, en la Figura 4.12 se presenta el algoritmo de Dijkstra en C++ utilizado para hallar los caminos más cortos desde un nodo origen hasta un nodo destino, esta función hace parte de una clase llamada `cTopology` perteneciente a una librería de OMNeT++.

```
DIJKSTRA (Grafo G, nodo_fuente s)
  Para u ∈ V[G] hacer
    distancia[u] = INFINITO
    padre[u] = NULL
  distancia[s] = 0
  adicionar (cola, (s,distancia[s]))
  mientras que cola no es vacía hacer
    u = extraer_minimo(cola)
    Para todos v ∈ adyacencia[u] hacer
      si distancia[v] > distancia[u] + peso (u, v) hacer
        distancia[v] = distancia[u] + peso (u, v)
        padre[v] = u
        adicionar(cola,(v,distancia[v]))
```

Figura 4.11. Pseudocódigo para el algoritmo Dijkstra. Tomada de [65]



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
void cTopology::calculateUnweightedSingleShortestPathsTo(Node * _target) {
    // multiple paths not supported :-{
    if (!_target) throw cRuntimeError(this, "..ShortestPathTo(): target node is NULL");
    target = _target;
    for (int i = 0; i < num_nodes; i++) {
        nodev[i].known = false; // not really needed for unweighted
        nodev[i].dist = INFINITY;
        nodev[i].out_path = NULL;
    }
    target -> dist = 0;
    std::deque < Node * > q;
    q.push_back(target);
    while (!q.empty()) {
        Node * v = q.front();
        q.pop_front();
        // for each w adjacent to v...
        for (int i = 0; i < v -> num_in_links; i++) {
            if (!(v -> in_links[i] -> enabl)) continue;
            Node * w = v -> in_links[i] -> src_node;
            if (!w -> enabl) continue;
            if (w -> dist == INFINITY) {
                w -> dist = v -> dist + 1;
                w -> out_path = v -> in_links[i];
                q.push_back(w);
            }
        }
    }
}
```

Figura 4.12. Código para Algoritmo Dijkstra en lenguaje C++.

### 4.6.1.2. Algoritmo *First-Fit*

Es uno de los algoritmos mayormente utilizados para resolver el problema de asignación de espectro gracias su baja complejidad y a la baja probabilidad de bloqueo que proporciona. Para la toma de decisiones, el algoritmo se basa de una matriz de capacidad conjunta o MCC en donde se almacenan los *slots* de frecuencia que se encuentran ocupados. A continuación, se describen los pasos seguidos por este algoritmo para llevar a cabo la asignación de espectro para las nuevas peticiones de conexión entrantes.

1. En primer lugar, se comprueba si la MCC está vacía, y en caso de ser así, se asignan los primeros *slots* (de menor índice) del espectro, en función de la capacidad demandada y del tamaño del *slot*.
2. Si no se cumple la condición anterior, se comprueba si los *slots* necesarios se pueden asignar entre la frecuencia mínima del espectro y el primer *slot* ocupado.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

3. Si el paso 2 no puede realizarse, se busca si existe suficiente ancho de banda como para asignar los *slots* necesarios entre la frecuencia final de un *slot* ya ocupado y la frecuencia inicial del siguiente *slot* ya ocupado.

Este paso se realizará con la ayuda de un bucle, para así revisar todos los huecos disponibles en el espectro que tengan *slots* ocupados tanto a su izquierda como a su derecha en el espectro.

4. Si no se cumple ninguna de las anteriores condiciones, se busca entre la frecuencia final del último *slot* ocupado y la frecuencia máxima del espectro [67].

5. Si se cumple alguno de los anteriores pasos, se asignan los *slots* demandados por la conexión y se termina el proceso. En caso contrario, la petición de conexión es bloqueada y el proceso finaliza.

En la Figura 4.13, se muestra el diagrama de flujo del algoritmo donde se representan cada uno de los pasos mencionados.



ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

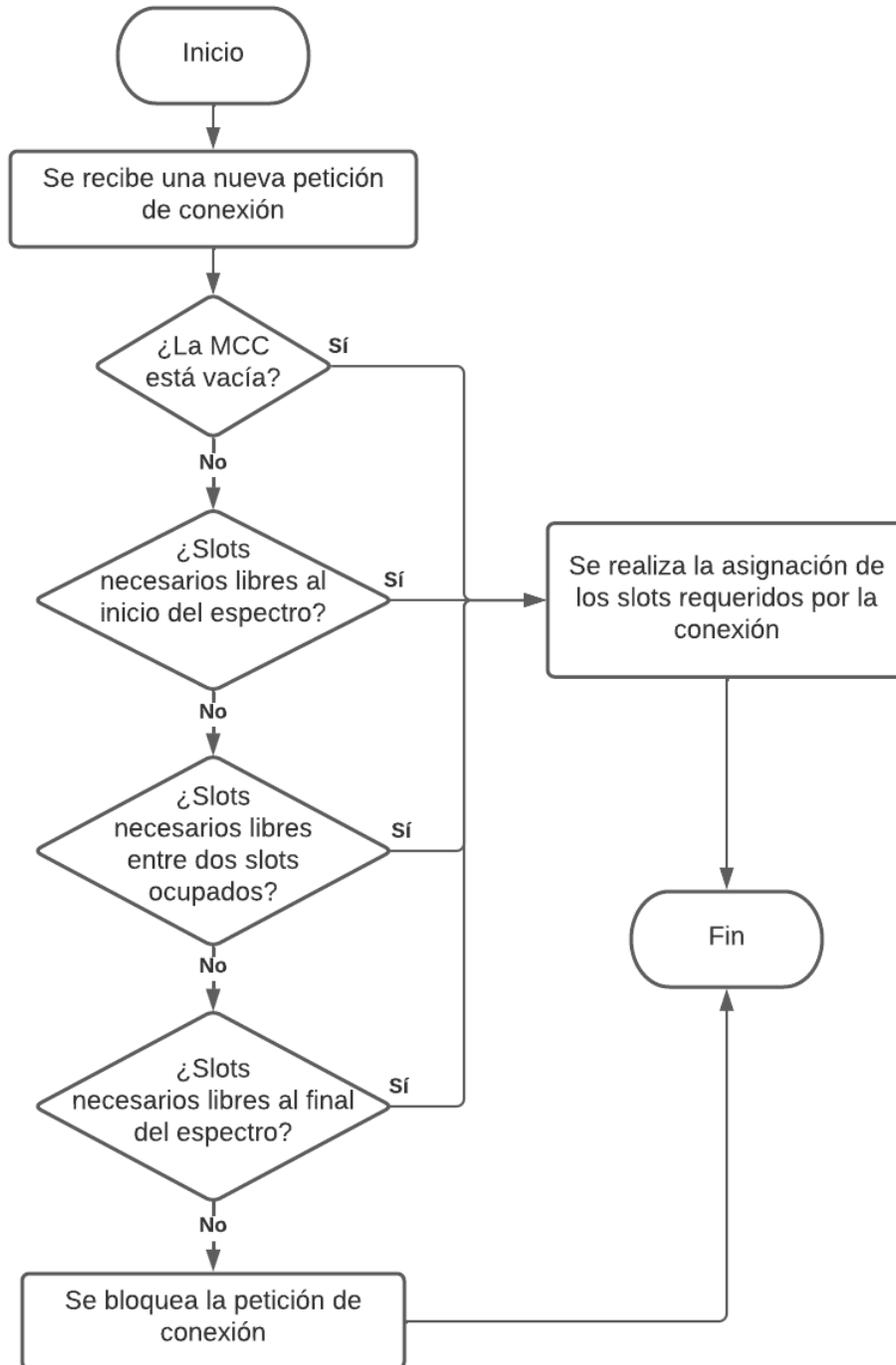


Figura 4.13. Diagrama de flujo de algoritmo First-Fit. Adaptada de [67]



## 4.6.2. Algoritmo cognitivo basado en *Deep Learning*

El diseño del algoritmo cognitivo basado en DL, el cual se basa de una técnica conocida como aprendizaje por refuerzo profundo o DRL para realizar las funciones de enrutamiento y asignación de espectro se realiza también en dos etapas. Para la parte de enrutamiento, se implementa un algoritmo de DRL basado en KSP el cual se basa en el algoritmo de Dijkstra descrito anteriormente. Este algoritmo calcula previamente una tabla de enrutamiento que consiste en una lista ordenada de las K rutas entre todos los pares de nodos de la red. En el momento en que llega una petición de conexión, se obtienen las K rutas más cortas de la tabla de enrutamiento a partir del nodo origen y el nodo destino y se selecciona la ruta más corta la cual contenga el número de *slots* de frecuencia necesarios para asignar a la conexión. Si no existen los *slots* requeridos en ninguna de las K rutas se rechaza la petición de conexión y se produce un bloqueo.

En cuanto al problema de asignación de espectro, se hace uso del mismo algoritmo de primer ajuste o FF descrito en la sección anterior. Estos dos algoritmos son implementados dentro de un agente. A continuación, se describe con mayor detalle el diseño del algoritmo basado en DRL.

### 4.6.2.1. Diseño Del Algoritmo RSA Basado En DRL

Para el desarrollo del algoritmo RSA basado en DRL, se utiliza una librería de Python llamada RSA-RL, la cual contiene funciones, agentes, entornos y redes para RL ya implementados. De esta librería se importan los elementos necesarios para el diseño del algoritmo. En primer lugar, se define el número de peticiones a realizar y el entorno, en este último se especifica el tipo de red y el elemento que se encarga de hacer las peticiones con unos parámetros específicos [68]. En la Figura 4.14 se muestra la porción de código en Python que realiza este proceso.

Para la selección del algoritmo de DRL, se importa la librería PFRL, la cual contiene varios algoritmos de DRL ya implementados y se selecciona de éstos el algoritmo DDQN, explicado en el capítulo 2. Se procede ahora, a construir una clase con el modelo de la red neuronal profunda, el código en Python para esto se muestra en la Figura 4.15. Se especifica el *explore* (método para equilibrar la exploración y explotación) y *replay buffer* (búfer que almacena experiencias pasadas) y se construye el algoritmo DDQN.

A continuación, se construye el agente basándose en una clase proporcionada por la librería RSA-RL llamada *KSPDRLAgent*, la cual emplea una tabla de k rutas más cortas y se implementan los métodos para que el agente pueda realizar una acción



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

a partir de la observación del estado del entorno. El código para la creación de este agente se muestra en la Figura 4.16.

Finalmente, se entrena y se evalúa el agente DRL un número de veces determinado y a partir de esto es posible obtener los parámetros correspondientes de la probabilidad de bloqueo, la utilización promedio de *slot* y la recompensa total final. En la Figura 4.17 se muestra la porción de código en Python correspondiente para realizar la evaluación del agente.

```
# configuraciones
n_requests = 100
n_envs, seed = 5, 12

# construir red
net = SingleFiberNetwork("nsf", n_slot=100, is_weight=True)
## construir peticiones
requester = UniformRequester(
    net.n_nodes,
    avg_service_time=10,
    avg_request_arrival_rate=12)
## construir entorno
env = DeepRMSAEnv(net, requester)
```

Figura 4.14. Porción de código en Python para la definición de la red y el entorno.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
class DeepRMSAv1_DNN(torch.nn.Module):  
  
    def __init__(self, SLOT: int, ICH: int, K: int, n_edges: int):  
        super().__init__()  
        self.SLOT = SLOT  
        # Red neuronal Convolutacional  
        self.conv = nn.Sequential(*[  
            nn.Conv2d(ICH, 1, kernel_size=(1,1), stride=(1, 1)),  
            nn.ReLU(),  
            # 2 capas convolucionales con 16 filtros  
            nn.Conv2d(1, 16, kernel_size=(n_edges,1), stride=(1, 1)),  
            nn.ReLU(),  
            nn.Conv2d(16, 16, kernel_size=(1,1), stride=(1, 1)),  
            nn.ReLU(),  
            # 2 capas convolucionales profundas con 1 filtro  
            nn.ZeroPad2d((1, 0, 0, 0)), # izq, der, arr, aba  
            nn.Conv2d(16, 16, kernel_size=(1,2), stride=(1, 1), groups=16),  
            nn.ReLU(),  
            nn.ZeroPad2d((1, 0, 0, 0)),  
            nn.Conv2d(16, 16, kernel_size=(1,2), stride=(1, 1), groups=16),  
            nn.ReLU(),  
        ])  
        # Capa Fully Connected  
        self.fc = nn.Sequential(*[  
            nn.Linear(SLOT*16, 128),  
            nn.ReLU(),  
            nn.Linear(128, 50),  
            nn.ReLU(),  
            nn.Linear(50, K),  
        ])  
  
    def forward(self, x):  
        h = x  
        h = self.conv(h)  
        h = h.view(-1, self.SLOT*16)  
        h = self.fc(h)  
        return pfrl.action_value.DiscreteActionValue(h)
```

Figura 4.15. Porción de código en Python para la creación de la red neuronal profunda.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
class DRLAgent(KSPDRLAgent):

    def preprocess(self, obs,s,d,bandwidth):
        """
        """
        net = obs.net
        source, destination, bandwidth= s,d,bandwidth
        # tabla de slots
        whole_slot = np.array(list(nx.get_edge_attributes(net.G, name="slot").values()))
        whole_slot = whole_slot.reshape(1, net.n_edges, net.n_slot).astype(np.float32)
        # mapa de fuente, destino, ancho de banda
        smap = np.ones_like(whole_slot) * vectorize(net.n_nodes, source)
        dmap = np.ones_like(whole_slot) * vectorize(net.n_nodes, destination)
        bmap = np.ones_like(whole_slot) * bandwidth
        # concatenar: (1, ICH, #edges, #slots)
        fvec = np.concatenate([whole_slot, smap, dmap, bmap], axis=0)
        return fvec.astype(np.float32, copy=False)

    def map_drlout_to_action(self, obs, out,s,d,bandwidth):
        net = obs.net
        s, d, bandwidth = s,d,bandwidth
        paths = self.path_table[sort_tuple((s,d))]
        # mapa rutas
        path=paths[out]
        #slots requeridos
        path_len = net.distance(path)
        n_req_slot = cal_slot(bandwidth, path_len)
        #First-Fit
        path_slot,te = net.path_slot(path)
        slot_index = SpectrumAssignment.first_fit(path_slot, n_req_slot)
        if (s>d):
            path=np.flip(path)
        if slot_index is None:
            return None
        else:
            return Action(path, slot_index, n_req_slot)
```

Figura 4.16. Porción de código en Python para la creación del agente DRL y sus métodos.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

```
def evaluation(envs, agent, s, d, bandwidth, n_requests):
    # iniciar simulación
    envs.reset()
    #
    batch_warming_up(envs, agent, n_requests=1000,s=s,d=d,bandwidth=bandwidth)
    # evaluación
    experiences = batch_evaluation(envs, agent, n_requests=n_requests,s=s,d=d,bandwidth=bandwidth)
    # cálculo de desempeño
    blocking_probs, avg_utils, total_rewards = batch_summary(experiences)
    #mostrar resultados
    for env_id, (blocking_prob, avg_util, total_reward) in enumerate(zip(blocking_probs, avg_utils, total_rewards)):
        print(f'[{env_id}-th ENV]Blocking Probability: {blocking_prob}')
        print(f'[{env_id}-th ENV]Avg. Slot-utilization: {avg_util}')
        print(f'[{env_id}-th ENV]Total Rewards: {total_reward}')
```

Figura 4.17. Porción de código en Python para la implementación del método para la evaluación del agente.

En la Figura 4.18 se muestra un diagrama de flujo con los pasos descritos para el desarrollo del algoritmo cognitivo DRL para el problema de RSA.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

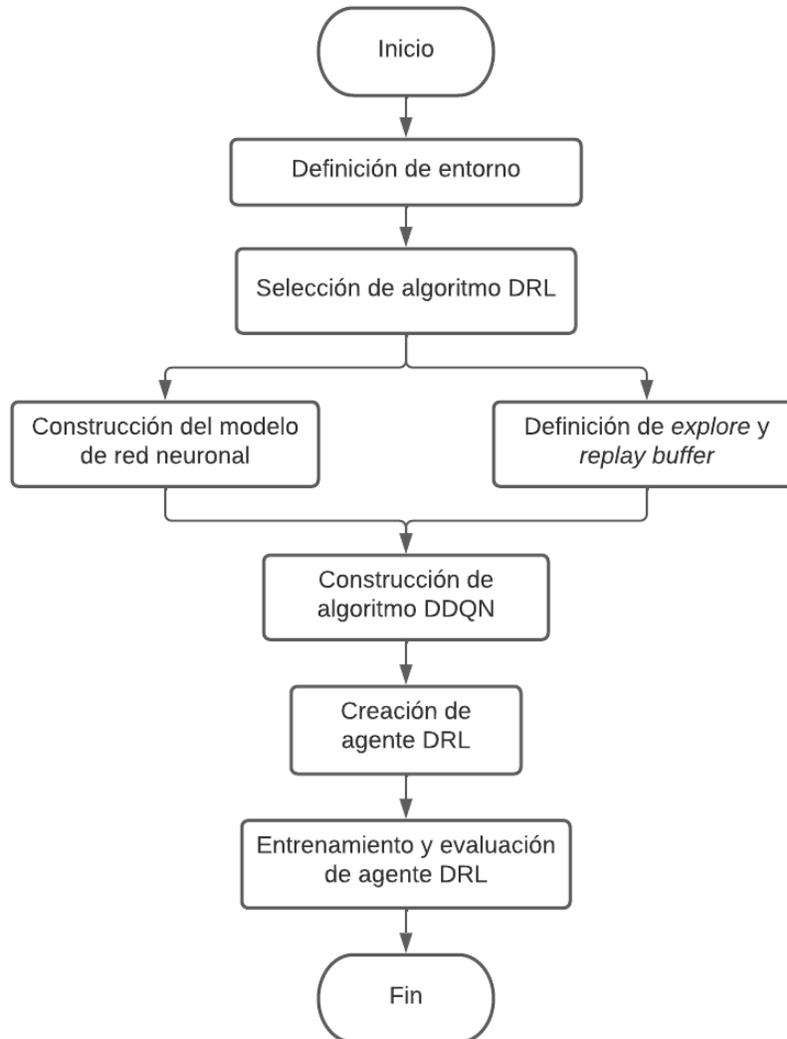


Figura 4.18. Diagrama de flujo para el desarrollo de algoritmo cognitivo basado en DRL.

Cabe resaltar que para la comunicación del algoritmo cognitivo basado en DL con la red en la herramienta IDE OMNeT++, se hizo uso de sockets, donde el algoritmo hace las veces de servidor y la red las veces de cliente. En el momento en que se realiza una nueva petición de conexión, se envía desde el nodo de origen al nodo de control la información necesaria para obtener los recursos para dicha conexión. El nodo de control a su vez, usando sockets para la comunicación, envía esta



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

información al algoritmo en Python (servidor) y espera a que este le responda devolviendo los recursos para la conexión. De esta manera se logra el intercambio de información entre entidades programadas en diferentes lenguajes.

Este algoritmo de DRL descrito tiene un enfoque cognitivo, ya que, a partir de la observación del entorno, es capaz de tomar una decisión, obtener una respuesta para evaluar la acción tomada y aprender de ella para tomar futuras decisiones. Este conjunto de pasos, hacen parte de un concepto conocido como ciclo cognitivo, el cual se describe a continuación.

### 4.6.2.2. CICLO COGNITIVO DEL ALGORITMO

En la actualidad existen diferentes métodos o ciclos cognitivos mediante los cuales se realizan funciones de razonamiento, aprendizaje y planeación, apoyándose de diferentes mecanismos tomados a partir de la IA. Estos métodos pueden ser incluidos en el nodo de una red OBS con el fin de realizar cambios tanto estructurales como funcionales, contribuyendo así a conseguir alguna mejora de desempeño del nodo o de toda la red [37].

En general, el ciclo cognitivo aplicado en redes está definido por seis procesos o estados: observación, orientación, planeación, aprendizaje, toma de decisiones y acción. En la Figura 4.19 se muestra un diagrama con las diferentes interacciones entre los componentes del ciclo cognitivo.

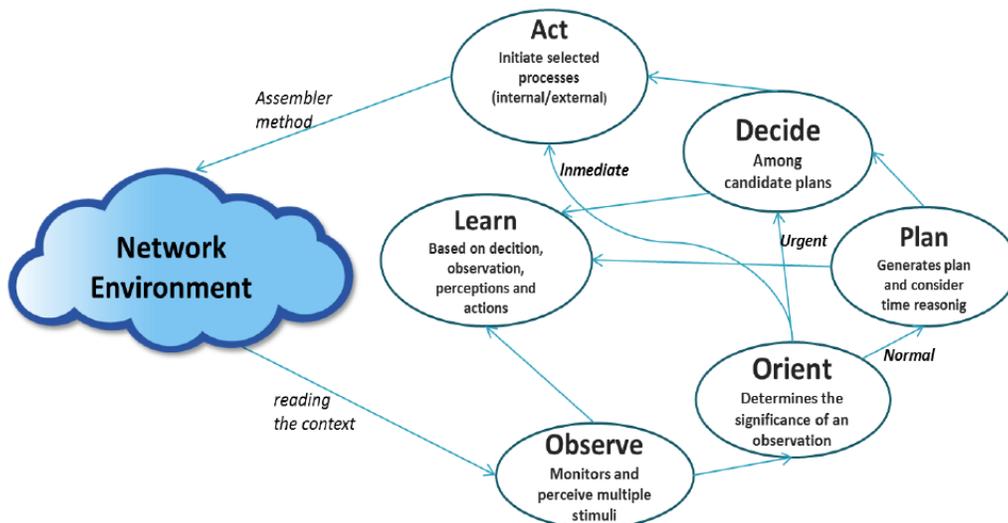


Figura 4.19. Diagrama de Ciclo Cognitivo. Tomada de [37]



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

Para el diseño del algoritmo basado en DRL, se toma como base uno de los principales modelos de ciclos cognitivo conocido como OODA (*Observe, Orient, Decide and Act*). Este modelo fue propuesto por un coronel para el entrenamiento en la fuerza aérea de Estados Unidos y ha sido adaptado para ser aplicado en diversos estudios e investigaciones de redes de telecomunicaciones ya que cuenta con estados esenciales bien definidos. El primer estado o estado de observación hace referencia a la recolección de datos por medio de los sentidos, en seguida, se tiene el estado de orientación en donde se realiza la construcción de una perspectiva del entorno a partir del análisis y síntesis de los datos recolectados. Como tercer estado está la decisión, en donde se determina la acción a tomar con base en la perspectiva del entorno creada, y finalmente, en el estado de acción se realiza la ejecución de la decisión. La Figura 4.20 muestra el modelo del ciclo cognitivo OODA [37].

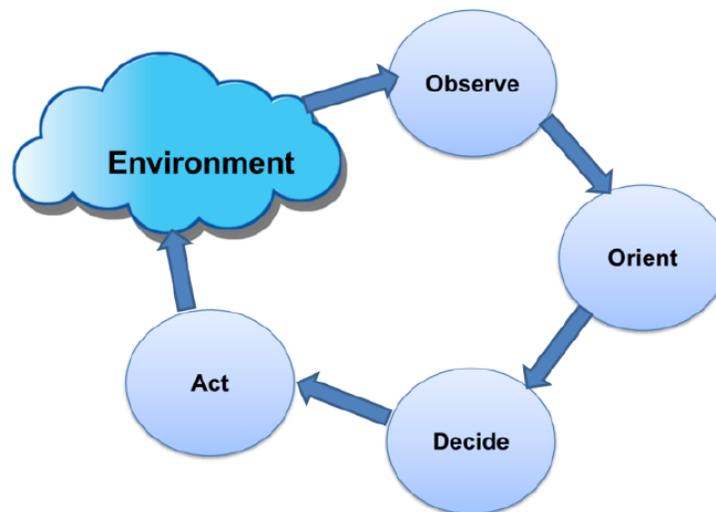


Figura 4.20. Modelo del Ciclo Cognitivo OODA. Tomada de [37]



## 5. ESCENARIOS DE SIMULACIÓN Y PRUEBAS

### 5.1. INTRODUCCIÓN

Para la definición de los escenarios de simulación, se establecen previamente las diferentes configuraciones de la red para cada uno de los casos y los aspectos que serán evaluados. En primer lugar, se incluirán independientemente en la red dos algoritmos diferentes para RSA, el algoritmo clásico basado en SCC y el algoritmo cognitivo basado en DL, representando estos, los dos principales casos de simulación. Se varía también, conformando los subcasos de los casos anteriores, la cantidad de espectro disponible, asignando valores de 100 y 200 GHz, esperando que con el valor más alto haya más posibilidad de transmitir información y por ende menos bloqueos en la red. Finalmente, se varía la velocidad de transmisión de enlace, con valores de 1,25 y 2,5 Gbps con el fin de evaluar la disposición de la red para transmitir información con cada uno de estos valores. Todos los casos y valores anteriores seleccionados se evalúan para diferentes valores de tráfico normalizado desde 0,1 a 1 y para diferentes longitudes de paquete: 1472kB, 100 MB y 1GB. Estas longitudes se definen con el fin de analizar y comparar las variaciones en el desempeño de la red que se obtienen cuando se manejan tamaños de ráfaga pequeños, medianos y grandes.

De acuerdo con lo anterior, se plantean los escenarios de simulación para el análisis del desempeño de la red OBS/SDEON basada en la topología NSFNeT en términos de la probabilidad de bloqueo y el retardo extremo a extremo de ráfaga. Para el cálculo de la probabilidad de bloqueo, se obtiene la relación entre la cantidad de ráfagas perdidas sobre la cantidad total de ráfagas generadas en la red, mientras que, para el retardo extremo a extremo, se considera el tiempo de ida y regreso cuando se realiza la petición al nodo de control sumado con el tiempo que tarda la ráfaga en transmitirse a lo largo de toda la ruta su y el tiempo de procesamiento de la ráfaga en cada uno de los nodos frontera y en el nodo de control. En las ecuaciones 4 y 5 se muestran las fórmulas utilizadas para calcular la probabilidad de bloqueo de ráfagas y el retardo extremo a extremo de ráfagas respectivamente.

$$\text{Probabilidad de bloqueo} = \frac{\text{Número de ráfagas perdidas}}{\text{Número de ráfagas generadas}} \quad (4)$$



**ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING**

$$\begin{aligned}
 & \textit{Retardo extremo a extremo} \\
 & = \textit{tiempo de petición} + \textit{tiempo de transmisión} \\
 & + \textit{tiempo de procesamiento} \qquad \qquad \qquad (5)
 \end{aligned}$$

Se presenta en la Tabla 5.1, el conjunto de casos y subcasos en donde se definen las diferentes configuraciones de los parámetros de simulación. Partiendo de la red NSFNeT OBS/SDEON, se configuran las características principales de acuerdo con cada caso, específicamente: el algoritmo implementado, el espectro disponible (o número de slots) y la velocidad de transmisión de enlace.

Tabla 5.1. Escenarios de simulación para pruebas.

<b>ESCENARIOS DE SIMULACIÓN</b>			
Topología de red NSFNeT			
<b>Casos de Simulación</b>			
<b>Caso 1</b>		<b>Caso 2</b>	
Red NSFNeT OBS/SDEON implementando algoritmo clásico basado en SCC		Red NSFNeT OBS/SDEON implementando algoritmo cognitivo basado en DL	
<b>Subcasos de Simulación</b>			
<b>Subcaso 1</b>		<b>Subcaso 2</b>	
Espectro de 100 GHZ (8 <i>slots</i> )		Espectro de 200 GHZ (16 <i>slots</i> )	
<b>Subcaso 1.1.</b>	<b>Subcaso 1.2.</b>	<b>Subcaso 2.1</b>	<b>Subcaso 2.2</b>
Velocidad de 1.25 Gbps	Velocidad de 2.5 Gbps	Velocidad de 1.25 Gbps	Velocidad de 2.5 Gbps

## 5.2. CASOS DE SIMULACIÓN

Como se muestra en la Tabla 5.1, se plantean dos casos principales de simulación para la ejecución de las pruebas y el posterior análisis de desempeño de los algoritmos diseñados para este trabajo de grado, a fin de estudiar y comparar el comportamiento de la red con cada uno de estos algoritmos integrados.

### 5.2.1. Caso 1: Red NSFNeT OBS/SDEON implementando algoritmo clásico basado en SCC

Para este caso, se caracteriza una red OBS/SDEON basada en la topología NSFNeT incluyendo el algoritmo clásico basado en SCC para el problema de RSA en el nodo de control.



## 5.2.2 Caso 2: Red NSFNeT OBS/SDEON implementando algoritmo cognitivo basado en DL

Para este caso, se caracteriza una red OBS/SDEON basada en la topología NSFNeT incluyendo el algoritmo cognitivo basado en DL para el problema de RSA.

### 5.2.3 Subcasos de Simulación

#### 5.2.3.1 Subcaso 1

Partiendo de la red OBS/SDEON se realizan algunas configuraciones de inicio. Se establece un espectro disponible de 100 GHz, lo cual corresponde a 8 *slots* de frecuencia, teniendo en cuenta que el espaciamiento o ancho del *slot* es de 12.5 GHz en redes con rejilla flexible, de acuerdo con la recomendación G.694.1 de la ITU-T.

#### 5.2.3.2 Subcaso 2

Para este caso, se modifica el espectro disponible, con el propósito de estudiar el efecto de esta variación en el desempeño de la red. Se define un espectro de 200 GHz, correspondiendo a un total de 16 *slots* disponibles para asignación en la red.

#### 5.2.3.3 Subcaso 1.1 y 2.1.

Se define para estos subcasos una velocidad de transmisión de enlace de 1.25 Gbps.

#### 5.2.3.4 Subcaso 1.2. y 2.2.

Se define para estos subcasos una velocidad de transmisión de enlace de 2.5 Gbps.

Las anteriores variaciones de velocidad de transmisión de enlace se realizan con el fin de estudiar y comparar el comportamiento de la red en cada uno de los casos con menor y mayor velocidad.

## 5.3 PARÁMETROS DE CONFIGURACIÓN PARA ESCENARIOS DE SIMULACIÓN

Se presentan en la Tabla 5.2, los diferentes valores de los parámetros de simulación configurados con su respectiva variable en OMNeT++, para el estudio del desempeño y comportamiento de la red NSFNeT OBS/SDEON con cada algoritmo.



**ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING**

Tabla 5.2. Parámetros de configuración de la red NSFNeT OBS/SDEON.

Parámetro	Nombre de Variable en OMNeT++	Valor o Intervalo
Espectro Disponible	bw_canal	100 GHz
		200 GHz
Número de Slots	num_slots	8 slots
		16 slots
Espaciamiento entre canales	tam_slot	12.5 GHz
Velocidad de Transmisión	wavelength_bit_rate	1.25 Gbps
		2.5 Gbps
Longitud de Paquete	packetLength	1472 kB
		100 MB
		1 GB
Tamaño mínimo de la ráfaga	minSizePadding	8 B
Tamaño máximo de la ráfaga	maxSize	8 GB
Número Máximo de Paquetes por Ráfaga	numPackets	8
Tiempo Máximo de Ensamble	maxTime	60 s

Para la longitud del paquete se definen tres valores diferentes, con el fin de observar los cambios obtenidos al variar la carga dentro de la red. En cuanto al ensamble de ráfagas, se definen los parámetros del tiempo máximo de ensamble y el tamaño máximo de ráfaga de modo que se puedan ensamblar hasta 8 paquetes por ráfaga, utilizándose el método de ensamble de ráfagas híbrido el cual termina el ensamble en el momento en que se alcanza el valor asignado de alguno de estos dos parámetros. Se define también un tamaño mínimo de ráfaga que en caso de no cumplirse se utilizan bits de relleno, al igual que un número de paquetes máximo por ráfaga.

Adicionalmente, se definen diferentes intervalos de tiempo entre generación de paquetes para generar diferentes densidades de tráfico dentro de la red mediante una fórmula donde se incluyen una duración promedio de *lightpath* y el número de nodos de la red como se muestra en la ecuación 6. A partir de lo anterior, se puede



**ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING**

---

observar los diferentes comportamientos que se obtienen en la red al estar sometida a diferentes flujos de tráfico.

$$\text{intervalo entre peticiones} = \frac{\text{duración promedio de lightpath}}{\text{número de nodos} * \text{carga normalizada}} \quad (6)$$

Para llevar a cabo las pruebas de simulación, se emplean 4 equipos con la herramienta IDE OMNeT++ versión 5.6.2 instalada. Las especificaciones técnicas de cada uno de estos equipos son descritas en la Tabla 5.3.

Tabla 5.3. Especificaciones técnicas de los equipos utilizados

<b>Equipo</b>	<b>Procesador</b>	<b>Memoria RAM</b>	<b>Disco Duro</b>	<b>Sistema Operativo</b>
<b>DELL</b>	Intel Core i5	8 GB	1000 GB HDD	Windows 10
<b>ASUS</b>	AMD Ryzen 7	16 GB	500 GB SSD	Windows 11
<b>HP</b>	Intel Core i3	4GB	500 GB HDD	Windows 10
<b>HP</b>	Intel Pentium	4GB	500 GB HDD	Windows 10

## 5.4 EJECUCIÓN DE LAS SIMULACIONES

Luego de definir los casos de simulación y plantear las diferentes configuraciones de los parámetros de la red, se realiza la ejecución de las simulaciones sobre la herramienta IDE OMNeT++. Para fines de estudio, se realizaron las pruebas de simulación por cada uno de los casos, con el siguiente orden: dos simulaciones (una para cada algoritmo implementado en la red) con diez valores de tráfico normalizado, desde 0.1 a 1, variando el espectro disponible con valores de 100 GHz y 200 Ghz y a su vez, con velocidades de transmisión de enlace de 1.25 y 2.5 Gbps, obteniéndose un total de 120 simulaciones por cada algoritmo. Cabe aclarar que el tiempo de simulación dentro de la herramienta que permite obtener resultados más precisos corresponde a un tiempo real significativamente más grande, dependiendo esto, de la capacidad del equipo utilizado, por lo que es conveniente, usar el máximo de equipos posibles en simultáneo para cuestiones de ahorro de tiempo.

Cabe resaltar que para la ejecución de las simulaciones de la red con el algoritmo cognitivo basado en DL se presentaron algunas dificultades debido a que, al requerir comunicación a través de sockets, se generó un tiempo de espera adicional cada vez que se realizaba una petición de conexión al nodo de control que limitó en gran medida la obtención de resultados más precisos por requerir mucho más tiempo para ello.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

A partir de las simulaciones realizadas, se obtienen los datos de los parámetros para medir el desempeño de la red, siendo para este caso, la probabilidad de bloqueo y el retardo extremo a extremo de ráfaga, los cuales después de obtenidos, se representan gráficamente con ayuda de la herramienta OriginPro 2021b para mejor comprensión y análisis.

### 5.5 ANÁLISIS DE RESULTADOS

En esta sección, se presenta el análisis de los resultados obtenidos a partir de las pruebas de simulación realizadas de la red NSFNeT OBS/SDEON modelada en la herramienta IDE OMNeT++ incluyendo el algoritmo clásico basado en SCC y el algoritmo cognitivo basado en DL.

#### 5.5.2 Simulación Caso 1 y 2, Subcaso 1

Los casos de simulación 1 y 2, correspondientes a la red NSFNeT OBS/SDEON con el algoritmo clásico basado en SCC y el algoritmo cognitivo basado en DL, respectivamente están conformados por sus respectivos subcasos. En el subcaso 1, se define un espectro de 100 GHz que corresponde a 8 *slots* de 12.5 GHz y a partir de este se originan los subcasos 1.1. y 1.2 en donde se definen velocidades de transmisión de enlace de 1.25 y 2.5 Gbps, respectivamente. En la Tabla 5.4, se muestra un resumen de los valores establecidos para los parámetros para cada uno de los casos y subcasos.

Tabla 5.4. Parámetros configurados para el subcaso 1.

Parámetro	Red NSFNeT OBS/SDEON con algoritmo clásico basado en SCC	Red NSFNeT OBS/SDEON con algoritmo cognitivo basado en DL
<b>Espectro disponible</b>	100 GHz (8 <i>slots</i> )	100 GHz (8 <i>slots</i> )
<b>Velocidad de transmisión de enlace</b>	1.25 y 2.5 Gbps	1.25 y 2.5 Gbps

Para cada simulación, se varía la densidad de tráfico en la red, la cual se ha normalizado con valores desde 0.1 a 1 Erlang, con un incremento de 0.1, estos valores influyen directamente en el valor obtenido para el tiempo entre generación de paquetes, y se varía, también, la longitud de paquete con el fin de obtener cargas altas, medias y bajas dentro de la red. Partiendo de las configuraciones establecidas, se realizan las representaciones gráficas de los resultados de los parámetros de desempeño considerados.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

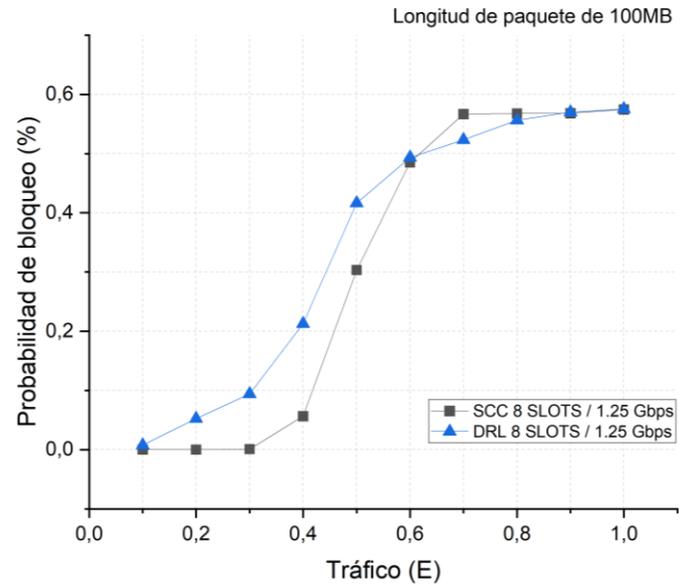
---

En la Figura 5.1 (a) y (b) se muestran las gráficas obtenidas a partir de los resultados de la probabilidad de bloqueo al simular la red NSFNet OBS/SDEON con 8 *slots* (100 GHz) y velocidades de 1.25 y 2.5 Gbps para una carga media (longitud de paquete de 100 MB), respectivamente, mientras que en la Figura 5.2 (a) y (b) se obtienen las gráficas de los resultados para una carga alta (longitud de paquete de 1 GB). Se omiten las gráficas de los resultados para una carga baja (longitud de paquete de 1472 kB) debido a que la probabilidad equivale a cero para los dos valores de velocidad de la red ya que al ser paquetes más pequeños se transmiten sin inconvenientes por la red y no se generan bloqueos en ningún momento.

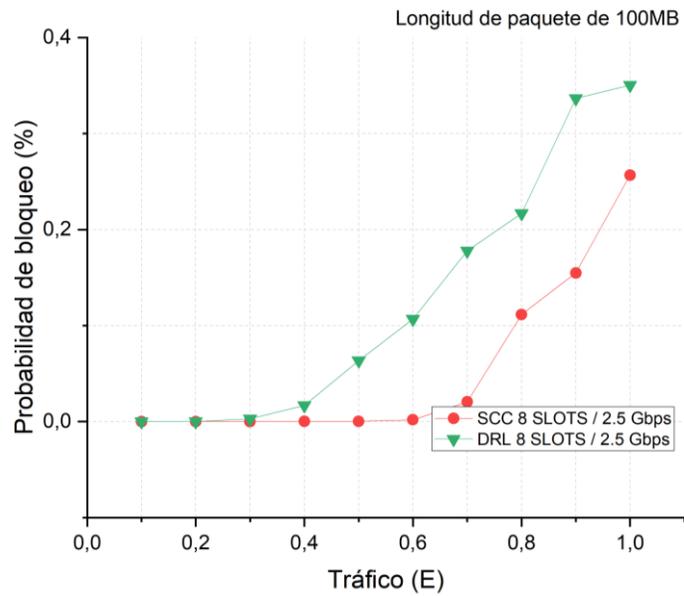
Se observa en estas gráficas que a medida que aumenta el tráfico en la red, la probabilidad de bloqueo también aumenta, esto debido a que, al haber una mayor cantidad de demandas en la red, el espectro disponible en la red empieza a saturarse más rápidamente y se dificulta realizar la asignación de éste para nuevas conexiones. Además, se observa un comportamiento similar en el desempeño de la red con los dos algoritmos ya que se obtienen valores cercanos de la probabilidad de bloqueo para las diferentes densidades de tráfico. Esta similitud se presenta debido a que los dos algoritmos utilizan la técnica de camino más corto para realizar el enrutamiento y FF para la asignación de recursos. En el caso de la la Figura 5.1 (a) y densidad de tráfico de 0.8, se observa un cambio en donde se obtiene una probabilidad de bloqueo más alta en la red con el algoritmo clásico basado en SCC debido a que a partir de estos valores el algoritmo cognitivo basado en DL ha mejorado su proceso de aprendizaje y se empieza a obtener un mejor desempeño al tomar mejores decisiones en la asignación de recursos para las conexiones. En general, debido a que el algoritmo cognitivo basado en DL aprende por sí mismo a partir de las acciones que ya ha tomado para tomar mejores acciones futuras, se presenta un mejor comportamiento (menor probabilidad de bloqueo) para cargas altas en la red con este algoritmo incluido.



# ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING



a)

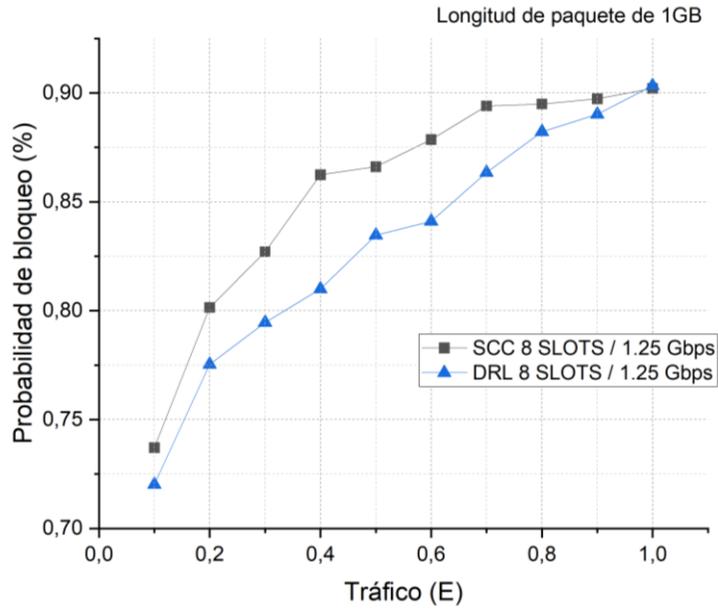


b)

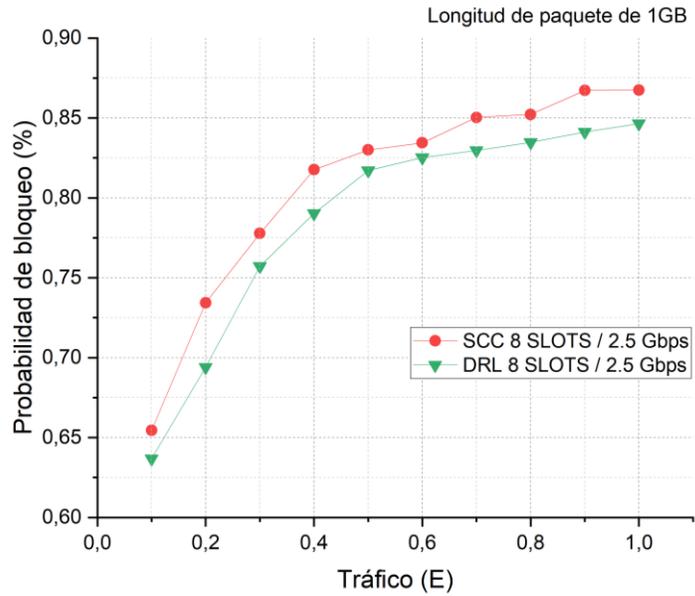
Figura 5.1. Probabilidad de bloqueo de red vs Tráfico con carga media. a) 1.25 Gbps b) 2.5 Gbps



# ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING



a)



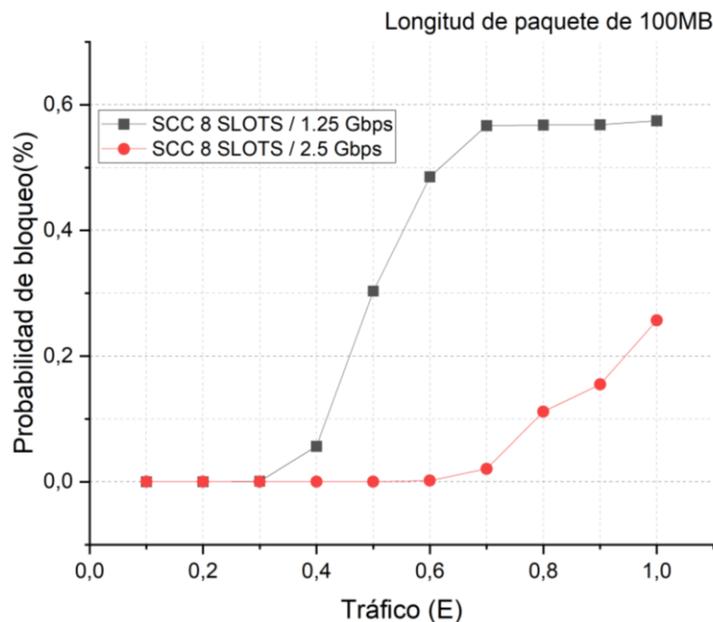
b)

Figura 5.2. Probabilidad de bloqueo de red vs Tráfico con carga alta. a) 1.25 Gbps b) 2.5 Gbps



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

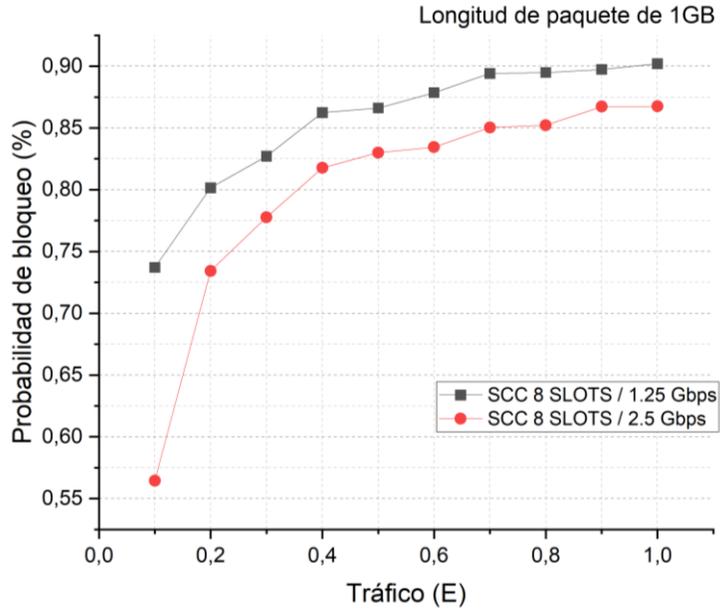
Para un mejor análisis comparativo de los efectos de la variación de la velocidad de transmisión de enlace en la probabilidad de bloqueo de la red, se obtienen las gráficas de la Figura 5.3 (a) y (b) las cuales corresponden a la red con el algoritmo clásico basado en SCC (caso 1) con carga media y alta respectivamente, y las gráficas de la Figura 5.4 (a) y (b), correspondiendo a la red con el algoritmo cognitivo basado en DL (caso 2) para carga media y alta, respectivamente. Se evidencia para las diferentes densidades de tráfico definidas, un desempeño significativamente mejor de los dos algoritmos en cuanto a probabilidad de bloqueo cuando se tienen velocidades de 2.5 Gbps, ya que, se logra que las ráfagas que circulan por la red sean transmitidas más rápidamente por las rutas respectivas permitiendo una liberación más pronta del *lightpath* y de los *slots* asignados a la conexión, quedando así disponibles para conexiones futuras. Se observa, como ejemplo, en la Figura 5.3 (a) con tráfico de 1, una disminución del 55.31% pasando de 0.57434 con velocidad de 1.25 Gbps a 0.25667 con velocidad de 2.5 Gbps.



a)

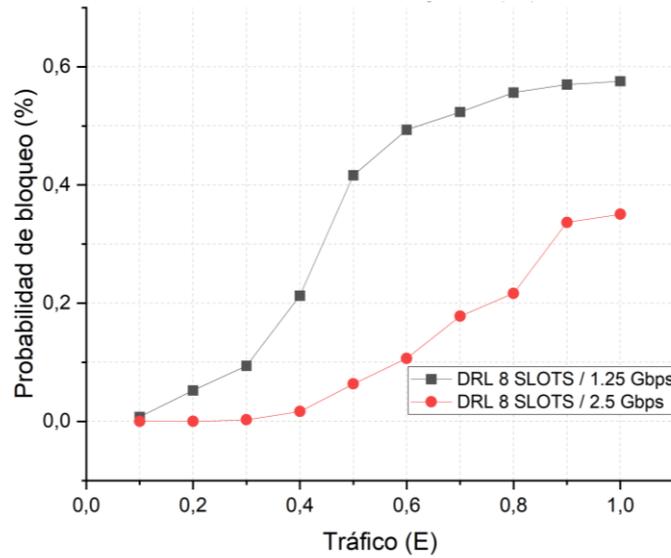


# ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING



b)

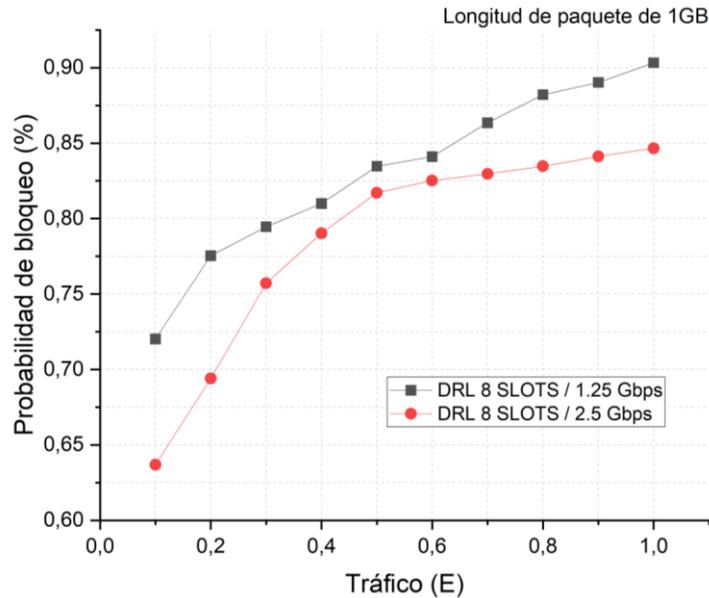
Figura 5.3. Probabilidad de bloqueo de red (con algoritmo clásico basado en SCC) vs Tráfico, Velocidades de transmisión de 1.25 y 2.5 Gbps. a) Carga media b) Carga alta.



a)



ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING



b)

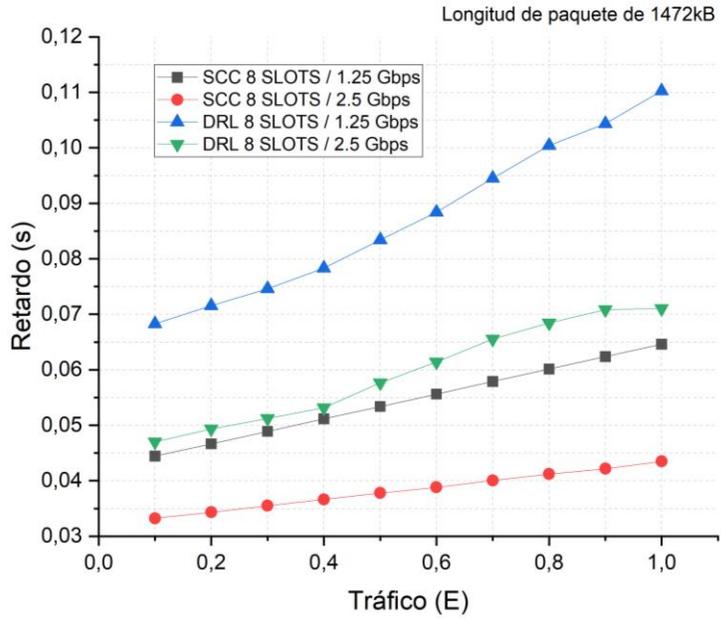
Figura 5.4. Probabilidad de bloqueo de red (con algoritmo cognitivo basado en DL) vs Tráfico, Velocidades de transmisión de 1.25 y 2.5 Gbps. a) Carga media b) Carga alta.

Otro de los parámetros a consideración para este trabajo de grado, es el retardo extremo a extremo de ráfaga, el cual se define como el tiempo transcurrido desde que la ráfaga se ensambla en el nodo origen hasta que esta llega al desensamblador del nodo destino. Como se evidencia en la Figura 5.5 (a), (b) y (c) al igual que en las gráficas anteriores, el retardo extremo a extremo incrementa proporcionalmente a medida que el tráfico en la red va incrementando. Este comportamiento se debe a que al tener un mayor número ráfagas entrantes (tráfico) en la cola de los planificadores de envío se aumenta el tiempo de espera para asignar recursos y que éstas sean transmitidas, lo que resulta en un retardo más grande. Adicionalmente, se obtienen retardos más altos cuando se usan velocidades de 1.25 Gbps, ya que, al ser velocidades menores, la ráfaga demora más tiempo en transmitirse por los enlaces de la ruta asignada. En general, la red con el algoritmo cognitivo basado en DL presenta mayores retardos debido a que este algoritmo fue desarrollado en Python, lo que generó la necesidad del uso de sockets (ya que en OMNeT++ se utiliza el lenguaje C++) para obtener los recursos para cada conexión, esto quiere decir, que en el momento en que se realiza una petición de conexión al nodo de control este debe enviar la información necesaria por medio de un socket

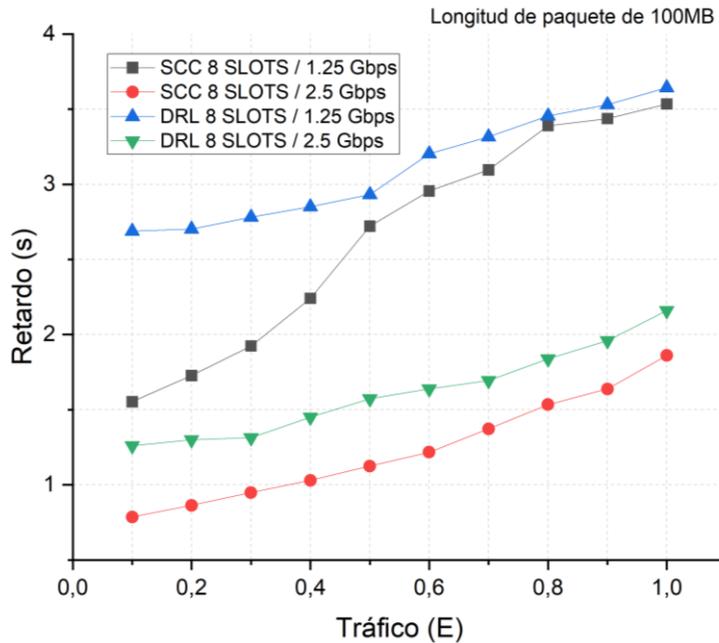


## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

al algoritmo y esperar para que este último devuelva los recursos para realizar la conexión.



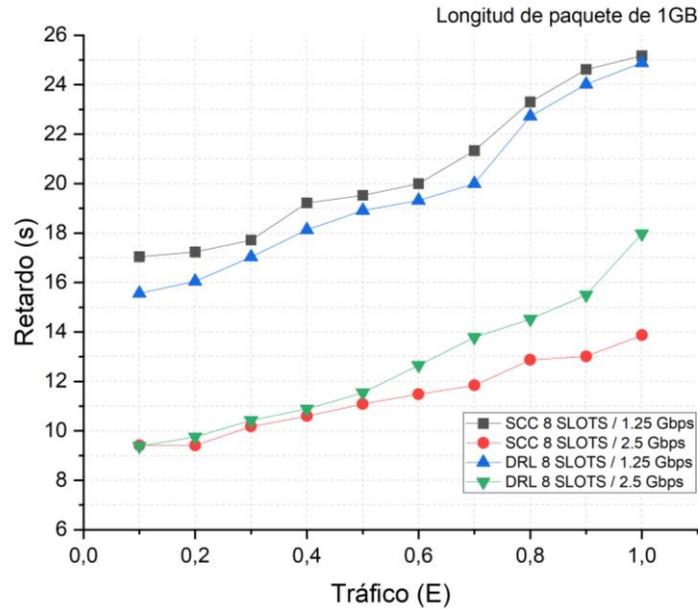
a)



b)



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING



c)

Figura 5.5. Retardo extremo a extremo de ráfaga vs Tráfico. Velocidades de 1.25 Gbps y 2.5 Gbps. a) Carga baja b) Carga media c) Carga alta

A partir de las gráficas presentadas, se puede concluir que, la red al incluir el algoritmo cognitivo basado en DL presenta un desempeño mejor con cargas altas ya que se obtienen valores menores de probabilidad de bloqueo para las diferentes densidades de tráfico definidas, mientras que en el retardo extremo a extremo de ráfaga, se obtiene un desempeño significativamente más bajo en comparación con la red con el algoritmo clásico basado en SCC. Este resultado se debe principalmente a que, en el algoritmo cognitivo basado en DL, el agente implementado logra aprender por sí mismo y de mejor manera la acción que debe tomar sobre el entorno (red) para realizar una mejor asignación de recursos para las peticiones de conexión y disminuir con esto los bloqueos en la red.

Se analiza ahora, los resultados obtenidos al aumentar el espectro disponible a 200 GHz o 16 *slots* (Subcaso 2) y variar la velocidad de transmisión de enlace.

### 5.5.3 Simulación Caso 1 y 2, Subcaso 2

Se simula ahora, la red NSFNet OBS/SDEON incluyendo por aparte cada uno de los algoritmos (caso 1 y 2) y considerando los parámetros especificados para el subcaso 2. Para este subcaso se define un espectro de 200 GHz (16 *slots*) y se



**ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING**

---

varía la velocidad de transmisión de enlace asignándole valores de 1.25 y 2.5 Gbps. En la Tabla 5.5, se presentan los parámetros configurados para las simulaciones de estos casos.

Tabla 5.5. Parámetros configurados para el subcaso 2.

<b>Parámetro</b>	<b>Red NSFNeT OBS/SDEON con algoritmo clásico basado en SCC</b>	<b>Red NSFNeT OBS/SDEON con algoritmo cognitivo basado en DL</b>
<b>Espectro Disponible</b>	200 GHz (16 <i>slots</i> )	200 GHz (16 <i>slots</i> )
<b>Velocidad de transmisión de enlace</b>	1.25 y 2.5 Gbps	1.25 y 2.5 Gbps

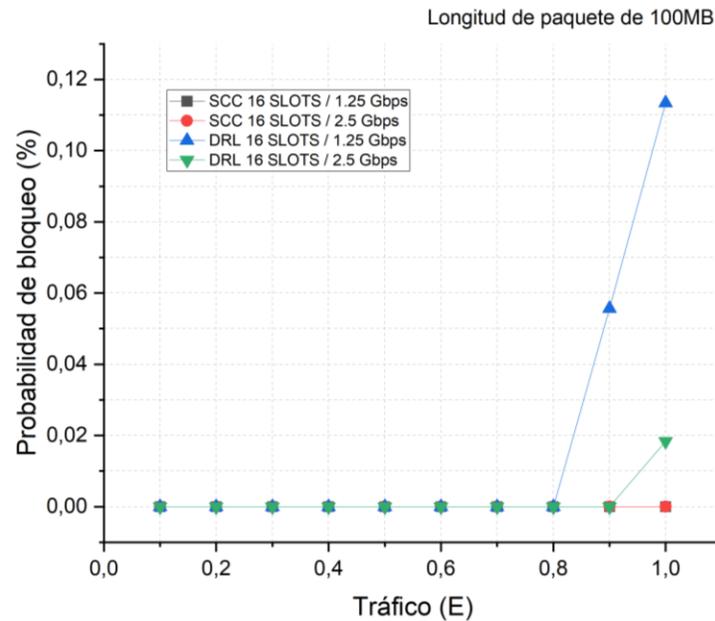
Al igual que para el subcaso 1, se realizan las simulaciones variando la densidad de tráfico en la red, normalizada entre 0.1 y 1 Erlang y con incremento de 0.1 y variando también la carga dentro de la red. Luego, se extraen los resultados de los parámetros de desempeño a partir de las simulaciones y se obtienen las gráficas para realizar el análisis correspondiente.

En la Figura 5.6 (a) y (b), se muestran las gráficas de la probabilidad de bloqueo vs el tráfico de la red con velocidades de 1.25 y 2.5 Gbps para cargas medias y altas respectivamente, donde se presenta un comportamiento similar al obtenido en las pruebas para el subcaso 1: la probabilidad de bloqueo aumenta a medida que aumenta el tráfico en la red. En contraste con los resultados del subcaso anterior, se evidencia una disminución significativa en la probabilidad de bloqueo para todas las densidades de tráfico definidas, esto debido a que para este caso se aumenta al doble el espectro disponible y por ende la cantidad de slots que pueden ser asignados a las conexiones entrantes, consiguiendo una mayor capacidad en la red para evitar saturación y eventuales bloqueos. Para carga media y velocidad de 1.25 Gbps, se evidencia en los valores de densidad de tráfico de 0.9 y 1, una probabilidad de bloqueo mayor para la red con el algoritmo cognitivo basado en DL con cambios de 0 a 0.05563 y de 0 a 0.1134 respectivamente, y, en cuanto a la red con velocidad de 2.5 Gbps, se observa un cambio cuando se tiene densidad de tráfico de 1 de 0 a 0.01832. Los anteriores cambios se producen debido a que la red con el algoritmo cognitivo basado en DL empieza a saturarse más rápidamente con valores altos de tráfico y por esta razón se bloquean algunas peticiones de conexión. Por otra parte, se obtiene una probabilidad de bloqueo menor en todos los valores de densidad de tráfico para la red con el algoritmo cognitivo basado en DL en el caso de cargas



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

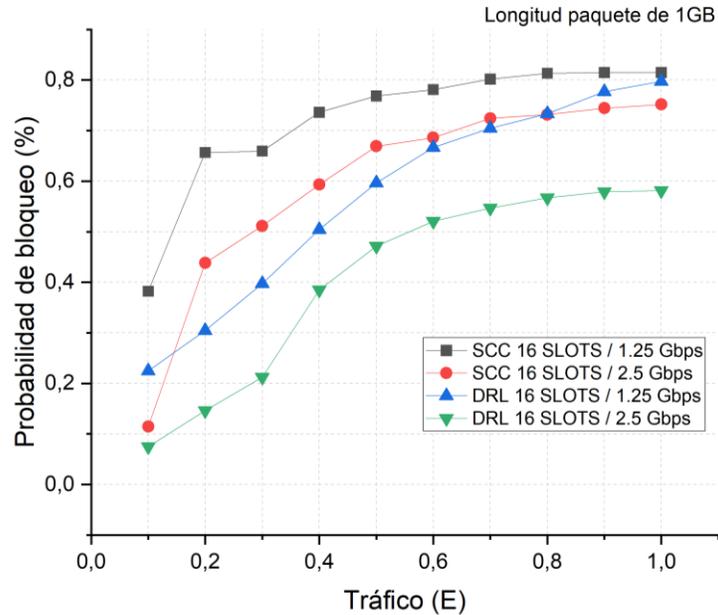
altas en comparación con la red con el algoritmo clásico basado en SCC incluido como en el subcaso 1. Al igual que en el subcaso anterior, no se tienen en cuenta los resultados para carga baja debido a que se obtiene una probabilidad de bloqueo igual a cero.



a)



ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING



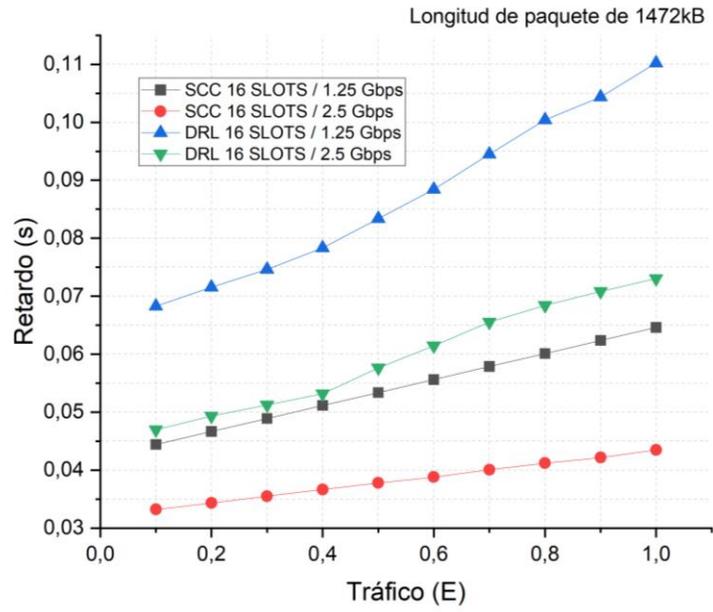
b)

Figura 5.6. Probabilidad de bloqueo vs Tráfico. a) Carga media b) Carga alta

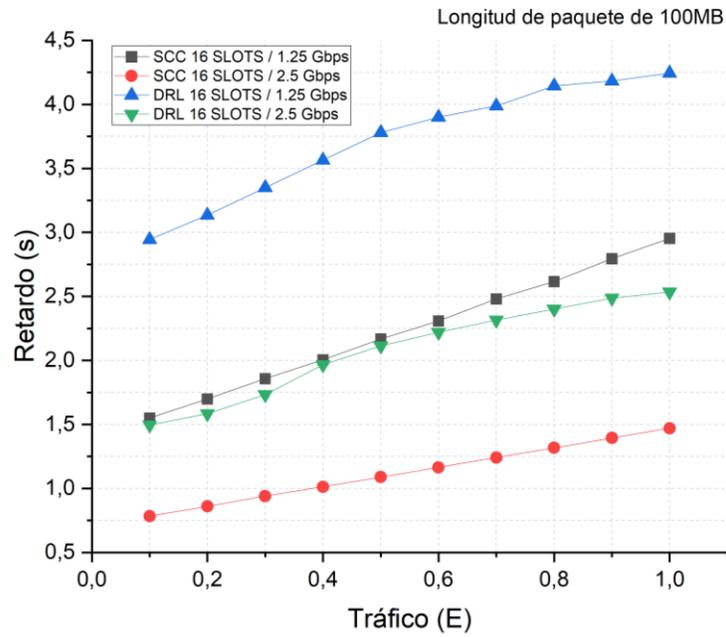
En cuanto al retardo extremo a extremo de ráfagas, se observa en la Figura 5.7 a), b) y c) para cargas bajas, medias y altas respectivamente, un incremento de este parámetro al tiempo que aumenta el tráfico de la red. Los resultados de las simulaciones de la red con el algoritmo clásico basado en SCC presentan en general, un mejor desempeño debido a que se obtienen valores de retardos más bajos en la mayoría de los casos y a medida en que se aumenta la carga se observa que los valores de los retardos para los dos algoritmos tienden a ser más cercanos, lo cual se debe a que el tiempo de transmisión de la ráfaga se vuelve mayor en comparación a los otros tiempos considerados para el cálculo del retardo. Respecto a la velocidad de transmisión de enlace, el retardo disminuye para los dos algoritmos con el valor de 2.5 Gbps y con cargas bajas ya que las ráfagas demoran un tiempo menor en transmitirse.



# ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING



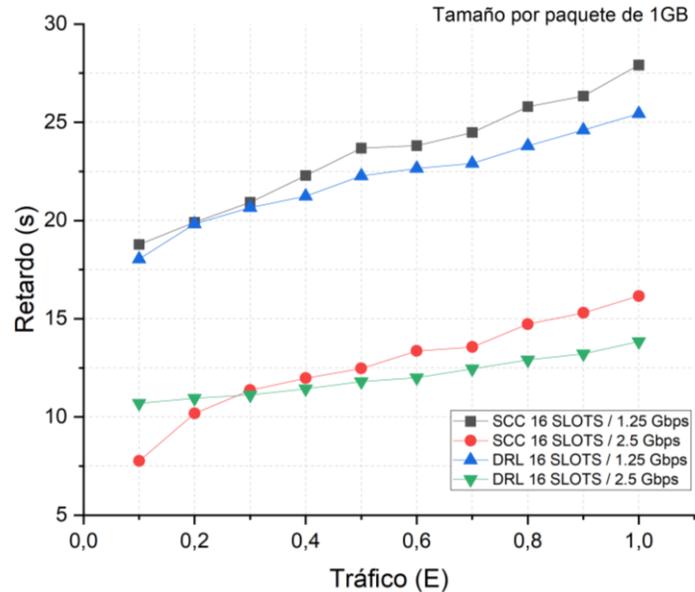
a)



b)



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING



c)

Figura 5.7. Retardo extremo a extremo vs Tráfico. a) Carga baja b) Carga media c) Carga alta

A partir de los resultados obtenidos de las simulaciones en cada uno de los casos, se evidencia en términos generales, un mejor comportamiento y desempeño en función de la probabilidad de bloqueo por parte de la red NSFNeT OBS/SDEON con el algoritmo cognitivo basado en DL cuando se tienen cargas altas, mientras que para cargas bajas o medias se presenta un mejor desempeño en la red con el algoritmo clásico basado en SCC. Respecto al retardo extremo a extremo de ráfaga, se obtiene un mejor desempeño por parte del algoritmo clásico basado en SCC en cargas bajas y medias, mientras que en cargas altas los valores de los retardos de los dos algoritmos tienden a ser más cercanos.

Para finalizar, se presenta en la Tabla 5.6 una síntesis de los valores promedio de los parámetros de desempeño estudiados para cada uno de los casos y subcasos simulados, dando una visión más clara y resumida de los resultados obtenidos a partir de las diferentes pruebas realizadas.



**ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING**

Tabla 5.6. Síntesis de los resultados de simulación

	Algoritmo clásico				Algoritmo cognitivo			
	8 slots (100 GHz)		16 slots (200 GHz)		8 slots (100 GHz)		16 slots (200 GHz)	
	1.25 Gbps	2.5 Gbps	1.25 Gbps	2.5 Gbps	1.25 Gbps	2.5 Gbps	1.25 Gbps	2.5 Gbps
<b>Probabilidad de bloqueo promedio</b>	0,90206	0,86751	0,81488	0,75178	0,89133	0,84656	0,7973	0,5812
<b>Retardo extremo a extremo promedio</b>	25,16638	13,868	27,91389	16,15898	24,8915	17,97186	25,43653	13,84587

A partir de Tabla 5.6, es posible realizar un análisis comparativo de los resultados de desempeño obtenidos en términos de la probabilidad de bloqueo y el retardo extremo a extremo para cada uno de los dos algoritmos implementados sobre la red NSFNet OBS/SDEON. Se presentan los diferentes casos donde se varía principalmente el espectro disponible y la velocidad de transmisión de enlace, todo sobre un valor de tráfico fijo normalizado de 1 Erlang y una longitud fija de paquete de 1 GB. De acuerdo con los valores resultantes, se observa que la red con el algoritmo cognitivo basado en DL presenta probabilidad de bloqueo ligeramente menor para todos los casos. Por ejemplo, analizando el caso con espectro de 100 GHz (8 slots) y velocidad de 1.25 Gbps, se obtiene para la red con el algoritmo clásico basado en SCC un valor de probabilidad de bloqueo de 0.9026 y para la red con el algoritmo cognitivo basado en DL un valor de 0.8913, consiguiendo así una ligera disminución de 1.25%. De igual manera, para el caso con espectro de 200 GHz (16 slots) y 1.25 Gbps, la probabilidad de bloqueo disminuye en un 2.15%, pasando de 0.8148 con el algoritmo clásico basado en SCC a 0.7973 con el algoritmo cognitivo basado en DL.

En cuanto al retardo extremo a extremo, se obtienen valores más bajos en la red con el algoritmo cognitivo basado en DL en la mayoría de los casos y al aumentar la velocidad de transmisión de enlace se consiguen valores más bajos. Con esto se



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

comprueba lo mencionado anteriormente, ya que para una longitud de paquete mayor se obtiene mejor desempeño en función de los dos parámetros evaluados en la red con el algoritmo cognitivo basado en DL.



## 6 CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS

En el presente capítulo, se presentan las conclusiones, recomendaciones y trabajos futuros que resultan después del desarrollo del trabajo de grado. Las conclusiones son clasificadas respecto a tres aspectos: los resultados de simulación, el desarrollo del trabajo de grado y las herramientas de simulación y de representación gráfica. Por otra parte, en las recomendaciones se abordan los aspectos a tener en cuenta en cuanto a la metodología de trabajo aplicada y la herramienta de simulación empleada. Para finalizar, se discuten algunas ideas de posibles trabajos futuros relacionados con las temáticas tratadas en el presente trabajo de grado.

### 6.2 CONCLUSIONES

#### 6.2.2 Respecto a los resultados de simulación

- Como resultado del análisis comparativo del desempeño de la red con los dos algoritmos incluidos independientemente se obtiene que la red NSFNeT OBS/SDEON que incluye el algoritmo cognitivo basado en DL presenta un mejor desempeño para todas las densidades de tráfico estudiadas en términos de la probabilidad de bloqueo cuando se tiene una longitud de paquete de 1 GB que la red NSFNeT OBS/SDEON que incluye el algoritmo clásico basado en SCC. En cuanto al retardo extremo a extremo de ráfaga, se presenta en la mayoría de los casos, un desempeño más bajo en la red con el algoritmo cognitivo basado en DL, esto debido principalmente por el tiempo adicional generado por el uso de sockets para la comunicación entre lenguajes de programación.
- El desempeño de la red NSFNeT OBS/SDEON (al incluir cualquiera de los dos algoritmos) presenta mejores resultados en los casos de menor densidad de tráfico en la red, velocidades de transmisión más altas o mayor espectro disponible ya que se obtienen valores más bajos para la probabilidad de bloqueo y para el retardo extremo a extremo de ráfaga.
- A medida en que se aumenta la cantidad de tráfico de la red NSFNeT OBS/SDEON se incrementan también la probabilidad de bloqueo y el retardo extremo a extremo de ráfaga, obteniéndose cada vez un desempeño más bajo en la red incluyendo cualquiera de los dos algoritmos.
- Al implementar el algoritmo cognitivo basado en DL en la red NSFNeT OBS/SDEON se consigue mejorar el aprovechamiento del espectro disponible ya que se consigue tomar mejores decisiones en la asignación de éste y se disminuye, por consiguiente, los bloqueos y retardos en la red. Con esto se



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

demuestra, las grandes ventajas que tiene el uso de técnicas de DL en cuanto a toma de decisiones bajo unas condiciones dadas.

### 6.2.3 Respecto al desarrollo del trabajo de grado

- El desarrollo de este trabajo de grado realiza un aporte en el estudio de algoritmos basados en técnicas de DL para la solución del problema de RSA en redes NSFNeT OBS/SDEON, presentando una propuesta que puede servir como base para el desarrollo de futuros proyectos de investigación similares. Este tipo de redes no han sido consideradas anteriormente en ningún proyecto dentro de la universidad del Cauca ni en otras universidades de Colombia.
- En el área de las redes SDEON, este trabajo realiza un aporte al proponer un algoritmo cognitivo basado en DL para el problema de RSA con el que se obtiene una mejora en el desempeño de la red cuando se tienen flujos de tráfico y longitud de ráfagas altos.

### 6.2.4 Respecto a las herramientas de simulación y de representación gráfica

- El uso de IDE OMNeT++ como herramienta de simulación de la red NFSNeT OBS/SDEON, se considera una decisión acertada, ya que, gracias a su versatilidad se puede construir cualquier tipo de modelo de simulación. No obstante, es una herramienta con una curva de aprendizaje alta ya que es necesario tener conocimientos previos de programación orientada objetos en C++ y otros aspectos relacionados con esta herramienta.
- El uso de OriginPro como herramienta para la generación de las gráficas a partir de los resultados obtenidos, se considera una decisión acertada, ya que, permitió facilitar el manejo de los datos extraídos de la red y la representación de éstos mismos de diferentes formas en gráficas de alta calidad.

## 6.3 RECOMENDACIONES

- Para el uso de la herramienta de simulación de IDE OMNeT++ se recomienda trabajar con equipos de cómputo de alta capacidad, debido a que ésta consume una gran cantidad de recursos y de esta manera se pueden evitar futuros problemas en la ejecución de las simulaciones. Además, esta herramienta presenta una curva de aprendizaje alta por lo que es recomendable un conocimiento previo de su funcionamiento y de los lenguajes en los que se basa, para facilitar la creación, implementación y modificación de los módulos de la red.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

- Es importante la aplicación de metodologías de trabajo y de simulación adecuadas que permitan llevar un buen orden en el proceso y tener una mejor comprensión y claridad durante el desarrollo del proyecto.
- Durante todo el proceso del trabajo de grado, es recomendable realizar una documentación detallada de cada paso que se va realizando con el fin de facilitar el desarrollo de las etapas posteriores del proyecto.

### 6.4 TRABAJOS FUTUROS

A partir del trabajo de grado desarrollado, surgen nuevas propuestas de investigación relacionadas con la temática abordada, las cuales son presentadas a continuación:

- Analizar el desempeño de una red NSFNeT OBS/SDEON al incluir otros métodos o técnicas de DL como CNN y GNN. En el presente trabajo de grado se utilizó un algoritmo basado en DRL.
- Analizar el impacto del uso de velocidades de transmisión altas en una red NSFNeT OBS/SDEON con valores mayores como 10 Gbps. En el trabajo presente se utilizaron únicamente velocidades de transmisión de 1.25 y 2.5 Gbps.
- Analizar el desempeño de una red OBS/SDEON basándose en una topología de red diferente como la topología GEANT. En el presente trabajo de grado se utilizó la topología de red NSFNeT.
- Implementar y caracterizar una red NSFNeT OBS/SDEON en otra herramienta de simulación como NC-TUNS u OPNET. En el presente trabajo de grado se utilizó la herramienta de simulación de OMNeT++.
- Analizar el desempeño de una red NSFNeT OBS/SDEON a nivel de emulación utilizando herramientas como ONOS y Mininet. En el presente trabajo de grado se utilizó la herramienta de simulación IDE OMNeT++.
- Analizar el desempeño de una red NSFNeT SDEON basada en otra técnica de conmutación óptica como OCS u OPS. En el presente trabajo de grado se utilizó la técnica de conmutación OBS.



## REFERENCIAS

- [1] J. Sócrates-Dantas, D. Careglio, J. Perelló, R. M. Silveira, W. V. Ruggiero, y J. Solè-Pareta, «Challenges and requirements of a control plane for elastic optical networks», *Comput. Netw.*, vol. 72, pp. 156-171, oct. 2014, doi: 10.1016/j.comnet.2014.07.007.
- [2] P. Bhaumik, S. Zhang, P. Chowdhury, S.-S. Lee, J. H. Lee, y B. Mukherjee, «Software-defined optical networks (SDONs): a survey», *Photonic Netw. Commun.*, vol. 28, n.º 1, pp. 4-18, ago. 2014, doi: 10.1007/s11107-014-0451-5.
- [3] K. Walkowiak, M. Klinkowski, B. Rabięga, y R. Goścień, «Routing and spectrum allocation algorithms for elastic optical networks with dedicated path protection», *Opt. Switch. Netw.*, vol. 13, pp. 63-75, jul. 2014, doi: 10.1016/j.osn.2014.02.002.
- [4] A. Alyatama, «Relative cost routing and spectrum allocation in elastic optical networks», *J. Opt. Commun. Netw.*, vol. 12, n.º 3, p. 38, mar. 2020, doi: 10.1364/JOCN.379585.
- [5] X. Chen, R. Proietti, y S. J. B. Yoo, «Building Autonomic Elastic Optical Networks with Deep Reinforcement Learning», *IEEE Commun. Mag.*, vol. 57, n.º 10, pp. 20-26, oct. 2019, doi: 10.1109/MCOM.001.1900151.
- [6] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-319-94463-0.
- [7] T. G. Robertazzi, *Introduction to Computer Networking*. Cham: Springer International Publishing, 2017. doi: 10.1007/978-3-319-53103-8.
- [8] G. Keiser, *Fiber Optic Communications*. Singapore: Springer Singapore, 2021. doi: 10.1007/978-981-33-4665-9.
- [9] M. Dasan, F. Francis, K. T. Sarath, E. Dipin, y T. Srinivas, «Optically Multiplexed Systems: Wavelength Division Multiplexing», en *Multiplexing*, S. Mohammady, Ed. IntechOpen, 2019. doi: 10.5772/intechopen.88086.
- [10] G. J. Andrade, «Estudio de la tecnología de conmutación óptica por ráfagas – OBS y análisis de migración de redes ópticas pasivas a esta tecnología», p. 120, 2011.
- [11] C. G. Argos, «Conmutación de Ráfagas Ópticas (OBS)», p. 11, 2008.
- [12] T. S. El-Bawab, Ed., *Optical switching*. New York ; London: Springer, 2005.
- [13] «Cisco Annual Internet Report - Cisco AIR Infographic (2018 - 2023)», *Cisco*. <https://www.cisco.com/c/en/us/solutions/executive-perspectives/annual-internet-report/infographic-c82-741491.html> (accedido 20 de enero de 2022).
- [14] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, y S. Matsuoka, «Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies», *IEEE Commun. Mag.*, vol. 47, n.º 11, pp. 66-73, nov. 2009, doi: 10.1109/MCOM.2009.5307468.



ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

- [15] J. J. Granada-Torres, A. M. Cárdenas-Soto, y N. Guerrero-González, «Redes ópticas elásticas: un nuevo paradigma en las futuras redes de telecomunicaciones», *Respuestas*, vol. 20, n.º 2, pp. 6-22, jul. 2015, doi: 10.22463/0122820X.350.
- [16] B. C. Chatterjee, N. Sarma, y E. Oki, «Routing and Spectrum Allocation in Elastic Optical Networks: A Tutorial», *IEEE Commun. Surv. Tutor.*, vol. 17, n.º 3, pp. 1776-1800, 2015, doi: 10.1109/COMST.2015.2431731.
- [17] D. Sharma y Dr. (Col) S. Kumar, «An Overview of Elastic Optical Networks and its Enabling Technologies», *Int. J. Eng. Technol.*, vol. 9, n.º 3, pp. 1643-1649, jun. 2017, doi: 10.21817/ijet/2017/v9i3/170903022.
- [18] A. F. S. Majin y E. C. Z. Quisoboní, «Análisis comparativo del desempeño de algoritmos RSA y RWA sobre una red óptica basada en la topología NSFNeT», p. 98, 2017.
- [19] «The Recent Contributions of Routing and Spectrum Assignment Algorithms in Elastic Optical Network (EON)», *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, n.º 3, pp. 2432-2442, ene. 2020, doi: 10.35940/ijitee.C8197.019320.
- [20] Z. Y. LI Yajie y Z. Y. LI Yajie, «Software Defined Optical Networks and Its Innovation Environment», *ZTE Commun.*, vol. 14, n.º 4, pp. 50-57, nov. 2019, doi: 10.3969/j.issn.1673-5188.2016.04.007.
- [21] A. S. Thyagaturu, A. Mercian, M. P. McGarry, M. Reisslein, y W. Kellerer, «Software Defined Optical Networks (SDONs): A Comprehensive Survey», *IEEE Commun. Surv. Tutor.*, vol. 18, n.º 4, pp. 2738-2786, 2016, doi: 10.1109/COMST.2016.2586999.
- [22] M. Kantor *et al.*, «A survey on multi-layer IP and optical Software-Defined Networks», *Comput. Netw.*, vol. 162, p. 106844, oct. 2019, doi: 10.1016/j.comnet.2019.06.022.
- [23] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, y S. Uhlig, «Software-Defined Networking: A Comprehensive Survey», *ArXiv14060440 Cs*, oct. 2014, [En línea]. Disponible en: <http://arxiv.org/abs/1406.0440>
- [24] M. Channegowda *et al.*, «Experimental demonstration of an OpenFlow based software-defined optical network employing packet, fixed and flexible DWDM grid technologies on an international multi-domain testbed», *Opt. Express*, vol. 21, n.º 5, p. 5487, mar. 2013, doi: 10.1364/OE.21.005487.
- [25] R. K. Jha y B. N. M. Llah, «Software Defined Optical Networks (SDON): proposed architecture and comparative analysis», *J. Eur. Opt. Soc.-Rapid Publ.*, vol. 15, n.º 1, p. 16, dic. 2019, doi: 10.1186/s41476-019-0105-4.
- [26] J. Mata *et al.*, «Artificial intelligence (AI) methods in optical networks: A comprehensive survey», *Opt. Switch. Netw.*, vol. 28, pp. 43-57, abr. 2018, doi: 10.1016/j.osn.2017.12.006.



ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

- [27] L. Zhang, X. Li, Y. Tang, J. Xin, y S. Huang, «A survey on QoT prediction using machine learning in optical networks», *Opt. Fiber Technol.*, vol. 68, p. 102804, ene. 2022, doi: 10.1016/j.yofte.2021.102804.
- [28] J. Gallego-Madrid, R. Sanchez-Iborra, P. M. Ruiz, y A. F. Skarmeta, «Machine learning-based zero-touch network and service management: A survey», *Digit. Commun. Netw.*, p. S2352864821000614, sep. 2021, doi: 10.1016/j.dcan.2021.09.001.
- [29] M. S. Frikha, S. M. Gammar, A. Lahmadi, y L. Andrey, «Reinforcement and deep reinforcement learning for wireless Internet of Things: A survey», *Comput. Commun.*, vol. 178, pp. 98-113, oct. 2021, doi: 10.1016/j.comcom.2021.07.014.
- [30] M. Sewak, *Deep Reinforcement Learning: Frontiers of Artificial Intelligence*. Singapore: Springer Singapore, 2019. doi: 10.1007/978-981-13-8285-7.
- [31] «Introducción al aprendizaje por refuerzo profundo | Teoría y práctica en Python», *Jordi TORRES.AI*. <https://torres.ai/aprendizaje-por-refuerzo/> (accedido 27 de enero de 2022).
- [32] C. Gambella, B. Ghaddar, y J. Naoum-Sawaya, «Optimization problems for machine learning: A survey», *Eur. J. Oper. Res.*, vol. 290, n.º 3, pp. 807-828, may 2021, doi: 10.1016/j.ejor.2020.08.045.
- [33] H. Dong, Z. Ding, y S. Zhang, Eds., *Deep Reinforcement Learning: Fundamentals, Research and Applications*. Singapore: Springer Singapore, 2020. doi: 10.1007/978-981-15-4095-0.
- [34] S. Dong, P. Wang, y K. Abbas, «A survey on deep learning and its applications», *Comput. Sci. Rev.*, vol. 40, p. 100379, may 2021, doi: 10.1016/j.cosrev.2021.100379.
- [35] N. C. Luong *et al.*, «Applications of Deep Reinforcement Learning in Communications and Networking: A Survey», *ArXiv181007862 Cs*, oct. 2018, Accedido: 28 de enero de 2022. [En línea]. Disponible en: <http://arxiv.org/abs/1810.07862>
- [36] L. Lei, Y. Tan, K. Zheng, S. Liu, K. Zhang, y X. Shen, «Deep Reinforcement Learning for Autonomous Internet of Things: Model, Applications and Challenges», *IEEE Commun. Surv. Tutor.*, vol. 22, n.º 3, pp. 1722-1760, 2020, doi: 10.1109/COMST.2020.2988367.
- [37] J. G. L. Perafán, «Diseño de métodos cross layer cognitivos para redes de comunicación óptica de ráfagas (OBS)», p. 573, 2011.
- [38] G. L. Perafán y P. E. J. Gómez, «Advances Toward “All Optical” Switching And Routing In Core Networks», vol. 9, p. 9, 2011.
- [39] X. Yu, Y. Chen, y C. Qiao, «A Study of Traffic Statistics of Assembled Burst Traffic in Optical Burst Switched Networks», vol. 4874, p. 11, 2002.
- [40] E. F. Arias Gerenas y I. J. Roa Rodríguez, «El problema de enrutamiento y asignación de espectro en redes ópticas flexibles: una solución desde la perspectiva de la optimización», p. 71, 2018.



ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

- [41] M. Klinkowski, M. Ruiz, L. Velasco, D. Careglio, V. Lopez, y J. Comellas, «Elastic Spectrum Allocation for Time-Varying Traffic in FlexGrid Optical Networks», *IEEE J. Sel. Areas Commun.*, vol. 31, n.º 1, pp. 26-38, ene. 2013, doi: 10.1109/JSAC.2013.130104.
- [42] B. Chatterjee y E. Oki, *Elastic Optical Networks: Fundamentals, Design, Control, and Management*. CRC Press, 2020.
- [43] S. Talebi, F. Alam, I. Katib, M. Khamis, R. Salama, y G. N. Rouskas, «Spectrum management techniques for elastic optical networks: A survey», *Opt. Switch. Netw.*, vol. 13, pp. 34-48, jul. 2014, doi: 10.1016/j.osn.2014.02.003.
- [44] R. W. Alaskar, I. Ahmad, y A. Alyatama, «Offline routing and spectrum allocation algorithms for elastic optical networks», *Opt. Switch. Netw.*, vol. 21, pp. 79-92, jul. 2016, doi: 10.1016/j.osn.2016.03.001.
- [45] A. M. López de Lerma, «Algoritmos de planificación para redes elásticas», p. 144, 2013.
- [46] B. C. Chatterjee y E. Oki, «Performance evaluation of spectrum allocation policies for elastic optical networks», en *2015 17th International Conference on Transparent Optical Networks (ICTON)*, Budapest, jul. 2015, pp. 1-4. doi: 10.1109/ICTON.2015.7193485.
- [47] R. S. Pressman, *Ingeniería del software: un enfoque práctico*. 2013. Accedido: 22 de octubre de 2020. [En línea]. Disponible en: [http://www.ingebook.com/ib/NPcd/IB\\_BooksVis?cod\\_primaria=1000187&codigo\\_libro=4272](http://www.ingebook.com/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=4272)
- [48] A. G. Petit, *Modelado y simulación. Aplicación a procesos logísticos de fabricación y servicios*. Univ. Politèc. de Catalunya, 2009.
- [49] T. Issariyakul y E. Hossain, *Introduction to Network Simulator NS2*. Boston, MA: Springer US, 2012. doi: 10.1007/978-1-4614-1406-3.
- [50] G. F. Riley y T. R. Henderson, «The ns-3 Network Simulator», en *Modeling and Tools for Network Simulation*, K. Wehrle, M. Güneş, y J. Gross, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15-34. doi: 10.1007/978-3-642-12331-3\_2.
- [51] «NS Network Simulator 3», *NS3 Simulator*. <https://ns3-code.com/ns-network-simulator-3/> (accedido 2 de febrero de 2022).
- [52] S.-Y. Wang y C.-C. Lin, «NCTUns 6.0: A Simulator for Advanced Wireless Vehicular Network Research», en *2010 IEEE 71st Vehicular Technology Conference*, Taipei, Taiwan, 2010, pp. 1-2. doi: 10.1109/VETECS.2010.5494212.
- [53] S.-Y. Wang y Y.-M. Huang, «NCTUNS Distributed Network Emulator», p. 34, 2012.



ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

- [54] «OPNET Network Simulator», *Opnet Projects*.  
<https://opnetprojects.com/opnet-network-simulator/> (accedido 3 de febrero de 2022).
- [55] «Opnet Modeler - EcuRed». [https://www.ecured.cu/Opnet\\_Modeler](https://www.ecured.cu/Opnet_Modeler) (accedido 3 de febrero de 2022).
- [56] A. Varga, «The OMNeT++ Discrete Event Simulation System», p. 7.
- [57] Madelayne Morales, M. Calle, D. Tovar, y J. Cuellar, *Simulando con OMNET: Selección de la herramienta y su utilización*. Universidad Icesi, 2013. doi: 10.13140/2.1.4890.4002.
- [58] «What is OMNeT++?» <https://omnetpp.org/intro/> (accedido 3 de febrero de 2022).
- [59] «MATLAB - El lenguaje del cálculo técnico».  
<https://la.mathworks.com/products/matlab.html> (accedido 3 de febrero de 2022).
- [60] «Getting Started with MATLAB», p. 187, 2005.
- [61] «Origin: Data Analysis and Graphing Software».  
<https://www.originlab.com/origin> (accedido 3 de febrero de 2022).
- [62] D. M. Zarzosa, «Implementación de un simulador de redes ópticas metropolitanas flexibles (gridless)», p. 73, 2016.
- [63] S. D. Ossa Hernández, J. A. Vargas Gutiérrez, y G. A. Gómez Agredo, «Propuesta de un algoritmo para la asignación dinámica de espectro en redes ópticas elásticas (EON)», *Lámpsakos*, n.º 23, p. 15, may 2020, doi: 10.21501/21454086.3145.
- [64] B. Mukherjee, I. Tomkos, M. Tornatore, P. Winzer, y Y. Zhao, Eds., *Springer Handbook of Optical Networks*. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-16250-4.
- [65] «Algoritmo de Dijkstra - EcuRed».  
[https://www.ecured.cu/Algoritmo\\_de\\_Dijkstra](https://www.ecured.cu/Algoritmo_de_Dijkstra) (accedido 22 de febrero de 2022).
- [66] «Algoritmo de Dijkstra».  
[http://i3campus.co/CONTENIDOS/wikipedia/content/a/algoritmo\\_de\\_dijkstra.html](http://i3campus.co/CONTENIDOS/wikipedia/content/a/algoritmo_de_dijkstra.html) (accedido 22 de febrero de 2022).
- [67] G. Horna, «Evaluación de algoritmos de asignación de recursos que ofrezcan protección en redes ópticas elásticas», p. 95, 2017.
- [68] *Optical-Networks-Group/rsa-rl-tutorials*. Optical-Networks-Group, 2022. Accedido: 24 de febrero de 2022. [En línea]. Disponible en: [https://github.com/Optical-Networks-Group/rsa-rl-tutorials/blob/14401693377c60bd3d21e4022bde2ace9865395f/tutorials/03\\_DR\\_L\\_Agent\\_en.ipynb](https://github.com/Optical-Networks-Group/rsa-rl-tutorials/blob/14401693377c60bd3d21e4022bde2ace9865395f/tutorials/03_DR_L_Agent_en.ipynb)



## ANEXOS

### 1. Archivo de configuración inicial

```
network = NSFnetUnificado.net.NSFnetComplete2

simtime-resolution = us
**.edgeNodeAssembler.rulesFile = "rules.dat"
debug-on-errors = true
num-rngs = 1024
seed-0-mt=6

**.tam_slot = 12.5 #GHz
***.bw_canal = 100 #GHz
**.bw_canal = 200 #GHz
**.wavelength_bit_rate = 1.25Gbps #velocidad de transmisión de enlace
***.wavelength_bit_rate = 2.5Gbps #velocidad de transmisión de enlace
***.packetLength = 1472kB
***.packetLength = 100MB #3
**.packetLength = 1GB #5

#Parte WRON Dinamica
**.dynamic=true
**.duracion_media_lightpath_dinamico = 18#exponential (1/6,3) #0.012527707
**.dynamic_RWA_algorithm = ${"Shortest_Path_First_Fit"}
***.dynamic_RWA_algorithm = "AlgoritmoDRL"
**.dynamic_load = ${1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1}
**.num_nodos = 14

#===== #=====
#=====
# Burstifier configuration (the same parameters for all)
#===== #=====
#=====

**.burstifier[*].minSizePadding = 8B
**.burstifier[*].addLastPacket = true
**.tamHeader = 8B
**.tamHeaderPacket = 1B
***.burstifier[*].maxTime = 0.005s #Cant.Original 5ms (0.005) 1 solo
paquete por rafaga
**.burstifier[*].maxTime = 60s
**.burstifier[*].maxSize = 8GB #Cant.Original 125e6B (125000000B)
**.burstifier[*].numPackets = 8 #Cant.Original 3
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

```
#===== #=====
#=====
# Burst Label field value
#===== #=====
#=====
**.numBurstifiers = 42

**.burstifier[0].destLabel = 0
**.burstifier[1].destLabel = 0
**.burstifier[2].destLabel = 0
**.burstifier[3].destLabel = 1
**.burstifier[4].destLabel = 1
**.burstifier[5].destLabel = 1
**.burstifier[6].destLabel = 2
**.burstifier[7].destLabel = 2
**.burstifier[8].destLabel = 2
**.burstifier[9].destLabel = 3
**.burstifier[10].destLabel = 3
**.burstifier[11].destLabel = 3
**.burstifier[12].destLabel = 4
**.burstifier[13].destLabel = 4
**.burstifier[14].destLabel = 4
**.burstifier[15].destLabel = 5
**.burstifier[16].destLabel = 5
**.burstifier[17].destLabel = 5
**.burstifier[18].destLabel = 6
**.burstifier[19].destLabel = 6
**.burstifier[20].destLabel = 6
**.burstifier[21].destLabel = 7
**.burstifier[22].destLabel = 7
**.burstifier[23].destLabel = 7
**.burstifier[24].destLabel = 8
**.burstifier[25].destLabel = 8
**.burstifier[26].destLabel = 8
**.burstifier[27].destLabel = 9
**.burstifier[28].destLabel = 9
**.burstifier[29].destLabel = 9
**.burstifier[30].destLabel = 10
**.burstifier[31].destLabel = 10
**.burstifier[32].destLabel = 10
**.burstifier[33].destLabel = 11
**.burstifier[34].destLabel = 11
**.burstifier[35].destLabel = 11
**.burstifier[36].destLabel = 12
**.burstifier[37].destLabel = 12
**.burstifier[38].destLabel = 12
**.burstifier[39].destLabel = 13
**.burstifier[40].destLabel = 13
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

```
**burstifier[41].destLabel = 13
```

```
*.edgeCore[0].x1 = 800  
*.edgeCore[0].y1 = 3500  
*.edgeCore[1].x1= 1300  
*.edgeCore[1].y1 = 1700  
*.edgeCore[2].x1 = 2200  
*.edgeCore[2].y1 = 5000  
*.edgeCore[3].x1 = 2600  
*.edgeCore[3].y1 = 3200  
*.edgeCore[4].x1 = 3700  
*.edgeCore[4].y1 = 4100  
*.edgeCore[5].x1 = 5000  
*.edgeCore[5].y1 = 3900  
*.edgeCore[6].x1 = 6500  
*.edgeCore[6].y1 = 3800  
*.edgeCore[7].x1 = 5500  
*.edgeCore[7].y1 = 6000  
*.edgeCore[8].x1 = 8200  
*.edgeCore[8].y1 = 2200  
*.edgeCore[9].x1 = 8000  
*.edgeCore[9].y1 = 3600  
*.edgeCore[10].x1 = 10100  
*.edgeCore[10].y1 = 2000  
*.edgeCore[11].x1 = 8200  
*.edgeCore[11].y1 = 5800  
*.edgeCore[12].x1 = 9000  
*.edgeCore[12].y1 = 4800  
*.edgeCore[13].x1 = 10200  
*.edgeCore[13].y1 = 4000
```

## 2. Archivo de topología de red NSFNeT

```
package NSFnetUnificado.net;  
  
import NodoControl.NodoControl;  
import EdgeAndCore2.EdgeCore;  
import NodoControl.sencillo.sencillo;  
  
network NSFnetComplete2  
{  
    parameters:  
        bool dynamic;  
        double tam_slot;  
        double bw_canal;
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

```
int num_nodos;  
int num_slots=int(bw_canal/tam_slot);  
double wavelength_bit_rate @unit(bps);  
double duracion_media_lightpath_dinamico;  
double dynamic_load;  
@display("bgb=11626.666,7773.333;bgi=maps/usa,s,bgg  
1000,2,greys95;bgs=0.075,km");
```

### submodules:

```
nodoControl: NodoControl {  
    num_nodos = num_nodos;  
    tam_slot = tam_slot;  
    bw_canal = bw_canal;  
    wavelength_bit_rate = wavelength_bit_rate;  
    dynamic_load = dynamic_load;  
  
@display("i=block/star_1;p=1477.6301,6830.4785;b=706.6666,613.3333");  
  
}  
  
edgeCore[14]: EdgeCore {  
  
    dynamic = dynamic;  
    num_nodos = num_nodos;  
    tam_slot = tam_slot;  
    bw_canal = bw_canal;  
    wavelength_bit_rate = wavelength_bit_rate;  
    dynamic_load = dynamic_load;  
    duracion_media_lightpath_dinamico =  
duracion_media_lightpath_dinamico;  
  
@display("p=171.09401,77.770004,exact,$x1,$y1");  
  
    gates:  
        in[num_slots*4];  
        out[num_slots*4];  
}  
  
TABLA: sencillo {  
  
    parameters:  
  
        @display("p=14500,12500;b=10,10");  
  
}
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

PETICIONES: sencillo {

**parameters:**

```
@display("p=6000,8500;b=10,10");
```

}

**connections allowunconnected:**

```
for i=0..num_slots-1 {  
    edgeCore[0].out[i] --> { datarate = wavelength_bit_rate; delay =  
0.003776667s;@display("ls=black,2"); } --> edgeCore[1].in[i];  
    edgeCore[1].out[i] --> { datarate = wavelength_bit_rate; delay =  
0.003776667s;@display("ls=black,2"); } --> edgeCore[0].in[i];  
}
```

```
for i=0..num_slots-1 {  
    edgeCore[0].out++ --> { datarate = wavelength_bit_rate; delay =  
0.002333333s;@display("ls=black,2"); } --> edgeCore[2].in[i];  
    edgeCore[2].out[i] --> { datarate = wavelength_bit_rate; delay =  
0.002333333s;@display("ls=black,2"); } --> edgeCore[0].in++;  
}
```

```
for i=0..num_slots-1 {  
    edgeCore[0].out++ --> { datarate = wavelength_bit_rate; delay =  
0.003316667s;@display("ls=black,2"); } --> edgeCore[3].in[i];  
    edgeCore[3].out[i] --> { datarate = wavelength_bit_rate; delay =  
0.003316667s;@display("ls=black,2"); } --> edgeCore[0].in++;  
}
```

```
for i=0..num_slots-1 {  
    edgeCore[1].out++ --> { datarate = wavelength_bit_rate; delay =  
0.005716667s;@display("ls=black,2"); } --> edgeCore[2].in++;  
    edgeCore[2].out++ --> { datarate = wavelength_bit_rate; delay =  
0.005716667s;@display("ls=black,2"); } --> edgeCore[1].in++;  
}
```

```
for i=0..num_slots-1 {  
    edgeCore[1].out++ --> { datarate = wavelength_bit_rate; delay =  
0.009443333s;@display("ls=black,2"); } --> edgeCore[6].in[i];  
    edgeCore[6].out[i] --> { datarate = wavelength_bit_rate; delay =  
0.009443333s;@display("ls=black,2"); } --> edgeCore[1].in++;  
}
```

```
for i=0..num_slots-1 {  
    edgeCore[2].out++ --> { datarate = wavelength_bit_rate; delay =  
0.006983333s;@display("ls=black,2"); } --> edgeCore[7].in[i];  
}
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
    edgeCore[7].out[i] --> { datarate = wavelength_bit_rate; delay =
0.006983333s;@display("ls=black,2"); } --> edgeCore[2].in++;
}

    for i=0..num_slots-1 {
        edgeCore[3].out++ --> { datarate = wavelength_bit_rate; delay =
0.001896667s;@display("ls=black,2"); } --> edgeCore[4].in[i];
        edgeCore[4].out[i] --> { datarate = wavelength_bit_rate; delay =
0.001896667s;@display("ls=black,2"); } --> edgeCore[3].in++;
    }

    for i=0..num_slots-1 {
        edgeCore[3].out++ --> { datarate = wavelength_bit_rate; delay =
0.007833333s;@display("ls=black,2"); } --> edgeCore[8].in[i];
        edgeCore[8].out[i] --> { datarate = wavelength_bit_rate; delay =
0.007833333s;@display("ls=black,2"); } --> edgeCore[3].in++;
    }

    for i=0..num_slots-1 {
        edgeCore[4].out++ --> { datarate = wavelength_bit_rate; delay =
0.0025s;@display("ls=black,2"); } --> edgeCore[5].in[i];
        edgeCore[5].out[i] --> { datarate = wavelength_bit_rate; delay =
0.0025s;@display("ls=black,2"); } --> edgeCore[4].in++;
    }

    for i=0..num_slots-1 {
        edgeCore[4].out++ --> { datarate = wavelength_bit_rate; delay =
0.00484s;@display("ls=black,2"); } --> edgeCore[7].in++;
        edgeCore[7].out++ --> { datarate = wavelength_bit_rate; delay =
0.00484s;@display("ls=black,2"); } --> edgeCore[4].in++;
    }

    for i=0..num_slots-1 {
        edgeCore[5].out++ --> { datarate = wavelength_bit_rate; delay =
0.002403333s;@display("ls=black,2"); } --> edgeCore[6].in++;
        edgeCore[6].out++ --> { datarate = wavelength_bit_rate; delay =
0.002403333s;@display("ls=black,2"); } --> edgeCore[5].in++;
    }

    for i=0..num_slots-1 {
        edgeCore[6].out++ --> { datarate = wavelength_bit_rate; delay =
0.002316667s;@display("ls=black,2"); } --> edgeCore[9].in[i];
        edgeCore[9].out[i] --> { datarate = wavelength_bit_rate; delay =
0.002316667s;@display("ls=black,2"); } --> edgeCore[6].in++;
    }

    for i=0..num_slots-1 {
```



ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

```
    edgeCore[7].out++ --> { datarate = wavelength_bit_rate; delay =
0.006576667s;@display("ls=black,2"); } --> edgeCore[12].in[i];
    edgeCore[12].out[i] --> { datarate = wavelength_bit_rate; delay =
0.006576667s;@display("ls=black,2"); } --> edgeCore[7].in++;
}

    for i=0..num_slots-1 {
    edgeCore[7].out++ --> { datarate = wavelength_bit_rate; delay =
0.003763333s;@display("ls=black,2"); } --> edgeCore[11].in[i];
    edgeCore[11].out[i] --> { datarate = wavelength_bit_rate; delay =
0.003763333s;@display("ls=black,2"); } --> edgeCore[7].in++;
}

    for i=0..num_slots-1 {
    edgeCore[8].out++ --> { datarate = wavelength_bit_rate; delay =
0.00198s;@display("ls=black,2"); } --> edgeCore[10].in[i];
    edgeCore[10].out[i] --> { datarate = wavelength_bit_rate; delay =
0.00198s;@display("ls=black,2"); } --> edgeCore[8].in++;
}

    for i=0..num_slots-1 {
    edgeCore[8].out++ --> { datarate = wavelength_bit_rate; delay =
0.00263s;@display("ls=black,2"); } --> edgeCore[13].in[i];
    edgeCore[13].out[i] --> { datarate = wavelength_bit_rate; delay =
0.00263s;@display("ls=black,2"); } --> edgeCore[8].in++;
}

    for i=0..num_slots-1 {
    edgeCore[9].out++ --> { datarate = wavelength_bit_rate; delay =
0.002583333s;@display("ls=black,2"); } --> edgeCore[10].in++;
    edgeCore[10].out++ --> { datarate = wavelength_bit_rate; delay =
0.002583333s;@display("ls=black,2"); } --> edgeCore[9].in++;
}

    for i=0..num_slots-1 {
    edgeCore[9].out++ --> { datarate = wavelength_bit_rate; delay =
0.002796667s;@display("ls=black,2"); } --> edgeCore[11].in++;
    edgeCore[11].out++ --> { datarate = wavelength_bit_rate; delay =
0.002796667s;@display("ls=black,2"); } --> edgeCore[9].in++;
}

    for i=0..num_slots-1 {
    edgeCore[9].out++ --> { datarate = wavelength_bit_rate; delay =
0.001503333s;@display("ls=black,2"); } --> edgeCore[13].in++;
    edgeCore[13].out++ --> { datarate = wavelength_bit_rate; delay =
0.001503333s;@display("ls=black,2"); } --> edgeCore[9].in++;
}
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
for i=0..num_slots-1 {
    edgeCore[10].out++ --> { datarate = wavelength_bit_rate; delay =
0.001366667s;@display("ls=black,2"); } --> edgeCore[12].in++;
    edgeCore[12].out++ --> { datarate = wavelength_bit_rate; delay =
0.001366667s;@display("ls=black,2"); } --> edgeCore[10].in++;
}

for i=0..num_slots-1 {
    edgeCore[12].out++ --> { datarate = wavelength_bit_rate; delay =
0.00082s;@display("ls=black,2"); } --> edgeCore[13].in++;
    edgeCore[13].out++ --> { datarate = wavelength_bit_rate; delay =
0.00082s;@display("ls=black,2"); } --> edgeCore[12].in++;
}
}
```

### 3. Algoritmo para generación de paquetes

```
Source::~Source(){
    cancelAndDelete(interrupcion);
}

void Source::initialize(int stage)
{
    if(stage==1)
    {
        num_nodos = (int)par("num_nodos");
        carga = (double)par("dynamic_load");
        duracion_media_lightpath =
(double)par("duracion_media_lightpath_dinamico");
        intervalo_medio_entre_peticiones = duracion_media_lightpath/((num_nodos-
1)*carga); //vamos a utilizar esta formula para generar paquetes
        rng_interrupcion = getRNG(0); //valores de manera aleatoria que se
definen para usarlos en las distribuciones
        rng_destino = getRNG(2);
        rng_duracion_lightpath = getRNG(3);
        if((bool)getParentModule()->par("dynamic")==true) //si el parametro es
true en el modulo NSFnet
        {
            interrupcion=new cMessage("Solicitud de Nuevo
Lightpath",INTERRUPCION); //mensaje con bandera interrupcion
        }
        scheduleAt(simTime()+exponential(intervalo_medio_entre_peticiones,0),interrupcio
n); // se envia la solicitud
    }
}
packetLengthBytes=&par("packetLength");
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
}  
  
void Source::handleMessage(cMessage *msg)  
{  
    switch (msg->getKind()) //estructura de control condicional  
    {  
        case INTERRUPCION: //si la bandera del mensaje es igual a  
            interrupcion  
            {  
                origen = getParentModule()->getParentModule()-  
>getParentModule()->getIndex(); //obtiene el id del mensaje que se crea cuando  
                este se crea y lo asigna como origen  
                destino = origen + 1 + intuniform (0,num_nodos-2,2); //  
                distribucion u de 0 a 12, el ultimo 2 corresponde a RNG(2) de destino  
                // origen=6;  
                // destino=1;  
                // algoritmo = "Shortest_Path_First_Fit";  
                algoritmo = "AlgoritmoDRL";  
  
                if(destino >= num_nodos)destino-=num_nodos; // si es mayor a 14  
                se resta este numero  
  
                tiempo_servicio = exponential (duracion_media_lightpath,3);  
                paquete_optico = new PacketOptico("Paquete",PETICION_LIGHTPATH);  
                //se crea el mensaje  
                paquete_optico->setDirOrigen(origen); // se llenan los campos  
                del mensaje  
                paquete_optico->setDirDestino(destino);  
                paquete_optico->setTiempo_servicio(tiempo_servicio);  
                paquete_optico->setLongitud(packetLengthBytes->intValue()*8);  
                paquete_optico->setAlgoritmo(algoritmo);  
                int type = intuniform(0,2);  
                paquete_optico->setPriority(type);  
                EV<<"Nueva Peticion"<<endl;  
                send(paquete_optico,"out");  
  
                scheduleAt(simTime()+exponential(intervalo_medio_entre_peticiones,0),msg); //se  
                programa un nuevo evento para el mismo  
                break;  
            }  
        delete(msg);  
    };  
};  
}
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

### 4. Algoritmo para clasificación de paquetes

```
#include "Classifier.h"

Define_Module(Classifier);

Classifier::~Classifier(){
    if (numOuts != 0){
        free(rules); //libera memoria dinámica
    }
}

//Abre el archivo Classifier_Rules, lee todas las reglas y crea objetos para cada una.
void Classifier::initialize(){

    numOuts = par("numOuts");
    droppedPacket = 0;
    WATCH(droppedPacket); //permite inspeccionar esta variable en la interfaz gráfica y se emite en el archivo instantáneo.

    if (numOuts != 0){

        //Almacena la información proveniente de Clasifier_Rules
        rules = (Classifier_Rules*)calloc(numOuts,sizeof(Classifier_Rules)); //se usa calloc para reservar memoria, de forma dinámica
        //lee una línea cada vez y crea la regla asociada para cada cola.
        char *line = (char*)calloc(1500,sizeof(char)); //Debido a esto, el máximo número de caracteres por línea es 1500.
        int i=0;

        //abre el archivo Rules.dat y lo almacena en rulesFile
        const char *rulesFile = par("rules");

        //Si rules está vacío, se muestra un mensaje de error y para la simulación.
        if(strlen(rulesFile) == 0){
            EV_ERROR<<"El archivo de reglas no se encuentra definido"<<endl;
        }
        // abre el archivo Rules.Dat en *ruleFile
        FILE *ruleFile = fopen(rulesFile,"r"); // r = leer el archivo...
        w=escribir...d=Borrar

        if(ruleFile != NULL){
            //fgets lee hasta que se encuentra un carácter \n
            while(fgets(line,1500,ruleFile) != NULL){
                if(strcmp(line,"\n") != 0 && line[0] != '#'){ //Ignora los comentarios (líneas que empiezan con #)
                    //toma "line" como parámetro de entrada
                }
            }
        }
    }
}
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
        //para realizar la comparación de los parámetros
        rules[i] = Classifier_Rules(line);
        i++;
    }
}
else{
    EV_ERROR<<"No es posible abrir el archivo de reglas"<<endl;
}

fclose(ruleFile);
// i debe ser igual a numQueues. Si no lo es, seguramente hay un error.
if(!(i == numOuts)){
    printf("(Classifier_Rules) Aviso: El despachador de reglas no
coincide con las colas de los modulos.\n");
}
free(line);
}
}

//Cuando llega un paquete, se compara con todas las reglas. Si coincide con
alguna regla, se manda al correspondiente compuerta de salida. Si no, se
descarta.
void Classifier::handleMessage(cMessage *msg){

    int i;
    pkt= check_and_cast<PacketOptico *> (msg);
    // se registra el paquete entrante
    //fin del registro
    //Verifica si el paquete coincide con alguna regla. Se mandará a la primera
compuerta donde coincide la regla.
    for(i=0;i<(int)numOuts;i++){
        if(rules[i].match(pkt)){
            send(pkt,"out",i);
            return;
        }
    }
    //Si el paquete no coincide con alguna regla, se descarta.
    delete pkt;
    //contador de paquetes descartados
    droppedPacket++;
}
```

### 5. Algoritmo para conformación de ráfagas

```
#include "Burstifier.h"
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

```
using namespace omnetpp;

Define_Module(Burstifier);

Burstifier::~Burstifier()
{
    burst.clear();
    cancelAndDelete(maxTime_msg);
}

void Burstifier::initialize()
{
    maxTime =par("maxTime"); // Tiempo máximo. Si este tiempo se cumple, la ráfaga será ensamblada con el contenido de la cola en ese instante.
    maxSize=&par("maxSize"); // Tamaño Máximo de la ráfaga.
    numPackets =par("numPackets"); // Máximo número de paquetes.
    minSizePadding=par("minSizePadding"); // Mínimo tamaño que una ráfaga debe tener. Si no logra conseguir este valor, se aumentan bits de relleno. Si es 0, no se aumentan bits de relleno
    addLastPacket=par("addLastPacket"); // Si es verdadero, permite que el último paquete se adicione a la ráfaga, incluso si la ráfaga excede su tamaño. Si es falso, la ráfaga se ensambla y el paquete será el primero de la siguiente ráfaga.
    tamHeader=par("tamHeader"); // Tamaño de la cabecera de la ráfaga.
    tamHeaderPacket=par("tamHeaderPacket"); // Tamaño de la cabecera de cada paquete insertado a la ráfaga.
    destLabel = par("destLabel"); // Etiqueta de destino
    indexRuta=0;
    saltos=0;

    //Inicio de la cola
    burst.setName("colabursts");

    //Inicio de contadores
    int tamHeaderBits = tamHeader*8;
    burstBits = tamHeaderBits;
    burstBytes=tamHeader;
    numPacketsInBurst = 0;

    //Se crea el mensaje de tiempo máximo
    maxTime_msg = new cMessage("maxTime");

    //Estadísticas
    burstCounter =0;
}
}
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
//Se reciben los paquetes, se almacenan en una cola y se ensamblan si se cumple
aguna condición
void Burstifier::handleMessage(cMessage *msg)
{
    if(msg->isSelfMessage()) //Si recibe un automensaje...
    {
        assemblyBurst(); //Ensambla la ráfaga e inicializa las variables
    }
    else //Si no, es un paquete?
    {
        pqt= check_and_cast<PacketOptico *> (msg);
        sourceAddress = pqt->getDirOrigen();
        destAddress = pqt->getDirDestino();
        ArraySize = pqt->getRutaOArraySize();

        bool desborde = false;
        int tamHeaderPacketBits = tamHeaderPacket*8;
        if(((burstBits + pqt->getLongitud() + tamHeaderPacketBits)> maxSize-
>intValue()*8)) desborde = true;
        // Comprobar si el burttifier está vacío
        if(burst.isEmpty()) // Si lo está, inicializar el tiempo máximo y la
cola
        {
            scheduleAt(simTime() + maxTime, maxTime_msg ); //Establece el tiempo
máximo
            //Registrar el momento en el que llega el primer paquete
            firstPacket_t = simTime();
        }
        else if(desborde && !addLastPacket)
        { // Entra si se necesita ensamblar la ráfaga antes de que el paquete
sea insertado en la cola
            if(burst.isEmpty())EV_ERROR<<"No es posible ensamblar una rafaga con
una cola vacia"<<endl;
            //Ensamblar la ráfaga e iniciar contadores
            assemblyBurst();
            if(maxTime_msg->isScheduled())cancelEvent(maxTime_msg); //Se cancela
el tiempo maximo y se programa uno nuevo
            scheduleAt(simTime() + maxTime, maxTime_msg);
            firstPacket_t = simTime();
            //Calcular si la sobrecarga se da con el primer paquete
            desborde = false;
            if(((burstBits + pqt->getLongitud() + tamHeaderPacketBits)> maxSize-
>intValue()*8)) desborde=true;
        }
        //Insertar el paquete actual en la cola
        // timeAssembled.collect(simTime()-firstPacket_t);
        burst.insert(pqt);
        burstBits += pqt->getLongitud() + tamHeaderPacketBits;
    }
}
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
burstBytes = burstBits/8;
numPacketsInBurst++;
//Si la sobrecarga no está habilitada, pero ocurre sobrecarga cuando se
inserta el primer paquete, se genera un error.
if(desborde && !addLastPacket)
    EV_ERROR<<"La sobrecarga se genero insertando el primer mensaje y
los requerimientos no permiten sobrecarga(overflowLastPacket es falso)"<<endl;

//Si ocurre la sobrecarga o se llena al máximo número de paquetes se
debe ensamblar la ráfaga.
if((desborde || numPacketsInBurst == numPackets) || burstBits ==maxSize-
>intValue()*8)
{
    assemblyBurst();
    if(maxTime_msg->isScheduled()) cancelEvent(maxTime_msg);
}
}
}
void Burstifier::assemblyBurst()
{
    // Comprobar si la rafaga logra el mínimo tamaño, si no, usar bits de
relleno

    int minTamanRellenoBits = minSizePadding*8;

    if(burstBits < minTamanRellenoBits) burstBits =minTamanRellenoBits;

    //Se crea la ráfaga e inserta el contenido de la cola
    Burst *bst = new Burst("Rafaga");
    bst->setRutaOArraySize(pqt->getRutaOArraySize());
    bst->setRutaEnlacesArraySize(pqt->getRutaEnlacesArraySize());

    for(int i=0;i<(int)ArraySize;i++){
        bst->setRutaEnlaces(i, pqt->getRutaEnlaces(i));//llenado del vector
ruta, con la puerta de salida correspondiente
        bst->setRutaO(i, pqt->getRutaO(i)); //llenado del vector ruta, con la
puerta de salida correspondiente
    }
    bst->setAlgoritmo(pqt->getAlgoritmo());
    bst->setIndexRuta(indexRuta);
    bst->setNumSaltos(saltos);
    bst->setNumPackets(numPacketsInBurst);
    bst->setLongitud(burstBits);
    bst->setSecNum(burstCounter);
    bst->setIdBurstifier(getId()); //retorna el índice del vector
correspondiente el módulo
    bst->setSourceAddress(sourceAddress);
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
bst->setDestAddress(destAddress);
bst->setKind(1);
bst->setTiempo_servicio(pqt->getTiempo_servicio());
bst->setLabel(destLabel);
bst->setName("Rafaga");
BurstifierInfo *info = new BurstifierInfo();
info->setLabel(destLabel); //se define en el .ini una etiqueta de destino
para cada burstifier
bst->setControlInfo(info);

EV<<"Numero de paquetes: "<<bst->getNumPackets()<<endl;
EV<<"Tamano de rafaga: "<<burstBytes<<endl;

//SenderID será establecido, una vez la ráfaga llegue a Sender
burstCounter++;

while(!(burst.isEmpty()))
{
    msgTemporal=(cMessage*)burst.pop();
    bst->insertMessage(msgTemporal);
}
//Iniciar contadores
burst.clear();
numPacketsInBurst=0;
int tamHeaderBits = tamHeader*8;
burstBits = tamHeaderBits;
firstPacket_t = 0;
//Una vez ensamblada la ráfaga, se mandan a la compuerta de salida.
send(bst,"out");
}
```

### 6. Algoritmo para planificación y envío de ráfagas

```
#include "Sender.h"

Define_Module(Sender);

Sender::Sender(){}

Sender::~Sender(){
    burst.clear();
    txQueue.clear();
    txQueue2.clear();
}

void Sender::initialize()
{
    numReceived=0;
    enviados=0;
}
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
void Sender::handleMessage(cMessage *msg)
{
    switch(msg->getKind())
    {
        case PETICION_LIGHTPATH:
        {
            bst = check_and_cast<Burst *>(msg); //guarda en mensaje en
mensaje_peticion, lanza una excepcion si msg es nulo
            origen = bst->getSourceAddress(); //se obtiene el nodo de ori
            destino = bst->getDestAddress(); //se obtiene el nodo de dest
            sprintf(mensaje, "Lightpath de %d a %d", origen, destino); //sale en la
simu

            bubble(mensaje); //mostrar en un bubble
            sendDirect(bst, getParentModule()->getParentModule()-
>getParentModule()->getParentModule()->getParentModule()-
>getSubmodule("nodoControl")->getSubmodule("WRON_Dinamica")-
>getSubmodule("nodo_control_dynamic_WRON"), "in"); //evio del mensaje directo al
nodo_control_dy..
            break;
        }
        case RUTA_Y_LONG_ONDA:
        {
            if (msg->getKind()==END_TX_MESSAGE){
                while(!txQueue.isEmpty()){
                    bst= (Burst *)txQueue.pop();
                }
            }
            else
            {
                ruta_y_long_encontrada = check_and_cast<RutaYLong *>(msg);
                origen = ruta_y_long_encontrada->getNodo_origen();
                destino = ruta_y_long_encontrada->getNodo_destino();

                if(ruta_y_long_encontrada->getBandera()==0)
                {
                    indexRuta=0;
                    saltos=0;
                    char pkname[40];
                    saltos++;
                    ArraySize = (unsigned int)ruta_y_long_encontrada-
>getRutaArraySize();
                    sprintf(pkname, "de %d -> %d, tamaño %llu bytes, salto %d",
origen, destino, ruta_y_long_encontrada->getLongitud()/8, saltos);
                    EV << "Rafaga " << pkname <<endl;
                    //--CONFIGURANDO LA RAFAGA-----//////////
                    bst= new Burst(pkname, BURST);
                    bst->setByteLength(ruta_y_long_encontrada->getLongitud()/8);
                }
            }
        }
    }
}
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
bst->setSourceAddress(origen);
bst->setDestAddress(destino);
bst->setRutaOArraySize(ArraySize);
bst->setRutaEnlacesArraySize(ArraySize);

for(int i=0;i<(int)ArraySize;i++){
    bst->setRutaO(i,ruta_y_long_encontrada-
>getIndexPuerta(i)); //llenado del vector ruta, con la puerta de salida
correspondiente
        bst->setRutaEnlaces(i,ruta_y_long_encontrada-
>getRuta(i));//llenado del vector ruta, con la puerta de salida correspondiente
    }
    indexRuta++;
    bst->setIndexRuta(indexRuta);
    bst->setNumSaltos(saltos);
    bst->setNumPackets(ruta_y_long_encontrada->getNumPackets());
    bst->setIdBurstifier(ruta_y_long_encontrada-
>getIdBurstifier()); //retorna el índice del vector correspondiente el módulo
    bst->setTiempo_servicio(ruta_y_long_encontrada-
>getTiempo_servicio());
    numOut = bst->getRutaO(indexRuta-1);
    cChannel *txChannel = gate("out", numOut)-
>getTransmissionChannel();
    simtime_t txFinishTime = txChannel-
>getTransmissionFinishTime();
    if (txFinishTime <= simTime())
    {
        send(bst, "out",numOut);
    }
    else
    {
        endTxMsg = new cMessage("endTxMsg",END_TX_MESSAGE);
        txQueue.insert(bst);
        scheduleAt(simTime()+txFinishTime, endTxMsg);
    }
    delete(ruta_y_long_encontrada);
    enviados++;
}
else{
    EV<<"Peticion bloqueada"<<endl;
}
}
break;
}

case BURST:
{
    if (msg->getKind()==END_TX_MESSAGE){
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
        while(!txQueue2.isEmpty()) {
            bst= (Burst *)txQueue2.pop();
        }
    }
    else {
        bst = check_and_cast<Burst *>(msg);
        indexRuta=bst->getIndexRuta(); //se extrae el indice de la ruta

        saltos=bst->getNumSaltos();
        origen=bst->getSourceAddress();
        destino= bst->getDestAddress();
        //en el caso de que no se cumpla la sig cvondicion va a seguir
        enviandose

        if(bst->getIndexRuta()<(bst->getRutaOArraySize())){
            EV<<"NODO INTERMEDIO"<<endl;
            saltos++;
            indexRuta++;
            numOut2 = bst->getRutaO(indexRuta-1);
            bst->setNumSaltos(saltos); //se aumenta el numero de saltos
            bst->setIndexRuta(indexRuta); //se aumenta el contador de
            posicion de la ruta puertas
            char pkname[40];
            sprintf(pkname, "de %d -> %d, tamaño %llu bytes, salto %d",
            origen, destino, bst->getByteLength(), saltos);
            EV << "Rafaga " << pkname <<endl;
            bst->setName(pkname);

            cChannel *txChannel2 = gate("out", numOut2)-
            >findTransmissionChannel();
            simtime_t txFinishTime;

            if (txChannel2!=nullptr){
                txFinishTime = txChannel2->getTransmissionFinishTime();
            }else{
                txFinishTime=simTime()+1;
            }

            if (txFinishTime <= simTime())
            {
                send(bst, "out",numOut2);
            }
            else
            {
                endTxMsg = new cMessage("endTxMsg",END_TX_MESSAGE);
                txQueue2.insert(bst);
                scheduleAt(simTime()+txFinishTime, endTxMsg);
            }
        }
    }
}
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
else{
    //Si entra a esta condicion se supone que llegó al destino
    int numOut3 = bst->getRuta0(indexRuta-1);
    EV<<"NODO DESTINO"<<endl;
    bubble("Recibido!");
    numReceived++;

    //-----CODIGO PARA LIBERAR LA RUTA-----//
    bst->setName("Liberacion de Lightpath");
    bst->setKind(LIBERAR_LIGHTPATH);
    send(bst, "out_control",numOut3);
}
}
break;
}
}
char buf[40];
cDisplayString& dispStrRecv= getParentModule()->getParentModule()-
>getParentModule()->getParentModule()->getDisplayString();
sprintf(buf, "R: %ld, S: %ld", numReceived, enviados);
dispStrRecv.setTagArg("t",0,buf);
}
```

### 7. Algoritmo para el acceder al estado de los enlaces de la red e imprimir tabla de enlaces en pantalla

```
#include "Estado.h"

Define_Module(Estado);

int Estado::numInitStages() const{return 1;}

Estado::~Estado()
{
    enlaces_en_topologia.clear();
    tabla_enlaces.clear();
    tiempos.clear();
    estado.clear();
}

void Estado::initialize(int stage)
{
    char estado_mostrar[10000];
    char auxiliar[100];
    int i,j,k; //indices
    int aux,auxdos;
    int identificador;
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
int indice_enlace;
cModule *submod;
cTopology::Node *nodo;
cTopology::Node *nodo_auxiliar;
int num_enlace_salida_nodo;

if(stage==0)
{
    num_nodos=(int)par("num_nodos");
    tam_slot=(double)par("tam_slot");
    bw_canal=(double)par("bw_canal");
    num_slots= bw_canal/tam_slot;

    netTopology.extractByNedTypeName((cStringTokenizer("EdgeAndCore2.EdgeCore
").asVector()));

    num_enlaces=0;

    for(i=0;i<num_nodos;i++) // se recorren los nodos para obtener el
numero de enlaces
    {
        submod = getParentModule()->getParentModule()-
>getSubmodule("edgeCore",i); //obtenemos los 14 nodos
        nodo= netTopology.getNodeFor(submod); // ahora se obtiene el
nodo para el grafo
        num_enlaces+= nodo->getNumOutLinks(); // num_enlaces e
signal al numero de links salientes dew cada nodo
    }

    tabla_enlaces.resize(num_enlaces); //inicializamos la tabla a cero

    for(i=0;i<(num_enlaces);i++)
    {
        tabla_enlaces[i].resize(5,0); // Se redimensiona a 4 y se
llena con 0ros
        tabla_enlaces[i][0]=i; // se llena cada fila en su posicion
0 con el numero del enlace
    }

    indice_enlace=0;
    int puerta;

    EV<<"-----"<<endl;
    EV<<"--Tabla de enlaces-- "<<endl;
    EV<<"-----"<<endl;

    for(i=0;i<num_nodos;i++)
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
{
    submod = getParentModule()->getParentModule()-
>getSubmodule("edgeCore",i);
    identificador=submod->getId(); //identificador para cada
valor de i identifica a uno de los core routers
    nodo= netTopology.getNodeFor(submod); // se obtiene el nodo
para el grafo
    aux= nodo->getNumOutLinks(); // aux almacena el numero de
conexiones de salida que tiene cada core router
    for(j=0;j<aux;j++) // para cada una de las conexiones de
salida
    {
        auxdos=nodo->getLinkOut(j)->getRemoteNode()-
>getModule()->getId(); // para cada nodo y cada una de sus conexiones de salida,
// auxdos
identifica el nodo remoto al que esta conectado.
        puerta=nodo->getLinkOut(j)->getLocalGate()-
>getIndex();
        tabla_enlaces[indice_enlace][1] = identificador;
        tabla_enlaces[indice_enlace][2] = auxdos;
        tabla_enlaces[indice_enlace][3] = nodo->getLinkOut(j)-
>getLocalGate()->getId();
        tabla_enlaces[indice_enlace][4] =j; //corresponde al
indice de la puerta por la que sale en cada nodo

        EV<<indice_enlace<<" : "<<getSimulation()-
>getModule(tabla_enlaces[indice_enlace][1])->getIndex()<<" --
"<<getSimulation()->getModule(tabla_enlaces[indice_enlace][2])-
>getIndex()<<endl; //
        indice_enlace++;
        //} // para cada enlace conocemos el origen y el
destino del mismo almacenado en tabla_enlaces
    }
}

// Inicializamos la tabla de estado
estado.resize(num_enlaces/num_slots); // la cantidad de filas
corresponde a 210 enlaces
for(i=0;i<(signed)estado.size();i++)estado[i].resize(num_slots,0); //
la cantidad de columnas al numero de long onda rellenas de ceros

// Inicializamos la tabla de tiempos de ocupacion
tiempos.resize(num_enlaces); // tiempos de ocu para cada enlace
for(i=0;i<(signed)tiempos.size();i++)tiempos[i].resize(1, 0.0); //
recorremos la matrix de tiempo

sprintf(estado_mostrar,"ESTADO DE LOS CANALES"); //en el buffer
estado_mostrar guarda lo que está entre comillas
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
        strcat(estado_mostrar, "\n\n");
        sprintf(auxiliar, "Links   ");
        strcat(estado_mostrar, auxiliar); //en el buffer estado_mostrar
guarda lo que está entre comillas
        strcat(estado_mostrar, "           ");
        sprintf(auxiliar, "Slots\n");
        strcat(estado_mostrar, auxiliar);
        strcat(estado_mostrar, "           ");

        for(i=0; i<num_slots; i++)
        {
            sprintf(auxiliar, "%4d", i);
            strcat(estado_mostrar, auxiliar); // concatena los indices de
las long de onda a estado mostrar
        }
        strcat(estado_mostrar, "\n\n");

        for(i=0; i<num_enlaces/num_slots; i++) // va cambiando cada
determinado numero de long de onda
        {
            sprintf(auxiliar, "%2d->%2d   ", getSimulation()-
>getModule(tabla_enlaces[i*num_slots][1])->getIndex(), getSimulation()-
>getModule(tabla_enlaces[i*num_slots][2])->getIndex());
            strcat(estado_mostrar, auxiliar);
            for(j=0; j<num_slots; j++) // va colocando los 0 de las long de
onda
            {
                >getCanvas();
                cCanvas *canvas = getParentModule()->getParentModule()-
                cRectangleFigure *rect = new cRectangleFigure("rect");

                rect->setBounds(cFigure::Rectangle(14400, 500, 450, 300));
                rect->setCornerRadius(2);
                rect->setLineColor(cFigure::Color("#000000"));
                rect->setLineWidth(1);
                rect->setFilled(true);
                rect->setFillColor(cFigure::Color("#ffffff"));
                canvas->addFigure(rect);

                sprintf(auxiliar, "%4d", estado[i][j]);
                strcat(estado_mostrar, auxiliar);
            }
            strcat(estado_mostrar, "\n");
        }

        cDisplayString& dispStr = getParentModule()->getParentModule()-
>getSubmodule("TABLA")->getDisplayString();
        dispStr.insertTag("t");
    }
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
dispStr.setTagArg("t",0,estado_mostrar);

    cDisplayString& dispStrdos = getParentModule()->getParentModule()-
>getSubmodule("PETICIONES")->getDisplayString();
    dispStrdos.insertTag("t");
    dispStrdos.setTagArg("t",0," ");

enlaces_en_topologia.resize(num_enlaces);

    for(i=0;i<netTopology.getNumNodes();i++)
    {
        nodo_auxiliar=netTopology.getNode(i);
        num_enlace_salida_nodo=nodo_auxiliar->getNumOutLinks();
        for(j=0;j<num_enlace_salida_nodo;j++)
        {

            if(nodo_auxiliar->getLinkOut(j)->isEnabled())
            {
                for(k=0;k<num_enlaces;k++)
                {
                    if((nodo_auxiliar->getModule()-
>getId()==tabla_enlaces[k][1])&&(nodo_auxiliar->getLinkOut(j)->getLocalGate()-
>getId()==tabla_enlaces[k][3]))
                    {
                        enlaces_en_topologia[k]=nodo_auxiliar->getLinkOut(j);
                    }
                }
                nodo_auxiliar->getLinkOut(j)->disable(); //se
desabilitan los enlaces
            }
        }
    }

bool Estado::checkChannelAvailability(int link,int wavelength, int slots)
{
    int count =0;
    for(int i=0;i<slots;i++){
        if(estados[link][wavelength+i]==0){
            count ++;
        }
    }
    if(count==slots){
        return(true);
    }
    else {
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
    return (false);  
  }  
}
```

### 8. Algoritmo para actualizar e imprimir en pantalla estado de enlaces de la red y calcular probabilidad de bloqueo

```
#include "Control_Dynamic_RWA.h"  
  
Define_Module(Control_Dynamic_RWA);  
  
int Control_Dynamic_RWA::numInitStages() const{return 2;}  
  
void Control_Dynamic_RWA::initialize(int stage) // se inicializan variables  
{  
    if(stage==1)  
    {  
        c_modulo_estado = getParentModule()->getParentModule()-  
>getParentModule()->getSubmodule("estado"); // se extrae la info del nodo estado  
        modulo_estado = check_and_cast<Estado *>(c_modulo_estado); // si  
c_modulo_estado es nullptr porque no encontró al submodulo llamado Estado o si  
c_modulo_estado no es de tipo estado se lanza una excepción con un mensaje  
apropiado  
  
        num_nodos=(int)par("num_nodos");  
        tam_slot=(double)par("tam_slot");  
        bw_canal=(double)par("bw_canal");  
        num_enlaces = modulo_estado->num_enlaces; // se extrae el numero de  
enlaces  
  
        recibidos=0;  
        perdidos=0;  
        retardo=0;  
        cuentaRetardo=0;  
        promedioRetardo=0;  
        probVec.setName("Probabilidad de bloqueo");  
        retardoVec.setName("Retardo Promedio");  
    }  
}  
  
void Control_Dynamic_RWA::handleMessage(cMessage *msg)  
{  
    //variables que me van a ayudar a manipular el mensaje, establecer rutas,  
etc  
    num_slots= bw_canal/tam_slot;  
    char estado_mostrar[10000];  
    char auxiliar[100];  
    char auxiliardos[500];  
    int i,j;
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
for(i=0;i<(int)num_slots*42;i++) //recorre cada uno de los enlaces
mediante un ciclo for
{ // esto permite limpiar el color de las rutas establecidas de nuevo a
negro, se ubica el modulo nodo con su indice y se atrapa en simulacion
  cDisplayString& dispStrcuatro = getParentModule()->getParentModule()-
>getParentModule()->getParentModule()->getSubmodule("edgeCore",getSimulation()-
>getModule(modulo_estado->tabla_enlaces[i][1])->getIndex()->gate(modulo_estado-
>tabla_enlaces[i][3])->getDisplayString();
  dispStrcuatro.parse("\ls=black,1");
}
switch (msg->getKind()) //revisamos la bandera del mensaje
{
  case PETICION_LIGHTPATH:
  {
    bubble("Peticion de LightPath Recibida");
    bst = check_and_cast<Burst *>(msg);
    origen = bst->getSourceAddress(); // se usaran para mostrar
en el mensaje
    destino = bst->getDestAddress();
    cDisplayString& dispStrdos = getParentModule()-
>getParentModule()->getParentModule()->getParentModule()-
>getSubmodule("PETICIONES")->getDisplayString();
    sprintf(auxiliardos,"PETICION %d->%d en
proceso\n",origen,destino); // se muestra mensaje, con nombre auxiliardos con
origen y dest
    dispStrdos.setTagArg("t",0,auxiliardos); //muestra en el
grafico el sprintf
    dispStrdos.setTagArg("t",2,"black"); // acomoda el color a
negro
    send(bst,"out_rwa"); //sale el mensaje hacia el modulo de
metodo rwa dinamico Shortest Path First Fit
    break;
  }

  case RUTA_Y_LONG_ONDA: // mensaje de regreso con ruta
  {
    ruta_y_long_encontrada = check_and_cast<RutaYLong *>(msg);
    origen = ruta_y_long_encontrada->getNodo_origen();
    destino = ruta_y_long_encontrada->getNodo_destino();
    burstBytes = ruta_y_long_encontrada->getLongitud()/8;
    //ENVIO POR MENSAJE DIRECTO AL NODO DE ORIGEN LA RUTA Y LONG
DE ONDA PARA QUE PUEDA TRANSMITIR

    if(ruta_y_long_encontrada->getBandera()==0)
    {
      sendDirect(ruta_y_long_encontrada,getParentModule()-
>getParentModule()->getParentModule()-
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
>getSubmodule("edgeCore",origen)->getSubmodule("edgeHost")-
>getSubmodule("Edge")->getSubmodule("edgeNodeAssembler")-
>getSubmodule("sender"),"in_control");
    recibidos++;

    for(i=0;i<(int)ruta_y_long_encontrada-
>getRutaArraySize();i++)
    {
        cDisplayString&dispStrtres = getParentModule()-
>getParentModule()->getParentModule()->getParentModule()-
>getSubmodule("edgeCore",getSimulation()->getModule(modulo_estado-
>tabla_enlaces[ruta_y_long_encontrada->getRuta(i)][1])->getIndex())-
>gate(modulo_estado->tabla_enlaces[ruta_y_long_encontrada->getRuta(i)][3])-
>getDisplayString();

        dispStrtres.parse("_ls=green,5");
        dispStrtres.setTagArg("t",3,"green");
        if(burstBytes<=15e6){
            modulo_estado-
>estado[(int)(ruta_y_long_encontrada-
>getRuta(i)/num_slots)][(int)ruta_y_long_encontrada->getRuta(i)%num_slots]=1;
        }
        else if(burstBytes<=900e6){
            modulo_estado->estado[(int)(ruta_y_long_encontrada-
>getRuta(i)/num_slots)][(int)ruta_y_long_encontrada->getRuta(i)%num_slots]=1;
            modulo_estado-
>estado[(int)(ruta_y_long_encontrada-
>getRuta(i)/num_slots)][(int)ruta_y_long_encontrada->getRuta(i)%num_slots+1]=1;
            modulo_estado-
>estado[(int)(ruta_y_long_encontrada-
>getRuta(i)/num_slots)][(int)ruta_y_long_encontrada->getRuta(i)%num_slots+2]=1;
        }
        else if(burstBytes<=12e9){
            modulo_estado-
>estado[(int)(ruta_y_long_encontrada-
>getRuta(i)/num_slots)][(int)ruta_y_long_encontrada->getRuta(i)%num_slots]=1;
            modulo_estado->estado[(int)(ruta_y_long_encontrada-
>getRuta(i)/num_slots)][(int)ruta_y_long_encontrada->getRuta(i)%num_slots+1]=1;
            modulo_estado->estado[(int)(ruta_y_long_encontrada-
>getRuta(i)/num_slots)][(int)ruta_y_long_encontrada->getRuta(i)%num_slots+2]=1;
            modulo_estado-
>estado[(int)(ruta_y_long_encontrada-
>getRuta(i)/num_slots)][(int)ruta_y_long_encontrada->getRuta(i)%num_slots+3]=1;
            modulo_estado-
>estado[(int)(ruta_y_long_encontrada-
>getRuta(i)/num_slots)][(int)ruta_y_long_encontrada->getRuta(i)%num_slots+4]=1;
        }
        else{
            EV<<"Valor Incorrecto"<<endl;
```



ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

```
    }  
    }  
    cDisplayString& dispStrdos = getParentModule()-  
>getParentModule()->getParentModule()->getParentModule()-  
>getSubmodule("PETICIONES")->getDisplayString();  
    sprintf(auxiliardos,"PETICION %d->%d  
ESTABLECIDA\n",origen,destino);  
    dispStrdos.setTagArg("t",0,auxiliardos);  
    dispStrdos.setTagArg("t",2,"green3");  
    }  
    else  
    {  
    origen = ruta_y_long_encontrada->getNodo_origen();  
    destino = ruta_y_long_encontrada->getNodo_destino();  
  
    cDisplayString& dispStrdos = getParentModule()-  
>getParentModule()->getParentModule()->getParentModule()-  
>getSubmodule("PETICIONES")->getDisplayString();  
    sprintf(auxiliardos,"PETICION %d->%d  
BLOQUEADA\n",origen,destino);  
    dispStrdos.setTagArg("t",0,auxiliardos);  
    dispStrdos.setTagArg("t",2,"red");  
    perdidos++;  
    if(perdidos!=0 && recibidos!=0){  
    pb = perdidos/(recibidos + perdidos);  
    probVec.record(pb);  
    EV<<"Probabilidad de bloqueo: "<<pb<<endl;  
    EV<<"Rafagas Perdidas: "<<(int)perdidos<<endl;  
    }  
    delete(ruta_y_long_encontrada);  
    }  
    break;  
    }  
}  
  
case LIBERAR_LIGHTPATH:  
{  
    bst = check_and_cast<Burst *>(msg);  
    origen = bst->getSourceAddress();  
    destino = bst->getDestAddress();  
    burstBytes = bst->getBitLength()/8;  
    retardo += bst->getRetardo();  
    cuentaRetardo++;  
    promedioRetardo = retardo/cuentaRetardo;  
  
    retardoVec.record(promedioRetardo);  
    EV<<"Retardo promedio: "<<promedioRetardo<<endl;
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
        cDisplayString& dispStrdos = getParentModule()-
>getParentModule()->getParentModule()->getParentModule()-
>getSubmodule("PETICIONES")->getDisplayString();
    sprintf(auxiliardos,"PETICION %d->%d LIBERADA\n",origen,destino);
    dispStrdos.setTagArg("t",0,auxiliardos);
    dispStrdos.setTagArg("t",2,"black");
    for(i=0;i<(int)bst->getRutaOArraySize();i++)
    {
        if(burstBytes<=15e6){
            modulo_estado->estado[(int)(bst-
>getRutaEnlaces(i)/num_slots)][(int)bst->getRutaEnlaces(i)%num_slots]=0;
        }
        else if(burstBytes<=900e6){
            modulo_estado->estado[(int)(bst-
>getRutaEnlaces(i)/num_slots)][(int)bst->getRutaEnlaces(i)%num_slots]=0;
            modulo_estado->estado[(int)(bst-
>getRutaEnlaces(i)/num_slots)][(int)bst->getRutaEnlaces(i)%num_slots+1]=0;
            modulo_estado->estado[(int)(bst-
>getRutaEnlaces(i)/num_slots)][(int)bst->getRutaEnlaces(i)%num_slots+2]=0;
        }
        else if(burstBytes<=12e9){
            modulo_estado->estado[(int)(bst-
>getRutaEnlaces(i)/num_slots)][(int)bst->getRutaEnlaces(i)%num_slots]=0;
            modulo_estado->estado[(int)(bst-
>getRutaEnlaces(i)/num_slots)][(int)bst->getRutaEnlaces(i)%num_slots+1]=0;
            modulo_estado->estado[(int)(bst-
>getRutaEnlaces(i)/num_slots)][(int)bst->getRutaEnlaces(i)%num_slots+2]=0;
            modulo_estado->estado[(int)(bst-
>getRutaEnlaces(i)/num_slots)][(int)bst->getRutaEnlaces(i)%num_slots+3]=0;
            modulo_estado->estado[(int)(bst-
>getRutaEnlaces(i)/num_slots)][(int)bst->getRutaEnlaces(i)%num_slots+4]=0;
        }
        else{
            EV<<"Valor incorrecto"<<endl;
        }
    }
    delete(bst);
    bubble("Lightpath Liberado");
    break;
}
};

    sprintf(estados_mostrar,"ESTADO DE LOS CANALES"); //en el buffer
estados_mostrar guarda lo que está entre comillas
    strcat(estados_mostrar,"\n\n");
    sprintf(auxiliar,"Links  ");
    strcat(estados_mostrar,auxiliar); //en el buffer estados_mostrar guarda lo que
está entre comillas
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
strcat(estado_mostrar, "          ");
sprintf(auxiliar, "Slots\n");
strcat(estado_mostrar, auxiliar);
strcat(estado_mostrar, "          ");

for(i=0; i<num_slots; i++)
{
    sprintf(auxiliar, "%4d", i);
    strcat(estado_mostrar, auxiliar);
}
strcat(estado_mostrar, "\n\n");

for(i=0; i<(int)num_enlaces/num_slots; i++)
{
    sprintf(auxiliar, "%2d->%2d  ", getSimulation()->getModule(modulo_estado->
tabla_enlaces[i*num_slots][1])->getIndex(), getSimulation()->
getModule(modulo_estado->tabla_enlaces[i*num_slots][2])->getIndex());
    strcat(estado_mostrar, auxiliar);
    for(j=0; j<num_slots; j++)
    {
        sprintf(auxiliar, "%4d", modulo_estado->estado[i][j]);
        strcat(estado_mostrar, auxiliar);
    }
    strcat(estado_mostrar, "\n");
}

cDisplayString& dispStr = getParentModule()->getParentModule()-
>getParentModule()->getParentModule()->getSubmodule("TABLA")-
>getDisplayString();
dispStr.setTagArg("t", 0, estado_mostrar);

for (i=0; i<(int)num_nodos; i++)
{
    cDisplayString& dispStrcinco = getParentModule()->getParentModule()-
>getParentModule()->getDisplayString();
    sprintf(auxiliar, " S=%d, P=%d,
T=%d", int(recibidos), int(perdidos), int(recibidos+perdidos));
    dispStrcinco.setTagArg("t", 0, auxiliar);
}
}
```

### 9. Algoritmo para el enrutamiento y asignación de espectro con restricción de continuidad de espectro



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

```
#include "Shortest_Path_First_Fit.h"

Define_Module(Shortest_Path_First_Fit);

Shortest_Path_First_Fit::~Shortest_Path_First_Fit(){
    burst.clear();
    path.clear();
    enlaces_en_topologia.clear();
}

int Shortest_Path_First_Fit::numInitStages() const{return 2;}

void Shortest_Path_First_Fit::initialize(int stage)
{
    cTopology::Node *nodo_auxiliar;
    cModule *c_modulo_estado;
    Estado *modulo_estado;

    int num_enlace_salida_nodo;
    int i,j,k;

    if(stage==1)
    {
        tam_slot=(double)par("tam_slot");
        bw_canal=(double)par("bw_canal");
        num_nodos=(int)par("num_nodos");

        // Se crea e inicia el grafo, creandose un objeto netTopology de tipo
cTopology
        c_modulo_estado = getParentModule()->getParentModule()-
>getParentModule()->getSubmodule("estado"); //se extrae info de estado
        modulo_estado = check_and_cast<Estado *>(c_modulo_estado); // se chequea
y se guarda info en modulo_estado

netTopology.extractByNedTypeName((cStringTokenizer("EdgeAndCore2.EdgeCore").asVe
ctor())); //se extrae la topologia por el nombre de tipo ned, cStringTokenizer
analiza la cadena de palabras y las analiza una por una, los convierte en un
vector cadena

        enlaces_en_topologia.resize(modulo_estado->num_enlaces); // cambia el
tamaño del vector al numero especificado de enlaces 42
        num_enlaces = modulo_estado->num_enlaces;
        for(i=0;i<(int)netTopology.getNumNodes();i++) //recorremos el numero de
nodos de la red
        {
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
nodo_auxiliar=netTopology.getNode(i); // extraemos el nodo actual y lo guardamos en nodo_auxiliar
num_enlace_salida_nodo=nodo_auxiliar->getNumOutLinks(); // numero de enlaces de salida del nodo_auxiliar

for(j=0;j<num_enlace_salida_nodo;j++) // recorremos los enlaces de salida de cada nodo
{
    if(nodo_auxiliar->getLinkOut(j)->isEnabled()) // si el link en cuestion esta disponible
    {
        for(k=0;k<(int)modulo_estado->num_enlaces;k++) // recorremos el numero de enlaces
        { // si el id del nodo_aux es igual al id del modulo en la tabla de enlace y si el id de la puerta del nodo_aux es igual al id de la puerta de salida en la tabla
            if((nodo_auxiliar->getModule()->getId()==modulo_estado->tabla_enlaces[k][1])&&(nodo_auxiliar->getLinkOut(j)->getLocalGate()->getId()==modulo_estado->tabla_enlaces[k][3]))
            {
                enlaces_en_topologia[k]=nodo_auxiliar->getLinkOut(j); // se extrae en un vector el enlace saliente de ese nodo
            }
        }
        nodo_auxiliar->getLinkOut(j)->disable(); // va deshabilitando los enlaces
    }
}

}

}

}

void Shortest_Path_First_Fit::handleMessage(cMessage *msg)
{
    num_slots= bw_canal/tam_slot;
    bandera=0;
    // int i,j,w;
    cModule *red_fisica = getParentModule()->getParentModule()->getParentModule()->getParentModule(); //obtenemos la NSFnet
    cModule *c_modulo_estado = getParentModule()->getParentModule()->getParentModule()->getSubmodule("estado");
    Estado *modulo_estado = check_and_cast<Estado *>(c_modulo_estado);

    //Para entrar en los metodos del estado de la red hay que utilizar:
    modulo_estado->"Metodo"
    switch (msg->getKind())
    {
        case PETICION_LIGHTPATH:
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
{
    bst = check_and_cast<Burst *>(msg);
    origen = bst->getSourceAddress();
    destino = bst->getDestAddress();
    int64_t burstBytes = bst->getLongitud()/8;
    int slots=0;

    if(burstBytes<=15e6){
        slots=1;
    }
    else if(burstBytes<=900e6){
        slots=3;
    }
    else if(burstBytes<=12e9){
        slots=5;
    }
    else{
        EV<<"Valor incorrecto"<<endl;
    }
    for(int i=0;i<(int)num_enlaces/num_slots;i++){// se recorren los
enlaces de la tabla estado (42)

        for(int w=0;w<=(int)num_slots;w++){           // 8/16

            if(modulo_estado-
>checkChannelAvailability(i,w,slots)==true){

                for(int j=0;j<(int)slots;j++){

                    enlaces_en_topologia[((int)(num_slots*i)+w)+j]-
>enable(); //habilito solo las longitudes de ond necesarias
//                    EV<<"Activados: "<<(num_slots*i+w)+j<<endl;
//multiplos 8/16
//                    EV<<"Enlaces Topologia Habilitados:
"<<enlaces_en_topologia[(num_slots*i+w)+j]<<endl;
                }
                break; //la primera que encuentre y luego sale del ciclo al
siguiente enlace
            }
        }
    }

    modulo_auxiliar = red_fisica->getSubmodule("edgeCore", origen );
    assert( modulo_auxiliar != NULL ); //debe ser diferente de null para
que no se lance un error
    origenod = netTopology.getNodeFor( modulo_auxiliar ); //obtenemos
el equivalente al nodo
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
assert( origenod != NULL ); //debe ser dif de null para que no
lance un error

modulo_auxiliar = red_fisica->getSubmodule("edgeCore", destino );
assert( modulo_auxiliar != NULL );
destinonod = netTopology.getNodeFor( modulo_auxiliar ); // se
obtiene el equivalente al nodo destino
assert( destinonod != NULL );

algoritmo = bst->getAlgoritmo();

if(strcmp(algoritmo, "Shortest_Path_First_Fit")==0){

    EV<<"Algoritmo Shortest Path First Fit"<<endl;

netTopology.calculateUnweightedSingleShortestPathsTo(destinonod); //calculo de
la ruta desde todos los nodos al destino
    longitud = origenod->getDistanceToTarget(); //se guarda en
longitud el número de saltos al destino
    if(origenod->getDistanceToTarget()!=HUGE_VAL ){
        path.resize( num_nodos ); // se redimensiona el vector path
        int j=0;
        while(origenod!=destinonod){
            path[j] = origenod->getPath(0); // en cada posicion se
guarda cada enlace de la ruta de un origen a un destino
            j++;
            origenod = origenod->getPath(0)-> getRemoteNode();
        }
    }
    else{//si yo hay ninguna forma de ruta hacia ese destino
        bandera=1;
    }
}
else if(strcmp(algoritmo, "AlgoritmoDRL")==0){

    EV<<"Algoritmo DRL"<<endl;
    SOCKET ConnectSocket = serversock::createConnection();
    serversock::sendValues(ConnectSocket,origen,destino);
    serversock::shutdownSocket(ConnectSocket);

    auto rutas = serversock::receiveData(ConnectSocket);
    int bytesRecibidos=rutas.bytesRecibidos;
    int len= (int)sizeof(rutas.rutasArray);
    int swap=0;

    if(origen>destino){

        std::swap(origen,destino);
```



ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON  
ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y  
ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

```
        swap=1;
    }

    for(int i=1;i<len;i++){
        if(rutas.rutasArray[2*i-1]=='0'){
            rutas.rutasArray[2*i-2]='0'+10;
        }else if(rutas.rutasArray[2*i-1]=='1'){
            rutas.rutasArray[2*i-2]='0'+11;
        }else if(rutas.rutasArray[2*i-1]=='2'){
            rutas.rutasArray[2*i-2]='0'+12;
        }else if(rutas.rutasArray[2*i-1]=='3'){
            rutas.rutasArray[2*i-2]='0'+13;
        }
    }

    int limite1 =0;
    int limite2 =0;
    int limite3 =0;
    int limite4 =0;
    int limite5 =0;

    for(int i=1;i<len;i++){
        if(origen==(int)rutas.rutasArray[2*i]-48){
            limite1=i;
            break;
        }
    }
    for(int i=limite1+1;i<len;i++){
        if(origen==(int)rutas.rutasArray[2*i]-48){
            limite2=i;
            break;
        }
    }
    for(int i=limite2+1;i<len;i++){
        if(origen==(int)rutas.rutasArray[2*i]-48){
            limite3=i;
            break;
        }
    }
    for(int i=limite3+1;i<len;i++){
        if(origen==(int)rutas.rutasArray[2*i]-48){
            limite4=i;
            break;
        }
    }
    for(int i=limite4+1;i<len;i++){
        if(origen==(int)rutas.rutasArray[2*i]-48){
            limite5=i;
        }
    }
}
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

```
        break;
    }
}
int_matrix_t T;
T.resize(6);
T[0].resize(limite1, -1);
T[1].resize(limite2-limite1, -1);
T[2].resize(limite3-limite2, -1);
T[3].resize(limite4-limite3, -1);
T[4].resize(limite5-limite4, -1);
T[5].resize(bytesRecibidos-T[0].size()-T[1].size()-T[2].size()-
T[3].size()-T[4].size(), -1);

for(int i=0;i<T[0].size();i++){
    T[0][i]=(int)rutas.rutasArray[2*i]-48;
}
for(int i=0;i<T[1].size();i++){
    T[1][i]=(int)rutas.rutasArray[2*(i+limite1)]-48;
}
for(int i=0;i<T[2].size();i++){
    T[2][i]=(int)rutas.rutasArray[2*(i+limite2)]-48;
}
for(int i=0;i<T[3].size();i++){
    T[3][i]=(int)rutas.rutasArray[2*(i+limite3)]-48;
}
for(int i=0;i<T[4].size();i++){
    T[4][i]=(int)rutas.rutasArray[2*(i+limite4)]-48;
}
for(int i=0;i<T[5].size();i++){
    T[5][i]=(int)rutas.rutasArray[2*(i+limite5)]-48;
}

if(swap==1){
    for(int i=0;i<T[0].size()/2;i++){
        std::swap(T[0][i],T[0][T[0].size()-i-1]);
    }
    for(int i=0;i<T[1].size()/2;i++){
        std::swap(T[1][i],T[1][T[1].size()-i-1]);
    }
    for(int i=0;i<T[2].size()/2;i++){
        std::swap(T[2][i],T[2][T[2].size()-i-1]);
    }
    for(int i=0;i<T[3].size()/2;i++){
        std::swap(T[3][i],T[3][T[3].size()-i-1]);
    }
    for(int i=0;i<T[4].size()/2;i++){
        std::swap(T[4][i],T[4][T[4].size()-i-1]);
    }
}
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
    }
    for(int i=0;i<T[5].size()/2;i++){
        std::swap(T[5][i],T[5][T[5].size()-i-1]);
    }
    for(int i=0;i<T[0].size();i++){
    }
    for(int i=0;i<T[1].size();i++){
    }
    for(int i=0;i<T[2].size();i++){
    }
    for(int i=0;i<T[3].size();i++){
    }
    for(int i=0;i<T[4].size();i++){
    }
    for(int i=0;i<T[5].size();i++){
    }
    std::swap(origen,destino);
    swap=0;
}

int filas = (int)T.size();
int columns = T[0].size();
int longarray[6];

for(int i=0;i<filas;i++){
    longarray[i]=(T[i].size())-1;
}

s=T[0][0];
d=T[0][1];
longitud=(int)longarray[0];
nslots=0;
m=0;
for(int i=0;i<longitud;i++){
    for(int j=0;j<num_enlaces/num_slots;j++)
    {
        if(s==getSimulation()->getModule(modulo_estado-
>tabla_enlaces[j*num_slots][1])->getIndex() && d==getSimulation()-
>getModule(modulo_estado->tabla_enlaces[j*num_slots][2])->getIndex())
        {
            path.resize(num_nodos);
            nslots=num_slots*j;
            int aux=0;
            int bloqueo=0;
            int count=round(num_slots/slots);
            for(int j=0;j<count;j++){
                if(checkAvailability(nslots+aux, slots)){
                    path[i-m]=enlaces_en_topologia[nslots+aux];
```





## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
        }
    }
}
if(m==1){longitud--;}
if(m==2){longitud-=2;}
if(m==3){longitud-=3;}
if(m==4){longitud-=4;}
if(m==5){longitud-=5;}
}
else{
    EV<<"Algoritmo Incorrecto"<<endl;
}

// Deshabilitar enlaces
for(int i=0;i<num_enlaces;i++)
{
    enlaces_en_topologia[i]->disable();
}

//para cada long de onda
paquete_ruta_y_longonda = new RutaYLong("Paquete con Ruta y Slots
Asignados", RUTA_Y_LONG_ONDA);
paquete_ruta_y_longonda->setNodo_origen(origen);
paquete_ruta_y_longonda->setNodo_destino(destino);
paquete_ruta_y_longonda->setTiempo_servicio(bst-
>getTiempo_servicio());
paquete_ruta_y_longonda->setIndexRuta(bst->getIndexRuta());
paquete_ruta_y_longonda->setSecNum(bst->getSecNum());
paquete_ruta_y_longonda->setNumSaltos(bst->getNumSaltos());
paquete_ruta_y_longonda->setNumPackets(bst->getNumPackets());
paquete_ruta_y_longonda->setLongitud(bst->getLongitud());
paquete_ruta_y_longonda->setIdBurstifier(bst->getIdBurstifier());
paquete_ruta_y_longonda->setLabel(bst->getLabel());
paquete_ruta_y_longonda->setBandera(bandera);

//ahora voy a establecer si hay algun elemento del vector en -1
if(bandera==0){// aqui se permite setear la ruta porque todas las
long de onda de la ruta se asignaron
    paquete_ruta_y_longonda->setRutaArraySize((unsigned
int)longitud);
    paquete_ruta_y_longonda->setIndexPuertaArraySize((unsigned
int)longitud);
    for(int i=0;i<longitud;i++) // se recorren los saltos
establecidos en la ruta
    {
        for(int j=0;j<num_enlaces;j++)
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
        {
            if(path[i]==enlaces_en_topologia[j]) // si cada salto de
la ruta es igual a un enlace en la topologia
            {
                paquete_ruta_y_longonda->setRuta(i,j); //se setea la
ruta
                paquete_ruta_y_longonda-
>setIndexPuerta(i,modulo_estado->tabla_enlaces[j][4]);
                j=num_enlaces; //sale del ciclo for en caso de que
se cumpla la condicion
            }
        }
    }

    send(paquete_ruta_y_longonda,"out_control_rwa");
    bandera=0;
    path.clear();
    enlaces_en_topologia.clear();
    burst.clear();
    delete(bst);
    break;
}

}

}

bool Shortest_Path_First_Fit::checkAvailability(int n, int slots)
{
    int count =0;
    for(int i=0;i<slots;i++){
        if(enlaces_en_topologia[n+i]->isEnabled()){
            count ++;
        }
    }
    if(count==slots){
        return(true);
    }
    else {
        return (false);
    }
}
```

10. Algoritmo principal en Python para el enrutamiento y asignación de espectro basado en DRL usando sockets.



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

```
import functools
import errno
import pfrl
import torch
import numpy as np

from server import obtenerDatos
from libs import evaluation, DeepRMSAv1_DNN, action_sampler, DRLAgent
from rsarl.envs import DeepRMSAEnv, make_multiprocess_vector_env
from rsarl.requester import UniformRequester
from rsarl.networks import SingleFiberNetwork
from rsarl.networks.topology import nsf

# Set the device id to use GPU. To use CPU only, set it to -1.
gpu = -1

if __name__ == '__main__':

    # exp settings
    n_requests = 10
    n_envs, seed = 5, 12

    # # build network
    net = SingleFiberNetwork("nsf", n_slot=100, is_weight=True)
    # # build requester
    requester = UniformRequester(
        net.n_nodes,
        avg_service_time=10,
        avg_request_arrival_rate=12)
    # # build env
    env = DeepRMSAEnv(net, requester)
    # # envs for training and evaluation

    try:

        envs = make_multiprocess_vector_env(env, n_envs, seed,
test=False)
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

```
test_envs = make_multiprocess_vector_env(env, n_envs, seed,
test=True)

except IOError as e:
    if e.errno == errno.EPIPE:
        pass
    # Handling of the error

# evaluation with test environment

evaluation = functools.partial(evaluation, envs=test_envs,
n_requests=n_requests)

    # Experimental Settings
K = 5
# slot-table(1) + one-hot-node * 2 + bandwidth(1)
ICH = 1 + 2 * net.n_nodes + 1
# build DNN for Q-function
q_func = DeepRMSAv1_DNN(net.n_slot, ICH, K, net.n_edges)
# Specify optimizer
optimizer = torch.optim.Adam(q_func.parameters(), eps=1e-2)

# Specify Explore and Replay Buffer

# random action function
action_sampler = functools.partial(action_sampler, k=K)

# Set the discount factor that discounts future rewards.
gamma = 0.99

# Use epsilon-greedy for exploration
explorer = pfrl.explorers.ConstantEpsilonGreedy(epsilon=0.1,
random_action_func=action_sampler)

# DQN uses Experience Replay.
# Specify a replay buffer and its capacity.
replay_buffer = pfrl.replay_buffers.ReplayBuffer(capacity=10 ** 6,
num_steps=50)

# Build DDQN
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

```
# Now create an agent that will interact with the environment.
DDQN = pfrl.agents.DoubleDQN(
    q_func,
    optimizer,
    replay_buffer,
    gamma,
    explorer,
    minibatch_size=50,
    update_interval=1,
    replay_start_size=500,
    target_update_interval=100,
    gpu=gpu,
)

#Step 2: Develop your algorithm by using KSPDRLAgent

agent = DRLAgent(k=5, drl=DDQN)
# prepare path table
agent.prepare_ksp_table(net)

#Step 3: Training and Evaluate DRL Agent

while True:
    obses = envs.reset()
    resets = [False for _ in range(len(obses))]
    conn,s,d=obtenerDatos()
    print("origen y destino: ",s,d)

    if s==0 and d==0:
        print("nada recibido")
    else:
        acts,p = agent.batch_act(obses,s,d,bandwidth)

        obses, rews, dones, infos = envs.step(acts)
        agent.batch_observe(obses, rews, dones,
resets,s,d,bandwidth)

        if s>d: f=np.flip(acts[0][0])
        else: f = acts[0][0]
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
        rutas=np.concatenate((f,p[0][0],p[0][1],p[0][2],p[0][3],p[0]
[4]),dtype='S2')
        print(rutas)
        print(len(rutas))
        conn.sendall(rutas)
        continue
```

### 11. Algoritmo para desensamblar ráfaga y hallar retardo extremo a extremo

```
#include "EdgeNodeDisassembler.h"

Define_Module(EdgeNodeDisassembler);

EdgeNodeDisassembler::~EdgeNodeDisassembler()
{
    //Limpia la cola de las ráfagas recibidas
    list<Burst*>::iterator iter_list;
    Burst *item;
    for(iter_list = receivedBursts.begin();iter_list != receivedBursts.end();
iter_list++){
        item = *iter_list;
        delete item;
    }
    receivedBursts.clear();
}

void EdgeNodeDisassembler::initialize()
{
    recvBursts = 0;
}

void EdgeNodeDisassembler::handleMessage(cMessage *msg)
{
    Burst *recvBurst = check_and_cast<Burst *>(msg);
    sourceAddress = recvBurst->getSourceAddress();

    //codigo para calcular el retardo extremo a extremo de la rafaga
    if (sourceAddress == 0) { pDelay = 0.0045*2;}
    else if (sourceAddress == 1) { pDelay = 0.0076*2;}
    else if (sourceAddress == 2) { pDelay = 0.0022*2;}
    else if (sourceAddress == 3) { pDelay = 0.0049*2;}
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
else if (sourceAddress == 4) { pDelay = 0.0041*2;}
else if (sourceAddress == 5) { pDelay = 0.0057*2;}
else if (sourceAddress == 6) { pDelay = 0.0069*2;}
else if (sourceAddress == 7) { pDelay = 0.0046*2;}
else if (sourceAddress == 8) { pDelay = 0.0095*2;}
else if (sourceAddress == 9) { pDelay = 0.0082*2;}
else if (sourceAddress == 10) { pDelay = 0.0103*2;}
else if (sourceAddress == 11) { pDelay = 0.0066*2;}
else if (sourceAddress == 12) { pDelay = 0.0072*2;}
else if (sourceAddress == 13) { pDelay = 0.0097*2;}

eed = simTime().dbl() - msg->getCreationTime().dbl() + pDelay;
EV<<"Retardo: "<<eed<<endl;

recvBurst->setRetardo(eed);

if (recvBurst->getKind()==LIBERAR_LIGHTPATH) //revisamos la bandera del
mensaje
{
    sendDirect(recvBurst,getParentModule()->getParentModule()-
>getParentModule()->getParentModule()->getSubmodule("nodoControl")-
>getSubmodule("WRON_Dinamica")->getSubmodule("nodo_control_dynamic_WRON"),"in");
}

receivedBursts.push_back(recvBurst);

//Adiciona la ráfaga en el contador de ráfagas recibidas
recvBursts++;

// updateDisplay(); //llama la funcion para mostrar el numero de rafagas
recibidas

int bId,nSeq; // valores de ID de la ráfaga

bId = recvBurst->getIdBurstifier();
nSeq = recvBurst->getSecNum();

list<Burst*>::iterator i;
Burst *actElem;
//Busca desde el comienzo porque la ráfaga que se está buscando es
probablemente una de la últimas
for(i = receivedBursts.begin(); i != receivedBursts.end(); i++){
    actElem = *i;
    if((actElem->getIdBurstifier() == bId) && (actElem->getSecNum() ==
nSeq)){
        break;
    } //Se encuentra la rafaga
}
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

```
}
if(i != receivedBursts.end()){ //Si el iterador superior no llega hasta el
final...significa que la ráfaga fue encontrada.
    Burst *burstIni= check_and_cast< Burst* > (*i);
    cMessage *tempPack;
    while(burstIni->hasMessages()){ //Libera los paquetes hasta que la
cola de burstIni esté vacía
        tempPack = burstIni->retrieveMessage();
        //Envía el paquete a un buffer intermedio por lo tanto, todos los
paquetes no serían liberados a la red eléctrica al mismo tiempo.
        send(tempPack,"out");
    }
    //Limpieza
    i = receivedBursts.erase(i);
}
else{
    printf("<OBS_BurstDisassembler><t=%s> Error, la rafaga id=(%d,%d) no
encontrada\n",simTime().str().c_str(),bId,nSeq);
    delete msg;
}
// }
}

void EdgeNodeDisassembler::updateDisplay()
{
    char buf[40];
    sprintf(buf, "Rafagas recibidas: %ld ", recvBursts);
    getDisplayString().setTagArg("t",0,buf);
}
}
```

### 12. Algoritmo para recibir paquetes en el destino

```
#include "Sink.h"

Define_Module(Sink);

//Simplente borra todos los mensajes que llegan a este destino

void Sink::initialize()
{
    // Initialize variables
    numReceived = 0;
};

void Sink::handleMessage(cMessage *msg)
{
```



## ANÁLISIS DE DESEMPEÑO DE UNA SDEON COMO UNA SOLUCIÓN DE RSA CON ALGORITMO CLÁSICO BASADO EN RESTRICCIÓN DE CONTINUIDAD DE ESPECTRO Y ALGORITMO COGNITIVO BASADO EN DEEP LEARNING

---

```
numReceived++;  
// updateDisplay();  
delete msg;  
};  
  
void Sink::updateDisplay()  
{  
    char buf[40];  
    sprintf(buf, "Paquetes recibidos: %ld", numReceived);  
    getDisplayString().setTagArg("t",0,buf);  
}
```