

ALGORITMO COGNITIVO PARA LA ASIGNACIÓN DE RECURSOS DE RED, EN UNA PLATAFORMA DWDM



SEBASTIÁN DAVID OSSA HERNÁNDEZ
JOSÉ ALEJANDRO VARGAS GUTIÉRREZ

Tesis de Pregrado en Ingeniería Electrónica y Telecomunicaciones

Director:

MsC. (c) Gustavo Adolfo Gómez

Codirector:

MsC. (c) Neil Andrés Jiménez Pinzón

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones

Popayán, 2019

SEBASTIÁN DAVID OSSA HERNÁNDEZ
JOSÉ ALEJANDRO VARGAS GUTIÉRREZ

ALGORITMO COGNITIVO PARA LA ASIGNACIÓN DE
RECURSOS DE RED, EN UNA PLATAFORMA DWDM

Tesis presentada a la Facultad de Ingeniería Electrónica y Telecomunicaciones de
la Universidad del Cauca para la obtención del Título de

Ingeniero en
Electrónica y Telecomunicaciones

Director:
MsC (c) Gustavo Adolfo Gómez

Codirector:
MsC (c) Neil Andrés Jiménez Pinzón

Popayán
2019

Contenido

Introducción.....	1
GENERALIDADES REDES ÓPTICAS PASIVAS, SDN Y COGNITIVIDAD.....	4
1.1 Aspectos Generales de las Redes Ópticas	4
1.1.1 Sistemas WDM	4
1.1.2 Técnicas de conmutación	7
1.1.3 Redes ópticas elásticas - EON (Elastic Optical Networks)	16
1.1.4 Routing and Spectrum Allocation.....	21
1.2 SDN.....	24
1.2.1 Redes SDN.....	24
1.2.2 Capa de aplicación	26
1.2.3 Capa de control	27
1.2.4 Capa de datos	27
1.2.5 SDN en redes ópticas.....	28
1.3 Metaheurísticas y Machine Learning.....	30
1.3.1 Metaheurísticas	30
1.3.2 Machine Learning	33
1.3.3 Cognitividad en redes de datos	36
1.3.4 Elementos básicos en una red cognitiva	39
ALGORITMOS, HERRAMIENTAS Y METODOLOGÍAS	41
2.1 Algoritmos Actuales de Redes Ópticas para la Asignación de Recursos de Red	41
2.1.1 RSA Estático.....	42
2.1.2 RSA Dinámico	42

2.1.3	Algoritmos RSA	43
2.2	Herramientas para el Desarrollo del Trabajo.....	44
2.2.1	MATLAB	44
2.2.2	OptSim.....	45
2.2.3	OMNeT++	45
2.2.4	Resumen de herramientas de simulación.....	46
2.3	Metodología de Trabajo	46
2.4	Metodología de Simulación	47
2.5	Definición del Escenario de Simulación	50
2.5.1	Nodo fuente-destino	51
2.5.2	Nodo Edge.....	51
2.5.3	Módulo de ingreso o ensamblador.....	52
2.5.4	Módulo de egreso o desensamblador.....	53
2.5.5	Nodo Core	54
2.6	Caracterización de la Red	55
DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO PARA LA ASIGNACIÓN DE RECURSOS DE RED		58
3.1	Algoritmos Implementados.....	58
3.1.1	Algoritmo Dijkstra.....	58
3.2	Centralización de la Red	63
3.2.1	Funcionamiento de la red centralizada	63
3.3	Inclusión del Algoritmo en la Red.....	66
3.3.1	Metaheurística implementada.....	71
3.4	Escenarios de Simulación	72
3.4.1	Caso de simulación 1	74
3.4.2	Caso de simulación 2	74
3.4.3	Sub casos de simulación	75
3.4.4	Casos de tercer nivel	75
3.4.5	Configuración de los parámetros de simulación	75
3.5	Ejecución de las Simulaciones	76
COMPARACIÓN, ANÁLISIS, RESULTADOS Y CONCLUSIONES		77
4.1	Análisis de los Resultados, Red distribuida.....	77
4.1.1	Caso red distribuida con ocho slots a 1.25 Gbps y 2.5 Gbps	77

4.1.2	Caso red distribuida con dieciséis slots a 1.25 Gbps y 2.5 Gbps	80
4.1.3	Caso red distribuida con ocho slots a 1.25 Gbps y 2.5 Gbps, estudio del retardo de extremo a extremo	82
4.1.4	Caso red distribuida con dieciséis slots a 1.25 Gbps y 2.5 Gbps, estudio del retardo de extremo a extremo.....	84
4.2	Análisis de los Resultados, Red Centralizada.....	86
4.2.1	Caso red centralizada con ocho slots a 1.25 Gbps y 2.5 Gbps	86
4.2.2	Caso red centralizada con dieciséis slots a 1.25 Gbps y 2.5 Gbps ...	88
4.2.3	Caso red centralizada con ocho slots a 1.25 Gbps y 2.5 Gbps, estudio del retardo extremo a extremo	90
4.2.4	Caso red centralizada con dieciséis slots a 1.25 Gbps y 2.5 Gbps, estudio del retardo extremo a extremo.....	91
4.3	Conclusiones y Recomendaciones	94
4.3.1	Conclusiones respecto a los resultados obtenidos.....	94
4.3.2	Conclusiones respecto al desarrollo del trabajo de grado	95
4.3.3	Conclusiones respecto a las herramientas empleadas.....	95
4.3.4	Recomendaciones	95
4.4	Trabajos Futuros	96
REFERENCIAS.....		97
Anexo A: Tiempo entre Generación de Paquetes en la Herramienta OMNeT++		103
Anexo B: Compendio de los Resultados Obtenidos en Todos los Casos de Estudio.		108

LISTA DE FIGURAS

Figura 1.1 - Comparación entre CWDM y DWDM.....	5
Figura 1.2 - Ejemplo de sistema DWDM	6
Figura 1.3 - Vista típica de una red óptica.....	7
Figura 1.4 - Nodo OXC.....	8
Figura 1.5 - Esquema funcional de una red OBS.....	9
Figura 1.6 - Estructura de un paquete OPS	12
Figura 1.7 - Nodo OPS.....	14
Figura 1.8 - Arquitectura de una EON.....	16
Figura 1.9 - Áreas de investigación de Flexgrid	17
Figura 1.10 - Rejilla WDM (a), canales separados 100 GHz, (b), canales separados 50 GHz, (c), canales separados 25 GHz	18
Figura 1.11 - Ejemplo de concepto de ancho de ranura.....	19
Figura 1.12 - Señal OFDM en el dominio de la frecuencia.....	20
Figura 1.13 - (a) Respuesta en frecuencia, (b) Respuesta en el tiempo, de una señal N-WDM.....	21
Figura 1.14 - Rejilla de espectro óptico en RSA.....	22
Figura 1.15 - RSA semi-elástico.....	23
Figura 1.16 - Expansión/Contracción en RSA.....	24
Figura 1.17 - Reasignación de espectro en RSA	24
Figura 1.18 - Arquitectura SDN	25
Figura 1.19 - Planos de SDN.....	26
Figura 1.20 - Modelo de arquitectura para SDON	29
Figura 1.21 - Representación esquemática de un transceptor óptico ajustable	30
Figura 1.22 - Clasificación de las metaheurísticas más usuales	31
Figura 1.23 - Clasificación de metaheurísticas.....	32
Figura 1.24 - Ejemplos de fuentes de inspiración para las metaheurísticas.....	33
Figura 1.25 - Aprendizaje activo con refuerzo	34
Figura 1.26 - Clasificación de arquitecturas cognitivas	36
Figura 1.27 - Ciclo cognitivo.....	37
Figura 1.28 - Ciclo cognitivo en redes cognitivas	38
Figura 1.29 - Representación de los procesos cognitivos por planos	39

Figura 2.1 - Logo de MATLAB.....	44
Figura 2.2 - Modelo en cascada.....	46
Figura 2.3 - Diagrama de flujo de la simulación.....	49
Figura 2.4 - Modulo de nodo Core y Edge.....	50
Figura 2.5 - Nodo fuente-destino en OMNeT++.....	51
Figura 2.6 - Nodo Edge en OMNeT++.....	52
Figura 2.7 - Módulo ensamblador visto en OMNeT++.....	53
Figura 2.8 - Nodo Core.....	54
Figura 2.9 - Unidad de control.....	55
Figura 2.10 - Red NSFNet.....	56
Figura 2.11 - Red NSFNet implementada en OMNeT++.....	56
Figura 3.1 - Ejemplo de.....	59
Figura 3.2 - Diagrama de flujo de LAUC adaptado.....	61
Figura 3.3 - Diagrama de flujo del proceso de centralización de la red.....	64
Figura 3.4 - Composición de la cabecera PCC.....	65
Figura 3.5 - Ejemplo de operación del algoritmo péndulo.....	67
Figura 3.6 - Diagrama de flujo del algoritmo péndulo.....	68
Figura 4.1a - Pb con 8 slots y velocidad de 1.25 Gbps.....	78
Figura 4.2b - Pb con 8 slots y velocidad de 2.5 Gbps.....	79
Figura 4.3a - Pb con 16 slots y velocidad de 1.25 Gbps.....	81
Figura 4.4b - Pb con 16 slots y velocidad de 2.5 Gbps.....	81
Figura 4.5a - Retardo de extremo a extremo, 8 slots y velocidad de 1.25 Gbps... ..	82
Figura 4.6b - Retardo de extremo a extremo, 8 slots y velocidad de 2.5 Gbps.....	83
Figura 4.7a - Retardo de extremo a extremo, 16 slots y velocidad de 1.25 Gbps.	84
Figura 4.8b - Retardo de extremo a extremo, 16 slots y velocidad de 2.5 Gbps... ..	85
Figura 4.9a - Pb con 8 slots y velocidad de 1.25 Gbps.....	86
Figura 4.10b - Pb con 8 slots y velocidad de 2.5 Gbps.....	87
Figura 4.11a - Pb con 16 slots y velocidad de 1.25 Gbps.....	88
Figura 4.12b - Pb con 16 slots y velocidad de 2.5 Gbps.....	89
Figura 4.13a - Retardo de extremo a extremo, 8 slots y velocidad de 1.25 Gbps.	90
Figura 4.14b - Retardo de extremo a extremo, 8 slots y velocidad de 2.5 Gbps... ..	90
Figura 4.15a - Retardo de extremo a extremo, 16 slots y velocidad de 1.25 Gbps.....	91
Figura 4.16b - Retardo de extremo a extremo, 16 slots y velocidad de 2.5 Gbps.	92
Figura A.1a – Función de densidad de probabilidad (pdf) para cada carga de tráfico empleada.....	105
Figura A.2b – Función de distribución de probabilidad (cdf) para cada carga de tráfico empleada.....	106

LISTA DE TABLAS

Tabla 1.1 - Comparación de los esquemas de contención.....	11
Tabla 1.2 - Comparación entre las técnicas de conmutación.....	15
Tabla 1.3 - Resumen de formulación de problemas en aprendizaje de máquinas	35
Tabla 2.1 - Comparación de herramientas de simulación.....	46
Tabla 3.1 - Escenarios de simulación propuestos.....	73
Tabla 3.2 - Escenarios de simulación propuestos.....	74
Tabla 3.3 - Parámetros de simulación.....	75
Tabla 3.4 - Descripción de los equipos de cómputo usados.....	76
Tabla 4.1 - Parámetros de simulación, casos 1 y 2, subcaso 1.....	78
Tabla 4.2 - Parámetros de simulación, casos 1 y 2, subcaso 2.....	80
Tabla 4.3 - Parámetros de simulación, casos 1 y 2, subcaso 1, red centralizada .	86
Tabla 4.4 - Parámetros de simulación, casos 1 y 2, subcaso 2, red centralizada .	88
Tabla 4.5 - Síntesis de los resultados de la red distribuida.....	93
Tabla 4.6 - Síntesis de los resultados de la red centralizada.....	93
Tabla B.1a – Resultados obtenidos para todos los casos de simulación en la red distribuida.....	109
Tabla B.2b – Resultados obtenidos para todos los casos de simulación en la red centralizada.....	110

LISTA DE ECUACIONES

Ecuación (1.1)	18
Ecuación (1.2)	18
Ecuación (1.3)	19
Ecuación (3.1)	72

LISTA DE ACRÓNIMOS

AI	<i>Artificial intelligence – Inteligencia Artificial</i>
API	<i>Application Programming interface – Interfaz de programa de aplicación</i>
BCO	<i>Bee Colony Optimization – Optimización por Colonia de Abejas</i>
BCP	<i>Burst Control Packet – Paquete Controlador de Ráfagas</i>
CF	<i>Central Frequency – Frecuencia Central</i>
CWDM	<i>Coarse Wavelength Division Multiplexing – Multiplexación Amplia por División de Longitud de onda</i>
DR	<i>Dispersión Residual</i>
DWDM	<i>Dense Wavelength Division Multiplexing – Multiplexación Densa por División de Longitud de onda</i>
EON	<i>Elastic Optical Networks – Redes ópticas elásticas</i>
FDL	<i>Fiber Delay Lines – Líneas de Fibra de Retardo</i>
FDM	<i>Frequency Division Multiplexing – Multiplexación por División de Frecuencia</i>
FS	<i>Frequency Slot – Frecuencia de ranura</i>
FWM	<i>Four Wave Mixing – Mezcla de Cuatro Ondas</i>
HP	<i>Header Processing – Procesamiento de Cabecera</i>

IP	<i>Internet Protocol – Protocolo de Internet</i>
ITU	<i>International Telecommunication Union – Unión Internacional de las Telecomunicaciones</i>
LAN	<i>Local Area network – Red de Área Local</i>
LP-SA	<i>Link-Path-Slot-Assignment – Asignación de Ranura de Ruta de Enlace</i>
MAC	<i>Media Access Control – Control de Acceso al Medio</i>
MAN	<i>Metropolitan Area Network – Red de Área Metropolitana</i>
ML	<i>Machine Learning – Aprendizaje de Máquina</i>
NBI	<i>Northbound Interface – Interfaz hacia el norte</i>
OAN	<i>Optical Acces Network – Red de Acceso Óptico</i>
OBS	<i>Optical Burst Switching – Conmutación Óptica de Ráfagas</i>
OCS	<i>Optical Circuit Switching – Conmutación Óptica de Circuitos</i>
OFDM	<i>Orthogonal Frequency-Division Multiple Access</i>
OPS	<i>Optical Packet Switching – Conmutación Óptica de Paquetes</i>
OSI	<i>Open System Interconnection – Interconexión de Sistemas Abiertos</i>
OTN	<i>Optical transport network – Red de Transporte Óptico</i>
OXC	<i>Optical Cross-Conect – Conector Óptico cruzado</i>
PB	<i>Probabilidad de Bloqueo</i>
PCE	<i>Path Computation Element – Elemento de computación de ruta</i>
PCC	<i>Paquete Controlador de Paquetes</i>
PD	<i>Packet Delination – Delineación de Paquete</i>
QoS	<i>Quality of Service – Calidad del Servicio</i>

QoT	<i>Quality of Transmission – Calidad de la Transmisión</i>
RSA	<i>Routing and Spectrum Allocation – Asignación de Direccionamiento y Espectro</i>
RWA	<i>Routing and Wavelength Assignment – Asignación de Direccionamiento y Longitud de Onda</i>
SA	<i>Spectrum Allocation – Asignación de espectro</i>
SDH	<i>Synchronous Digital Hierarchy – Jerarquía Digital Síncrona</i>
SDN	<i>Software Defined Networking – Redes Definidas por Software</i>
SDON	<i>Software Defined Optical Networking – Redes Ópticas definidas por Software</i>
SW	<i>Slot Width – Ancho de Ranura</i>
TDM	<i>Time Division Multiplexing – Multiplexación por División de Tiempo</i>
TCP	<i>Traffic Control Protocol – Protocolo de control de tráfico</i>
TG	<i>Traffic Grooming – Tráfico Granulado</i>
UDP	<i>User Datagram Protocol – Protocolo de datagrama de usuario</i>
VLAN	<i>Virtual Local Area Network – Red de Área Local Virtual</i>
WAN	<i>Wide Area Network – Red de Área Amplia</i>
WC	<i>Wavelength Conversion – Conversión de Longitud de Onda</i>
WDM	<i>Wavelength Division Multiplexing – Multiplexación por División de Longitud de Onda</i>

Introducción

Las redes ópticas DWDM (multiplexación densa por división de longitud de onda) funcionan asignando longitudes de onda a los diferentes usuarios o servicios, en una porción estrecha del espectro óptico, lo que las hace más eficientes comparadas con redes CWDM (multiplexación amplia por división de longitud de onda). Actualmente el control de las redes se encuentra mayormente distribuido, esto quiere decir que, son los diferentes dispositivos que componen la red los que se encargan de hacer la asignación de recursos, el enrutamiento, la gestión de tráfico, entre otros, lo que hace que lograr una gestión eficiente de la red sea difícil de conseguir. SDN (redes definidas por software) busca solucionar este problema, centralizando el control de los recursos de red en un único lugar, mediante una visión holística, mejorando la eficiencia y abaratando costos al disminuir los ciclos de procesamiento y decisión de los dispositivos de red actuales.

Para la gestión y asignación de recursos de red, las redes ópticas actuales disponen de una serie de algoritmos estáticos que se encargan de controlar el flujo de la información, incluyendo la asignación de las frecuencias de las diferentes portadoras del sistema, esto, si bien permite el funcionamiento del sistema, no es muy eficiente, pues normalmente para la transmisión de un nuevo flujo de datos, independientemente del tipo de tráfico a transmitir (Voz, datos, videos), el sistema asigna la primera longitud de onda que se encuentre disponible, que podría no ser la más adecuada, ocasionando una disminución en la calidad del enlace óptico.

La gestión de recursos de red se logra en mayor medida, implementando algoritmos rígidos o estáticos distribuidos en los diferentes equipos o dispositivos que conforman la estructura de la red DWDM, estos se encargan de gestionar el tráfico, usualmente de forma independiente al tipo de información transmitida, lo que limita las capacidades de la red y aumenta los costos de operación al tener que introducir software complejo en cada dispositivo. SDN es una arquitectura emergente que permite ejercer un control total y centralizado en las redes actuales de información. SDN tiene muchos beneficios, ya que es posible asignar recursos según la demanda de tráfico, lo que hace que la red sea totalmente flexible y escalable, además de optimizar las inversiones en cuanto a infraestructura de

forma efectiva, permitiendo la introducción de nuevos algoritmos de gestión, más complejos y potentes, mejorando aún más la eficiencia de la red DWDM.

Los algoritmos cognitivos, basados en Machine Learning (aprendizaje de máquinas) son un campo de la inteligencia artificial que permiten que una computadora “aprenda”, logrando analizar el comportamiento de un sistema y aprender de él para estimar cuál es la mejor forma de ejecutar las tareas para las que fue diseñada. Dentro de esta área existen muchos tipos de algoritmos que se aplican en diferentes contextos, dependiendo de la solución a la que se quiera llegar o la cantidad de recursos disponibles.

Es por estos motivos que se plantea la idea de diseñar un algoritmo cognitivo, cuya función principal será la de asignar dinámicamente los recursos de una red óptica, específicamente la longitud de onda de la portadora sobre la cual se va a transmitir un servicio, permitiendo así que cada flujo de datos viaje sobre el mejor canal disponible, teniendo en cuenta parámetros como la diferenciación del tráfico entrante, la capacidad y el ancho de banda requerido, entre otros; donde el algoritmo operará de forma centralizada, heredando características y beneficios de la arquitectura SDN.

Organización del documento

En el **capítulo 1**, se detallarán aspectos básicos de la teoría que enmarca a las redes ópticas, las redes centralizadas definidas por software y el aprendizaje de máquinas, buscando ofrecer al lector un soporte para entender los conceptos que se tratarán durante el desarrollo del trabajo de investigación.

En el **capítulo 2**, se describirán algunos de los algoritmos utilizados actualmente para la asignación de espectro en redes ópticas, para posteriormente elegir aquellos que puedan ser de utilidad. También se hará un recuento y descripción de las herramientas software disponibles (Matlab, OptSim, OMNeT) para el desarrollo de este trabajo de grado, finalmente se hará la elección de una de ellas. Para cerrar el capítulo, se hablará acerca de la metodología de trabajo escogida y su aplicación al trabajo de grado.

En el **capítulo 3**, se tratará acerca del proceso de centralización de la red, los algoritmos que operan en la red, el diseño e implementación del algoritmo de asignación de espectro propuesto y los casos de simulación para evaluar su desempeño.

En el **capítulo 4**, se estudiarán los resultados obtenidos en la ejecución de las simulaciones (ver anexo B), también se hablará de las conclusiones encontradas al analizar los resultados y finalmente se plantearán algunos posibles trabajos futuros, extendiendo así esta línea de investigación.

Capítulo 1

GENERALIDADES REDES ÓPTICAS PASIVAS, SDN Y COGNITIVIDAD

En este capítulo se detallarán aspectos básicos de la teoría que enmarca a las redes ópticas, las redes centralizadas definidas por software y el aprendizaje de máquinas, buscando ofrecer al lector un soporte para entender los conceptos que se tratarán durante el desarrollo del trabajo de investigación.

Para el desarrollo de este capítulo, se toman como referencia libros, artículos e investigaciones realizadas a nivel mundial sobre los diferentes estándares de gestión de redes ópticas, SDN y Cognitividad, con el fin de dar un soporte científico y actualizado al trabajo de investigación.

1.1 Aspectos Generales de las Redes Ópticas

Gracias a los avances de las diferentes tecnologías que componen las redes ópticas, estas han llegado a convertirse en el principal soporte de las redes de telecomunicaciones actuales. Aspectos como las tecnologías de multiplexación, conmutación, entre otros, serán tratados en las siguientes secciones.

1.1.1 Sistemas WDM

Las redes ópticas de transporte u OTN (Optical Transport Network), son básicamente cualquier red de datos que haga uso de la tecnología óptica para permitir el envío de información sobre enlaces de fibra óptica, al mismo tiempo que se encarga de la conmutación, direccionamiento y verificación de la información [1]. La necesidad de ampliar la capacidad de las OTN para lograr el envío de más

información sobre el mismo hilo de fibra impulsó la aparición de varias técnicas de multiplexación, entre ellas WDM (Wavelength División Multiplexing), la cual permite que varias portadoras ópticas pueden viajar por un mismo hilo de fibra. Cada portadora posee una longitud de onda o lambda diferente, para lograr esto, se hace uso de un dispositivo denominado multiplexor en la entrada del sistema, encargado de combinar las diferentes señales ópticas dentro de la fibra, por otra parte, en el lado del receptor, se hace uso de un demultiplexor óptico, encargado de separar las señales en sus respectivas longitudes de onda originales. Es posible, en el dominio eléctrico de las señales ópticas, aplicar otras técnicas de multiplexación como TDM o FDM [2].

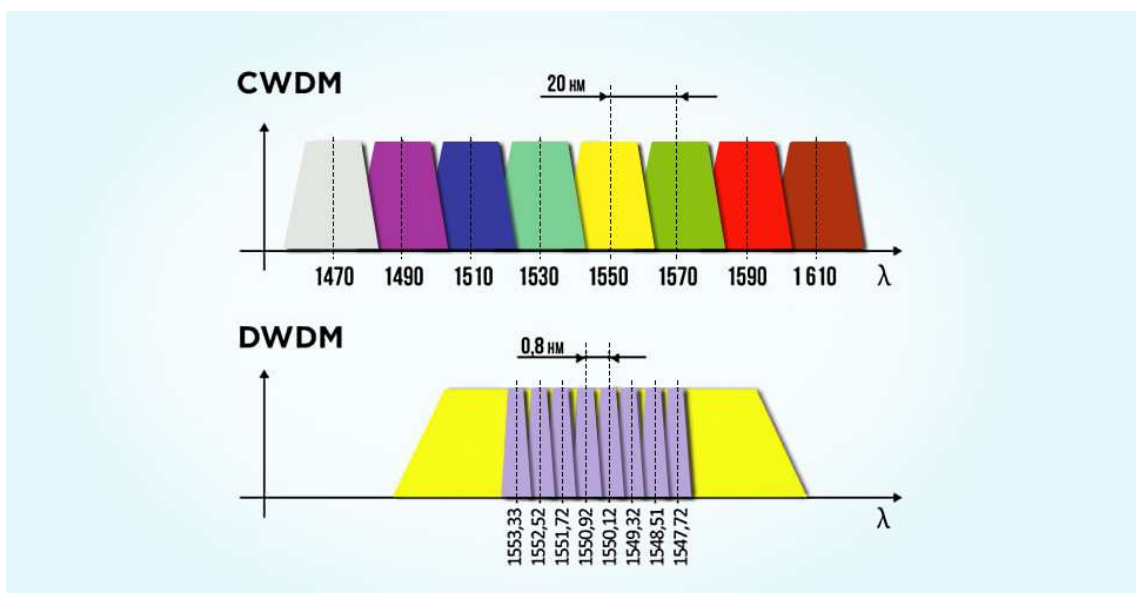


Figura 1.1 - Comparación entre CWDM y DWDM [3]

La rápida evolución de los sistemas ópticos permitió ir desde la multiplexación de unos pocos canales ópticos hasta el envío de decenas de canales sobre el mismo hilo, pasando por los sistemas CWDM (Coarse Wavelength División Multiplexing) hasta llegar a lo que hoy se conoce como sistemas DWDM (Dense Wavelength División Multiplexing), capaces de transmitir información en múltiples longitudes de onda a diferentes velocidades de datos y por largas distancias [4], siendo esta la tecnología de mayor interés para el desarrollo del trabajo de grado. La figura 1.1 muestra una comparación entre sistemas CWDM y DWDM.

Las redes ópticas DWDM funcionan asignando longitudes de onda en la ventana de transmisión C del espectro óptico, con un espaciado de 100 GHz, 50 GHz, 25 GHz y hasta 12,5 GHz entre los canales, por lo que posee una alta capacidad de transmitir información a muy largas distancias [5], a diferentes usuarios con diferentes servicios.



Figura 1.2 - Ejemplo de sistema DWDM [6]

DWDM ha permitido el desarrollo de grandes redes de datos que se encargan de soportar las telecomunicaciones del mundo moderno. Se encuentra en las ciudades como redes metropolitanas con topología de anillo o también cubriendo largas distancias con enlaces punto a punto. Dado la facilidad para transmitir grandes volúmenes de información, además que las redes ópticas DWDM soportan todos los protocolos de datos, pudiendo así transmitir servicios de como voz, datos o videos, ya sea en tiempo real o no. El poder soportar el protocolo IP (Internet Protocol), junto con la relativa facilidad de instalación, la gran cobertura de las DWDM y su alta capacidad de transmitir información han permitido que este tipo de redes sean un buen soporte para NGN (Next Generations Networking) [7]. En la figura 1.2 se aprecia el esquema básico de un sistema DWDM. Los sistemas de fibra óptica DWDM, pueden clasificarse en tres categorías [1, 2], según su modo de aplicación y área de cobertura, lo que determina también la topología más recomendada a usar.

Los sistemas DWDM encargados de conectar puntos que se encuentran a grandes distancias son denominados "Sistemas punto a punto" y se caracterizan por ser enlaces de miles de kilómetros, conformando lo que se conoce como "WAN" (Wide Area Network), sobre los cuales viajan grandes cantidades de información. Generalmente la topología "punto a punto" es implementada para estos casos, ya que permite un alto flujo de datos a gran velocidad con relativa sencillez. La interconexión de estos sistemas da lugar a diversas topologías [4].

Otro tipo de sistemas ampliamente utilizados son las denominadas "Redes de distribución", también conocidas como "MANs" (Metropolitan Area Network), encargadas de transportar información en un área geográfica bastante grande, pero no tanto para llegar a ser redes WAN. Las MAN, generalmente hacen uso de la topología en anillo, lo que les permite cubrir una gran área y añadir o extraer información fácilmente mediante la implementación de multiplexores "Add drop". Aunque también la topología en "Bus" es implementada por su facilidad para llevar la misma información a muchos usuarios [8].

Por último, las “Local Area Network” o LAN, son redes encargadas de transmitir información en un área geográfica limitada, como campus universitarios o zonas de oficinas. Las LAN ópticas suelen configurarse en configuración bus, anillo o estrella, siendo esta última una de las más implementadas debido a que la generación de datos de cada usuario es aleatoria [2].

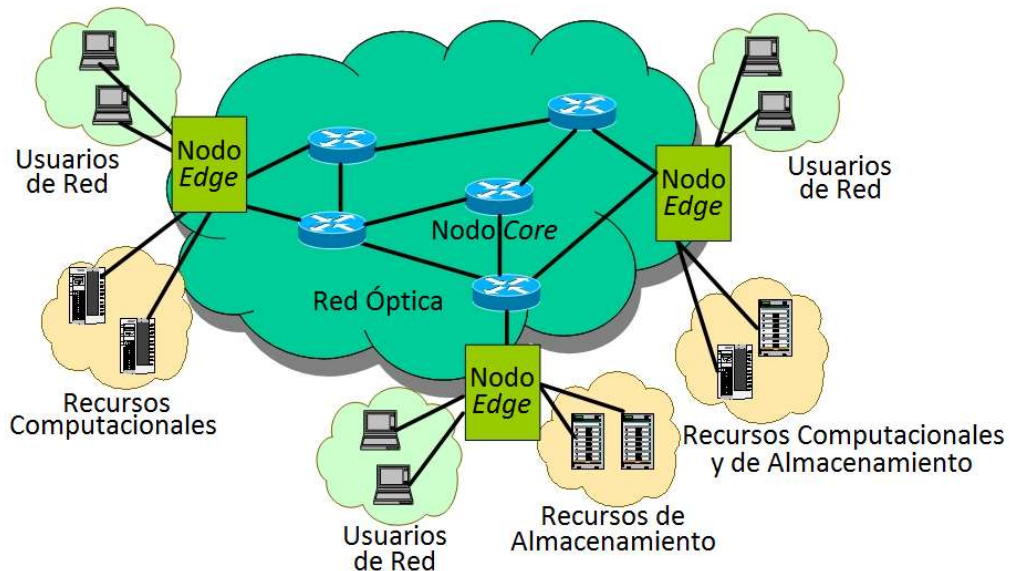


Figura 1.3 - Vista típica de una red óptica [9]

En la Figura 1.3, se puede apreciar cómo es una red óptica típica, que hace uso de diferentes topologías de red. Se observan algunos de los diferentes dispositivos que conforman la red, como enrutadores, nodos de borde y de núcleo, equipos de usuarios y servidores, cada uno de estos equipos cuenta con software especializado para la gestión de la red, que, aunque funcional, dificulta su administración¹, pues no es posible realizar una gestión centralizada y así aumentar la eficiencia, además, la inclusión de software especializado en cada equipo de red aumenta el costo de la misma e incrementa la probabilidad de fallos aislados en el sistema.

1.1.2 Técnicas de conmutación

La conmutación es la técnica por la cual se establece la ruta entre dos puntos de una red de datos. Dentro de las redes ópticas se pueden encontrar tres tipos, Conmutación eléctrica de circuitos OCS, conmutación óptica de ráfagas OBS y la conmutación óptica de paquetes OPS [10]. De las que se hablarán con mayor detenimiento a continuación:

¹ Se refiere la óptima utilización de los recursos de los que dispone la red, con el fin de lograr el mejor desempeño en relación costo-eficiencia.

- Conmutación eléctrica de circuitos

Optical circuit switching o OCS, permite la conmutación de datos entre dos nodos, denominados OXC, mediante el establecimiento de una ruta óptica entre ellos durante todo el tiempo que dure la transmisión, siendo necesaria la conversión óptico-eléctrica-óptica en cada uno de los nodos intermedios de la red. Lo que la hace lenta debido al tiempo de establecimiento de las rutas, el tiempo de conversión de las señales y la liberación de recursos. Es por esto que resulta poco escalable y poco eficiente, pues una conexión ocupa la totalidad del recurso óptico de la red, evitando que información diferente sea transmitida [11]. Sin embargo, dado que es la tecnología de red más madura, es la más implementada gracias a los grandes avances en la electrónica, por lo que muchos de los operadores de red hacen importantes inversiones en el despliegue y mejora de estas redes, lo que ha convertido a OCS en una tecnología de conmutación clave para soportar las redes ópticas [12]. La figura 1.4 muestra un nodo OXC.

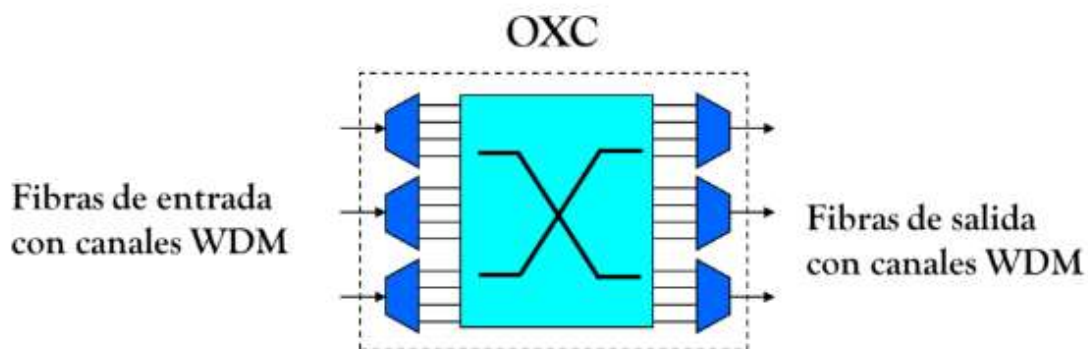


Figura 1.4 - Nodo OXC [13]

- Conmutación óptica de ráfagas

Optical burst switching o OBS, realiza un proceso denominado “ensamblaje de ráfagas”, que consiste en agrupar paquetes de información en el nodo de ingreso, hasta formar una ráfaga de información compuesta por datos de diferentes servicios o usuarios[14]. El tamaño de la ráfaga se limita por la cantidad de tiempo de conformación, la longitud de esta, o por una mezcla de estos dos parámetros, en busca de la mayor eficiencia [13]. Las ráfagas son precedidas por una cabecera óptica o BCP (Burst control packet), la cual es enviada por una longitud de onda diferente, encargada de configurar los nodos intermedios para formar la ruta por la cual será enviada la carga útil, la cual es despachada después de un tiempo de “offset” desde que es enviado el BCP, manteniendo así la información de usuario en el dominio óptico. En el destino se realiza el desensamblaje de las ráfagas y reenvío de paquetes a sus respectivos destinos, separando así los

paquetes individuales. Esto hace de OBS una técnica veloz y eficiente [15]. En la figura 1.5 se observa el diagrama de bloques de OBS.

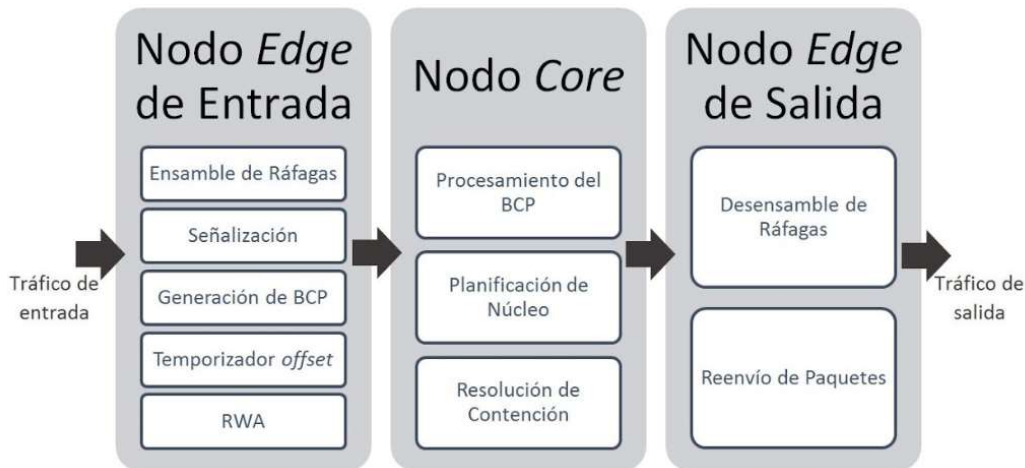


Figura 1.5 - Esquema funcional de una red OBS [13]

Una red óptica OBS consiste de nodos OBS interconectados mediante enlaces WDM y usuarios conectados a los nodos borde mediante enrutadores usualmente de tipo "IP (Internet Protocol)" con una interfaz OBS. Por otra parte, los nodos núcleo o "core" se encargan de la planeación de recursos y resolución de contenciones, esto último sí los recursos disponibles no satisfacen los requerimientos de planeación al tiempo que arriban las ráfagas de datos. [16]

OBS hace uso de la técnica RWA (routing and wavelength assignment) para solucionar el problema de "routing" que consiste en establecer una ruta adecuada entre el nodo origen y destino. RWA establece la ruta a seguir para los paquetes mediante el uso de tablas de enrutamiento y al mismo tiempo hace gestión para la distribución de las longitudes de onda a utilizar para el envío de la información [17].

El nodo Edge o frontera o borde se encarga de la recepción de paquetes procedentes de otras redes como por ejemplo Sonet, SDH, entre otras, luego hace un análisis de sus cabeceras con el fin de determinar a qué nodo borde de la red deben ser enviados. Una vez hecho esto, se encarga del ensamblaje de ráfagas y demás procesos mostrados en la figura 1.5, estos serán abordados a continuación.

- **Ensamblaje de ráfagas:** En este proceso se conforman las ráfagas con los paquetes de datos procedentes de niveles superiores como IP. El ensamblaje es realizado basado en criterios como destino de los paquetes,

longitud, tipo de tráfico, entre otros. Un algoritmo de ensamblaje determina las políticas de diseño de ráfagas, basado en la duración de la misma o la longitud en bytes o una mezcla de estos [13].

- **Señalización:** Otra de las funciones del nodo frontera es el establecimiento de la comunicación en una red OBS entre dos nodos frontera a través de nodos centrales, proceso en el cual se determina el funcionamiento de la red y se asignan los recursos de red requeridos para la transmisión de la ráfaga, este proceso se lleva a cabo utilizando un canal dedicado (diferente de los canales de datos). En OBS es posible emplear dos tipos de señalización, la distribuida y centralizada. En la señalización distribuida, la reserva de los recursos a usar se realiza nodo a nodo; en la señalización centralizada, un elemento central con una visión de toda la red el encargado de informar a todos los nodos involucrados la reserva de los recursos [16].
- **Generación del BCP:** El paquete de control de ráfaga o BCP es el encargado de transportar la información sobre la longitud de la ráfaga respectiva, tiempo de offset, calidad del servicio (solo si existe), identificación de la longitud de onda, identificación de puerto de entrada y demás. Este paquete es transmitido sobre una longitud de onda dedicada para el control de la red [13].
- **Temporizador de Offset:** El tiempo de offset consiste en la suma de tiempos de procesamiento, establecimiento de rutas en los nodos intermedios, tiempos de conmutación y tiempos de viaje de la información a través de las fibras que conforman el camino óptico establecido [16], El cálculo de tiempo de offset es de suma importancia pues después de cumplido este tiempo es cuando la ráfaga de datos será enviada. El tiempo offset es añadido a la cabecera BCP, garantizando así que la ráfaga pase a través del nodo de forma transparente [18].
- **RWA:** En las redes ópticas se presenta el problema de encaminamiento y asignación de longitudes de onda (Routing and Wavelength Assignment) que consiste en la necesidad de encontrar una ruta óptima para el envío de información al mismo tiempo que elige cuál de las longitudes de onda disponibles en el sistema WDM implementado es la más adecuada. Algunos algoritmos buscan resolver ambas tareas de manera simultánea, sin embargo, esto conlleva a una mayor complejidad computacional. Otra forma de resolver los dos problemas es aplicando un método secuencial (R + WA), reduciendo así la complejidad computacional [13].

El nodo core o núcleo, es el encargado de la conmutación de la ráfaga en el dominio óptico y se ubican en el centro de la red OBS, cumpliendo con las siguientes labores:

- **Procesamiento del BCP:** Aquí se analiza el paquete óptico BCP, el cual es llevado al dominio eléctrico para establecer así la configuración del nodo para que cuando llegue la ráfaga, esta pueda pasar a través de él manteniéndose en el dominio óptico [16].
- **Planificación de núcleo:** El análisis de los BCP permite a cada nodo core planificar los recursos de los conmutadores de los que hará uso, de tal manera que las ráfagas puedan pasar a través de ellos. Existen diferentes esquemas de planificación de recursos en redes OBS y se clasifican de acuerdo a tiempos de comienzo y fin de la ocupación de los recursos por parte de las ráfagas [13].
- **Resolución de la contención:** La contención ocurre cuando una ráfaga llega a un nodo OBS y encuentra que todos los recursos están ocupados o cuando dos ráfagas necesitan el mismo recurso. Para la solución de este problema existen varias técnicas que se listan en la tabla 1.

Tabla 1.1 - Comparación de los esquemas de contención [13]

Esquema	Ventajas	Desventajas
Conversión de longitudes de onda	Baja pérdida de ráfagas	Tecnología costosa
Almacenamiento en FDLs	Tecnología simple y madura	Causa pérdida de potencia y produce ruido
Deflexión de ruta	No requiere Hardware adicional	Información fuera de orden e inestable
Segmentación de ráfaga	Alta eficiencia y baja pérdida de ráfagas	Tecnología inmadura, alta complejidad, retardo y señalización extra

Finalmente, el nodo borde de salida se encarga de la conversión O/E de la ráfaga recibida, finalizando así el trabajo de la red OBS. Las funciones que cumple son las siguientes:

- **Desensamble de ráfagas:** Aquí se recuperan los paquetes individuales que conforman la ráfaga, llevando así la información a su estado inicial y deshaciendo la ráfaga.
- **Reenvío de paquetes:** Los paquetes individuales de datos son enviados a los destinos que les corresponden.

- **Conmutación óptica de paquetes**

Optical packets switching o OPS, es la técnica más compleja de conmutación de datos en las redes ópticas [10], En esta técnica, la información se agrupa en pequeños paquetes de datos y viaja junto a una cabecera de control o header, pasando a través de cada uno de los nodos intermedios de la ruta, dichos nodos deben de procesar la información de la cabecera llevándola al dominio eléctrico (OPS transparente), para configurar el enrutador, el cual conmuta la carga útil de forma óptica, haciendo de OPS la técnica que permite mayor velocidad de datos debido a la poca complejidad de sus cabeceras y a que evita la conversión O/E/O de la carga útil. Si la cabecera es analizada en el dominio óptico, es decir sin necesidad de realizar conversión óptico-eléctrica, esta versión es llamada “conmutación de paquetes totalmente ópticos”, siendo esta la forma más compleja y más prometedora de esta tecnología aún inmadura, aunque actualmente se realizan investigaciones que sugieren el uso de correladores ópticos para evitar la conversión O/E/O de la cabecera. [16], [13], [19].

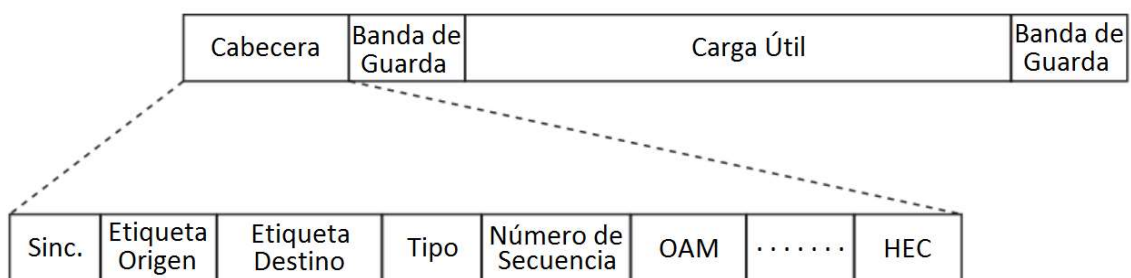


Figura 1.6 - Estructura de un paquete OPS [20]

OPS exige una altísima velocidad de procesamiento de cabeceras, situando estos procesos en el orden de unos pocos nanosegundos, también permite la reescritura de sus cabeceras en cada nodo intermedio, actualizando de ser necesario los parámetros que se encargan del transporte de la información de usuario a lo largo de la red. En la figura 1.6 se observa la composición de un paquete OPS y a continuación una breve descripción de su cabecera [20].

- **Sinc:** bits de delineamiento y sincronización.
- **Etiqueta origen:** dirección del nodo fuente.
- **Etiqueta destino:** dirección del nodo destino.
- **Tipo:** tipo y prioridad del paquete y carga útil transportada.
- **Número de secuencia:** número de secuencia del paquete, lo que permite la entrega ordenada de los mismos.
- **OAM:** funciones de Operación, Administración y Mantenimiento.
- **HEC (Header Error Correction):** corrección de errores en el encabezado.

OPS permite la implementación de una codificación más simple en sus cabeceras de control, para lograr una mayor velocidad de procesamiento de las mismas. En el nodo, la cabecera debe llevarse al dominio eléctrico, mientras que la carga útil permanece en espera en el dominio óptico, para ello se propone el uso de fibras de retardo o FDL siendo esto lo más cercano a una memoria óptica.

Según la figura 1.7, un nodo OPS consta de los siguientes elementos:

- **Interfaz de entrada**, esta se encarga de los siguientes procesos [13], [20]:
 - Delineamiento de los paquetes PD (Packet Delination): Identifica el principio y final de los paquetes y la carga de usuario.
 - Conversión de longitud de onda WC (Wavelength Conversion): Si es necesario convierte una longitud de onda externa en una interna.
 - Sincronización (Synchronization): Solo para conmutadores síncronos, alinea los diferentes paquetes que llegan desde distintas longitudes de onda y/o puertos.
 - Procesamiento del Encabezamiento HP (Header Processing): Conversión O/E del encabezado y su reenvío hacia la unidad de control del nodo.
- **Unidad de control**, encargada de:
 - Procesar el encabezado y fija la configuración del conmutador.
 - Actualizar la cabecera y la reenvía a la interfaz de salida.

- **Matriz de conmutación**, se encarga de:
 - Conmutación de la carga útil según la unidad de control.
 - Resolución de la contención.
- **Interfaz de salida**, se encarga de:
 - Reamplificar la señal óptica, reorganizar los datos y resincronizar los paquetes (proceso conocido como las 3R; Reamplification, Reshaping, Retiming).
 - Adicionar la cabecera actualizada.
 - Delinear el paquete.
 - Convertir de una longitud de onda interna a una externa de ser necesario.
 - Resincronización (Solo en conmutadores ópticos síncronos).
- **Buffer óptico**, este se encarga de almacenar la información de usuario manteniéndola en el dominio óptico, haciendo uso de las ya mencionadas FDL. Si bien su uso es sugerido, existen esquemas OPS que no hacen uso de estos dispositivos, sino que emplean otras técnicas para la solución de las contiendas [21], [22].
- **Convertidores de longitud de onda**, encargados de la conversión de una longitud de onda a otra solo de ser necesario. Algunos sistemas poseen un convertor para cada puerto de entrada o salida, mientras que otros comparten un convertor con varios o todos los puertos de los que disponen [15].

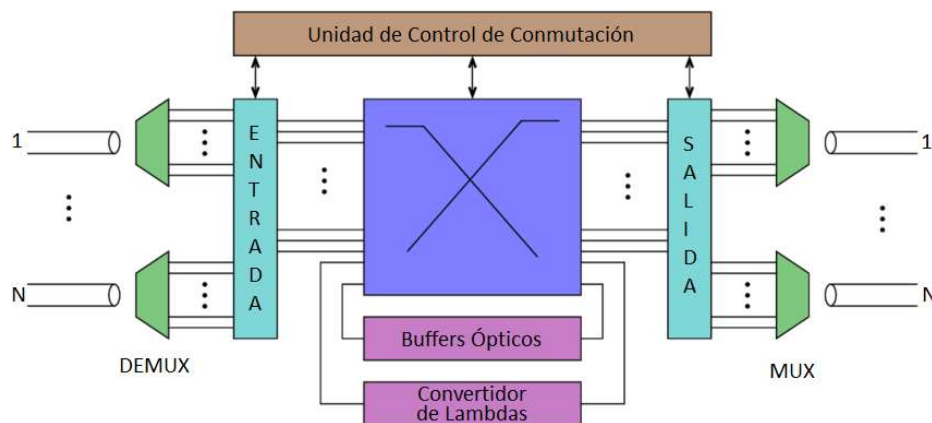


Figura 1.7 - Nodo OPS [21]

En OPS es necesario el planteamiento de esquemas para hacer frente a la contienda por los recursos, para ello se plantean tres métodos que pueden implementarse individualmente o de forma combinada [22], estos son descritos brevemente a continuación:

- **Uso de FDLs:** El uso de Fiber Delay Lines o FDL, permite almacenar la información manteniéndola en el dominio óptico mientras la contienda es solucionada, es decir mientras se liberan los recursos necesarios para continuar con el envío de la información. Si bien este método ofrece la menor pérdida de paquetes, puede resultar poco eficiente debido a que el tiempo que un paquete se mantiene en la FDL viene dado por la longitud de la misma, haciendo de este un valor fijo, por lo que cuando el paquete emerja del otro lado de la fibra de retardo puede que la contienda se haya solucionado tiempo antes o que aún continúe, es por esto que se sugieren arreglos de FDLs, buscando permitir mayor variedad en los tiempos de almacenamiento [16], [21].
- **Conversión de longitud de onda:** Su uso es menos recomendado que las FDL, pues con esta técnica lo que se busca es llevar los paquetes de una longitud de onda ocupada a otro disponible en el puerto de salida, haciendo necesaria una conversión de la información y quitando transparencia al proceso, lo que agrega retardos a la red [21].
- **Deflexión de ruta:** Este método explota el dominio espacial, puesto que aquí se busca cambiar el puerto original de salida del paquete en contienda por un puerto libre, lo que involucra un cambio de ruta de la información, lo que puede llevar a retardos adicionales y a un arribo en desorden de los paquetes en el nodo destino [21], por lo que es el método menos recomendado de los tres.

Tabla 1.2 - Comparación entre las técnicas de conmutación [10]

Técnica	Granularidad	Uso del ancho de banda	Latencia	Buffer óptico	Tolerancia a fallos	Complejidad
OCS	Amplia	Pobre	Alta	No requiere	Baja	Baja
OPS	Fina	Alta	Baja	Requiere	Alta	Alta (En desarrollo)
OBS	Moderada	Moderada	Baja	No requiere	Media	Moderada

Para hacer frente al problema de enrutamiento y gestión longitudes de onda, en OPS se propone la utilización de RSA (Routing and Spectrum Allocation), que consiste básicamente en realizar una asignación dinámica del espectro disponible en el sistema y no de las longitudes de onda como RWA. Dado el interés en esta técnica, será estudiada con mayor detalle más adelante.

En la tabla 1.2, se puede observar la comparativa entre las técnicas citadas. Para este trabajo de grado, se van a tomar algunos elementos de OBS y OPS con el fin de aprovechar los beneficios de una y otra tecnología.

1.1.3 Redes ópticas elásticas - EON (Elastic Optical Networks)

Sí bien WDM permitió maximizar el uso del espectro óptico, no permite el uso más eficiente del mismo, pues, un sistema WDM ocupa todos los recursos de los que dispone aunque la cantidad de tráfico sea baja, desperdiciando así ancho de banda y potencia, encareciendo su implementación [23]. Han sido propuestos sistemas WDM con diferentes esquemas de modulación, permitiendo el transporte de información a diferentes tasas de bits, brindando así cierta elasticidad [12], pero sin ser esta una solución definitiva al problema, lo que ha llevado a una amplia investigación en este campo.

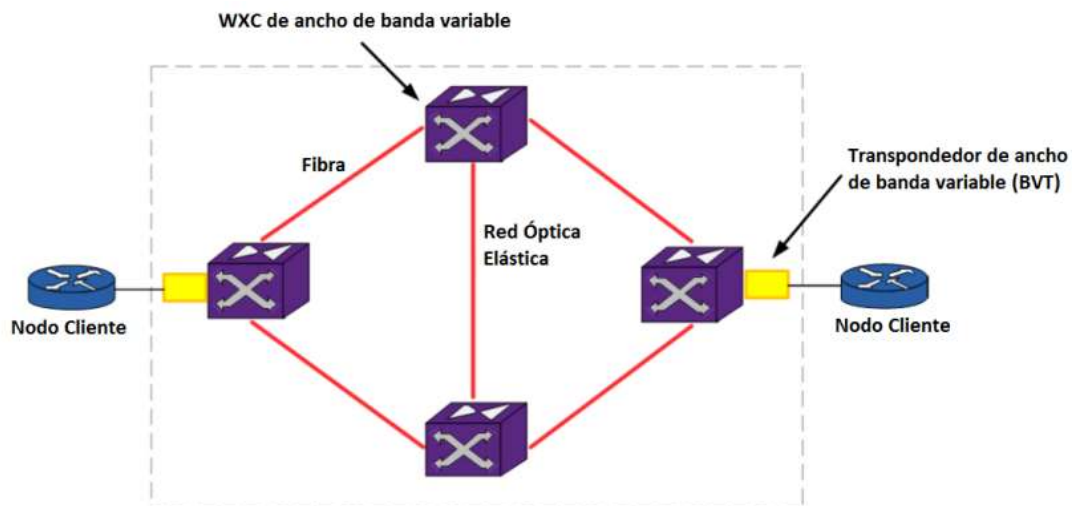


Figura 1.8 - Arquitectura de una EON [23]

Para mejorar la eficiencia y uso de recursos, las redes ópticas elásticas también conocidas como EON (Elastic Optical Networks) o Flexgrid, se perfilan como la tecnología para soportar el creciente flujo de información en las redes actuales, pues se espera que para el año 2021, la media de velocidad de conexión a internet alcance los 53 Mbps, valor superior a la tasa medida en el 2016 de 27,5

Mbps, se espera también que el tráfico de información per cápita aumente un 24% cada año hasta el año 2021, donde las expectativas están en una media de 35 Gb de datos al mes por usuario, frente a los 13 Gb del año 2016 [24], es por esto, que se necesita de una tecnología de red capaz de soportar el creciente tráfico de información, maximizando el uso de los recursos de red disponibles al mismo tiempo que aprovecha la infraestructura actual. La arquitectura de una red EON es mostrada en la figura 1.8.

Una red óptica elástica, se define como una red de transporte óptico (OTN, Optical Transport Network), que tiene la capacidad de adaptar el espectro asignado de las conexiones ópticas a los requerimientos de ancho de banda del tráfico transportado. Mientras que en las redes ópticas rígidas (como las actualmente utilizadas, redes WDM), cada conexión óptica ocupa la misma cantidad de espectro durante el tiempo de duración de la conexión [25]. Las EON También permiten cierto grado de flexibilidad sobre algunos otros parámetros de red como la distribución de longitudes de onda, ancho de banda, formatos de modulación, tasas de bit, entre otros, a medida que varían las condiciones de tráfico, la cantidad de usuarios y demás factores que afectan el comportamiento de la red, procurando siempre mantener un alto grado en la calidad de la transmisión o QoT [26]. En la figura 1.9 se observan las áreas de investigación de las redes elásticas.

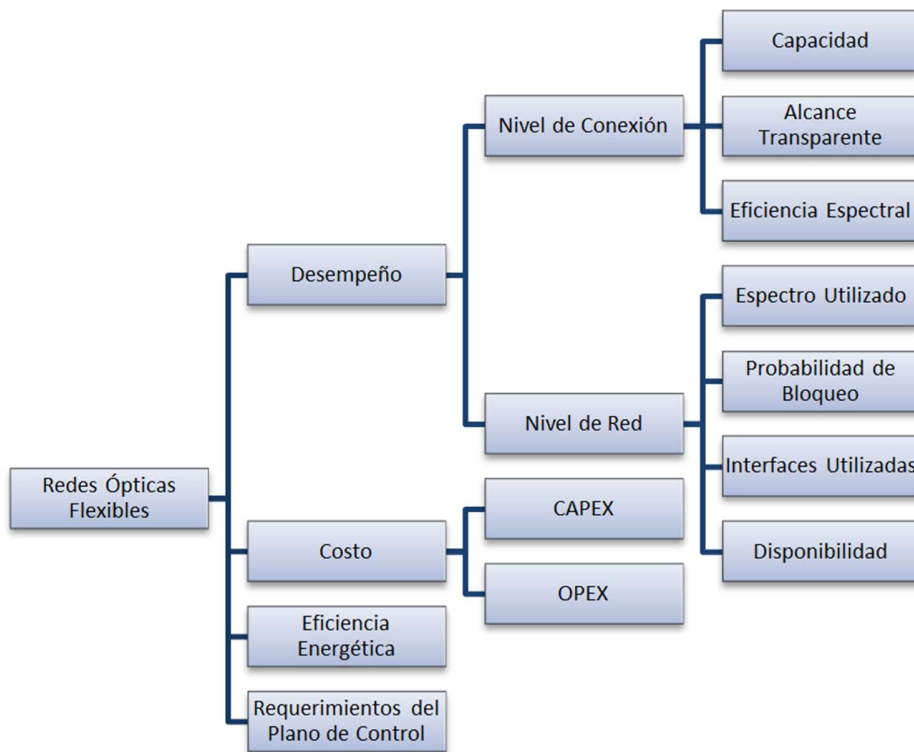


Figura 1.9 - Áreas de investigación de Flexgrid [26]

Como primer paso para lograr obtener una red EON es necesaria la sustitución de la rejilla de espectro fija, que cuenta con una separación de canales de 100 GHz, 50 GHz y 25 GHz (Figura 1.10), por una rejilla “flexible” establecida por la ITU-T en el estándar G.694.1, en la cual el espectro óptico de la ventana C (1530-1565 nm) es dividido en pequeños ranuras de frecuencia de 6.25 GHz cada una, dos segmentos conforman un slot de 12.5 GHz al que se le asigna una frecuencia central o CF, (Central Frequency), que debe coincidir con el principio o el final de estos segmentos [27]. Por lo que la CF tomando como referencia la frecuencia de 193.1 THz está dada por la ecuación 1.1.

$$CF = 193.1 [THz] + n * 0.00625 [THz] \quad \text{Ecuación (1.1)}$$

Donde n es entero positivo o negativo (incluido el 0), el cual tiene un valor mínimo de -246 (índice de la frecuencia correspondiente a 1565 nm) y máximo de 455 (índice de la frecuencia correspondiente a 1530 nm), que representa el índice de la frecuencia central de cada segmento [28], [29].

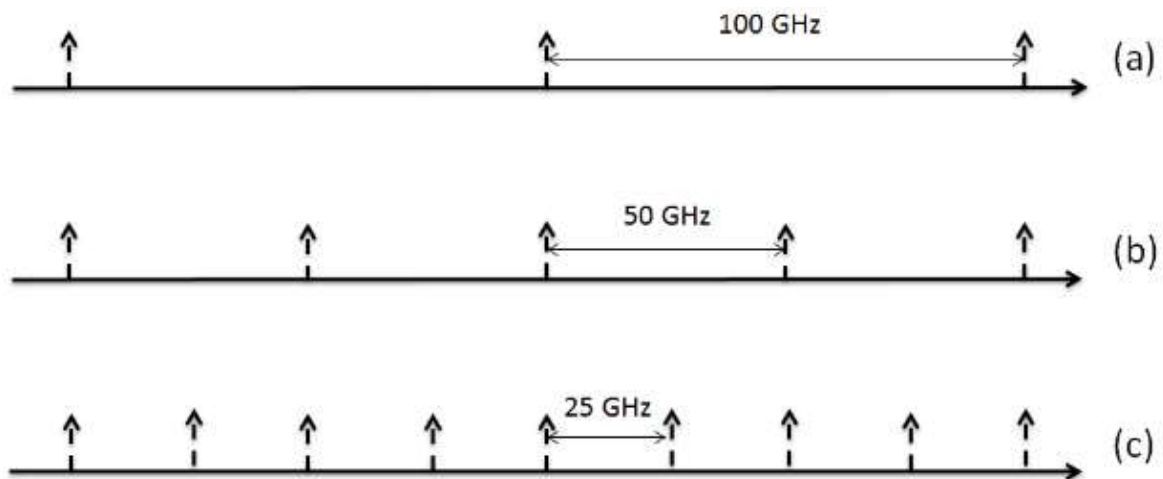


Figura 1.10 - Rejilla WDM (a), canales separados 100 GHz, (b), canales separados 50 GHz, (c), canales separados 25 GHz [5], [28]

Debe entenderse también el concepto de ancho de ranura o SW (Slot Width), que es igual al número de segmentos o intervalos de frecuencia (FSs, Frequency Slots) asignados a la conexión, multiplicado por 12.5 (tamaño de un slot en GHz). Por lo anterior, el ancho de ranura se puede expresar mediante la ecuación 1.2, donde m es un número entero positivo, y representa el número de slots de frecuencia necesarios para cumplir con el ancho de banda requerido.

$$SW = 12.5 [GHz] * m \quad \text{Ecuación (1.2)}$$

Por lo anterior, un canal puede definirse por su frecuencia central y su respectivo ancho de banda (diferencia entre su frecuencia final e inicial), como lo muestra la ecuación 1.3, en f_i es la frecuencia inicial del canal en THz. En la figura 1.11 se observa un ejemplo del concepto de ancho de ranura para facilitar su comprensión.

$$FC = f_i + 0.00625 * m \quad \text{Ecuación (1.3)}$$

Se puede observar de la figura 1.11, que el ancho de banda disponible se divide en slots de 12.5 GHz. Por ejemplo, para lograr conseguir el ancho de banda de 37.5 GHz (conexión 1) y 50 GHz (Conexión 2) se requieren 3 y 4 slots de frecuencia respectivamente. Por lo anterior, con los conceptos de (frecuencia central y ancho de ranura), es posible definir un canal en función de m y n [28].

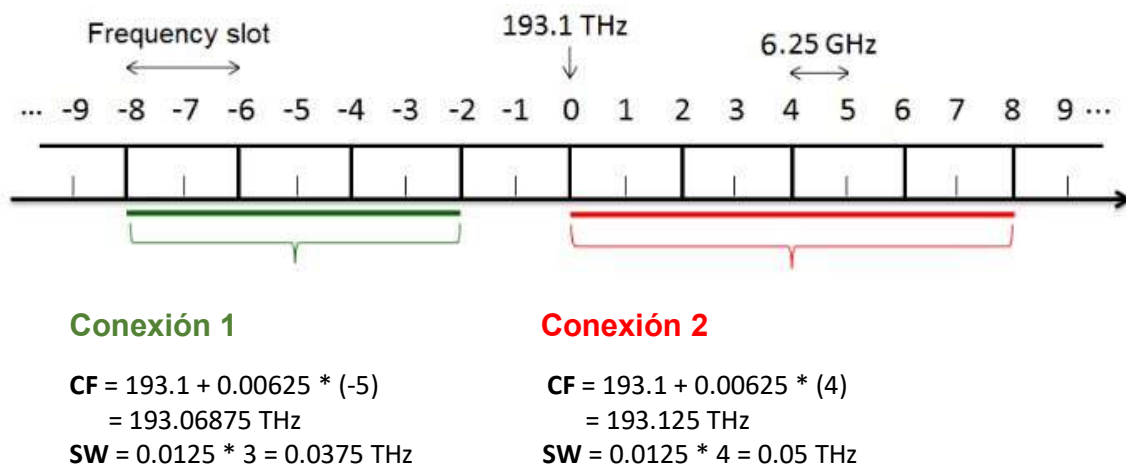


Figura 1.11 - Ejemplo de concepto de ancho de ranura [28]

Como se observa de la figura 1.9, las redes elásticas tienen diferentes campos de investigación, por lo que en este trabajo de grado se desarrollará un algoritmo situado en el nivel de red, específicamente para la gestión del espectro utilizado, haciendo de este el parámetro que otorgue la flexibilidad a la red, por lo que será el tema principal de estudio.

Dado que las EON buscan la transmisión de información tanto a bajas velocidades (10, 40 Gbps) como a altas tasas en el futuro (100, 400 y hasta 1000 Gbps), se hace necesario el estudio de diferentes técnicas para resolver el problema de gestión de espectro y permitir tasas de datos variables. Las EON generalmente hacen uso de diferentes técnicas, las más comunes son OFDM (Orthogonal Frequency-Division Multiple Access) y Nyquist WDM de las que se hablarán brevemente a continuación.

- **OFDM**

El acceso múltiple por división de frecuencias ortogonales o OFDM permite convertir un flujo de datos de alta velocidad en varios flujos de datos de menor tasa de bits, para ello divide una portadora de gran capacidad en varias subportadoras de menor capacidad, logrando así asignar a una conexión la cantidad de subportadoras necesarias para satisfacer la demanda de ancho de banda de la conexión [30, 31].

En la figura 1.12 se puede observar la característica clave de OFDM, la ortogonalidad de sus portadoras. Esta propiedad permite que las señales se solapen unas con otras, sin causar interferencia entre sí, pues la máxima potencia de cada señal subportadora se encuentra en los mínimos de potencia de las demás señales, permitiendo así acomodar muchas más señales en una menor cantidad del espectro al eliminar la necesidad de bandas de guarda entre señales [4], [23].

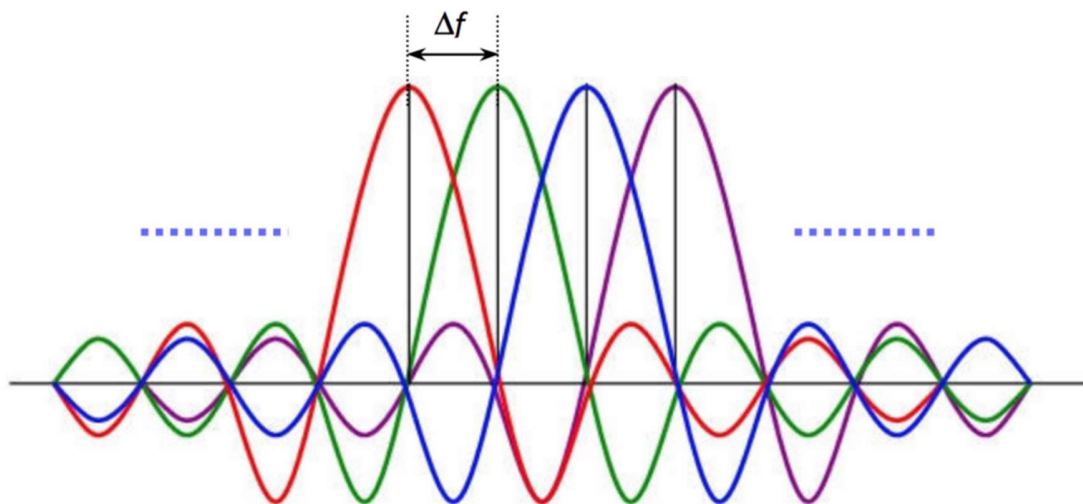


Figura 1.12 - Señal OFDM en el dominio de la frecuencia [32]

- **Nyquist WDM**

Nyquist WDM (N-WDM), permite conseguir una alta eficiencia espectral. Esta técnica consiste en la concatenación de varios canales WDM; La figura 1.13 muestra una representación de la técnica en cuestión.

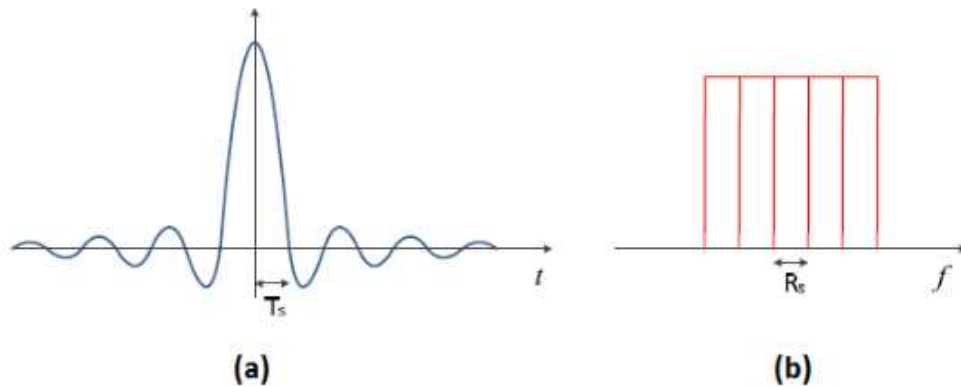


Figura 1.13 - (a) Respuesta en frecuencia, (b) Respuesta en el tiempo, de una señal N-WDM [27]

Aunque teóricamente esta técnica permite alcanzar la máxima eficiencia espectral (1 símbolo/Hz) es poco implementada debido a la necesidad de generar una señal rectangular en el dominio de la frecuencia, haciendo que las investigaciones se enfoquen mayormente en OFDM [27].

1.1.4 Routing and Spectrum Allocation

RSA es una de las técnicas más prometedoras para la asignación de espectro, dentro de las redes elásticas. Con esta técnica se busca resolver el problema de enrutamiento y asignación de espectro. Para ello se establece un camino de luz uniforme desde el nodo origen hasta el nodo destino y, que además contenga el ancho de banda demandado por el usuario para soportar el envío de información [33, 34]. El objetivo de RSA es entonces, encontrar una ruta con suficiente espectro libre para albergar el ancho de banda requerido por las demandas de tráfico. Es decir, que este problema se puede abordar de dos maneras, por un lado, el enrutamiento o establecimiento del camino óptico y por otro, la asignación de los recursos espectrales a la conexión óptica. La figura 1.14 muestra la distribución de espectro para RSA.

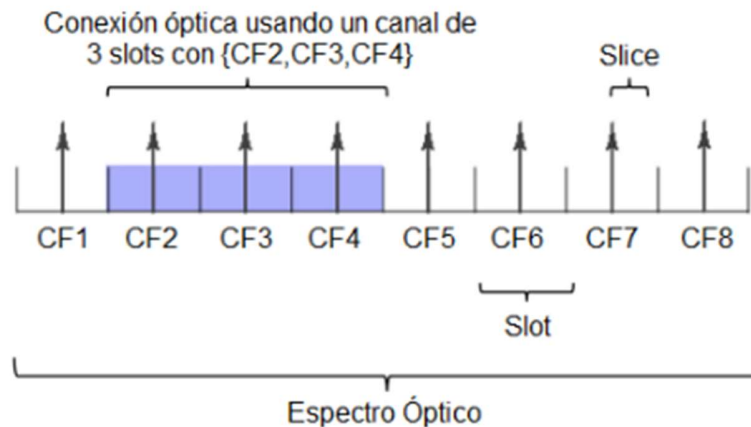


Figura 1.14 - Rejilla de espectro óptico en RSA [35]

RSA opera mediante la asignación de slots (unión de dos slices o porciones consecutivas de espectro de 6.25 GHz), definidos por su frecuencia central (CF). La CF indica el lugar donde se centra el espectro asignado a la conexión, por lo que el canal óptico RSA estará definido por la CF y la cantidad de slots asignados a la conexión. RSA debe considerar las siguientes condiciones de operación:

- **Contigüidad del espectro:** todos los slots de frecuencia que conforman el canal en una conexión óptica, deben ser adyacentes en el espectro.
- **Continuidad del espectro:** el canal asignado a una conexión óptica, debe ser el mismo para todo el camino óptico desde el nodo origen hasta el nodo destino (es decir, utilizando la misma frecuencia central).
- **No superposición del Espectro:** un slot de frecuencia en un enlace sólo se puede utilizar para una conexión a la vez.
- **Espectro limitado:** hay un número finito de slots en cada enlace.
- **Bandas de guarda:** como en todos los filtros pasa-banda reales, las funciones de transferencia no son perfectamente rectangulares. Esto hace que sea necesario separar las conexiones espectralmente adyacentes en cada fibra, por al menos un slot no asignado para evitar interferencias.

Existen varias maneras de realizar la asignación de espectro en RSA, a continuación, se hará mención de ellas.

- **Asignación de espectro fija en el tiempo:** En esta técnica, la cantidad de espectro fijada a un canal permanece fija durante todo el tiempo de

conexión, lo que la convierte en la técnica con menos elasticidad dentro de RSA, pues aquí, tanto la CF como los slots serán estáticos hasta que se liberen los recursos. [35].

- **Asignación de espectro semi-elástico en el tiempo:** Aquí, la CF se mantiene fija, pero el ancho espectral o cantidad de slots puede variar en el tiempo, ajustándose a los requerimientos de tráfico, bien sea asignando más slots (siempre que se cumplan las condiciones de continuidad y contigüidad) disponibles si es que se requiere más ancho de banda o reduciendo la cantidad de slots asignados si la conexión necesita menos recursos [33], [35], la figura 1.15 muestra una representación de esta técnica.

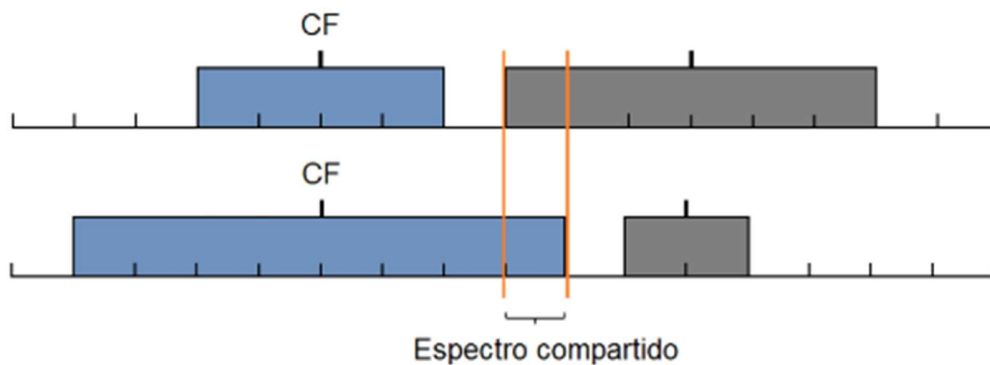


Figura 1.15 - RSA semi-elástico [35]

- **Asignación de espectro elástico en el tiempo:** En esta técnica, tanto la CF como el ancho espectral asignado, pueden variar en cada periodo de tiempo, como se observa en la Figura 1.16. De esta manera, los canales compartirán espectro, pero no se asignará el mismo slot a dos canales distintos en el mismo periodo de tiempo. Puede haber dos variantes de este esquema: ESA con Expansión/Contracción de espectro y ESA con reasignación de espectro.
 - a. **Expansión/Contracción:** en este caso, el espectro compartido entre dos peticiones será como máximo, la anchura de espectro asignado al más pequeño de los dos canales. El movimiento de la CF está limitado al rango de frecuencias de la petición más ancha, este caso se puede observar en la figura 1.16

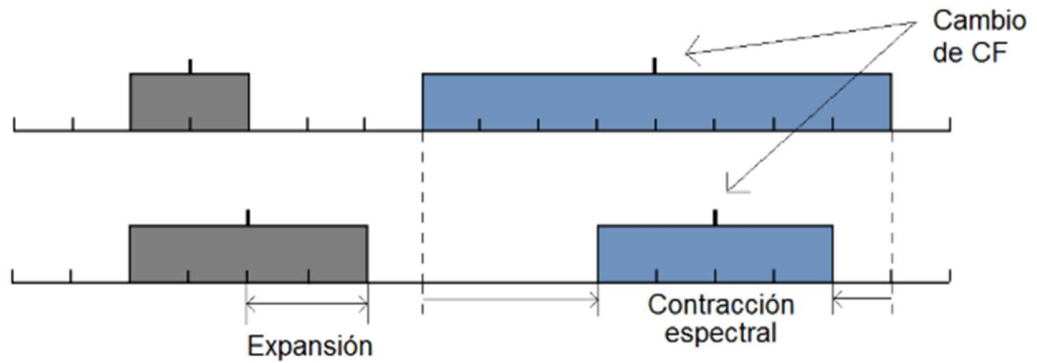


Figura 1.16 - Expansión/Contracción en RSA [35]

- b. Reasignación de espectro: cualquier parte del espectro puede ser recolocado en otro espacio. No hay limitación para reubicar la CF. La figura 1.17 representa este caso.

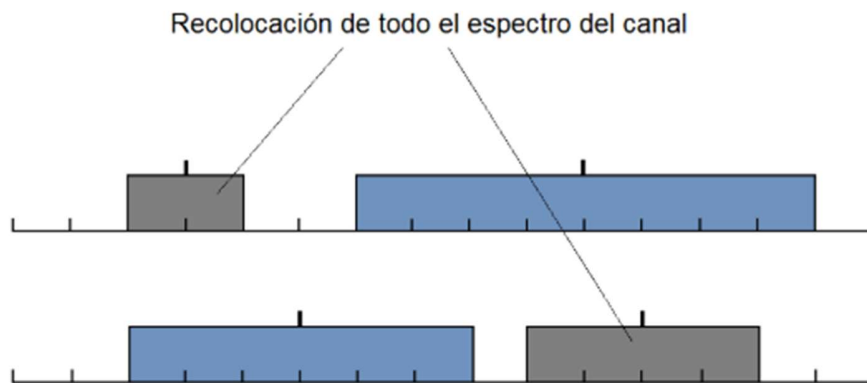


Figura 1.17 - Reasignación de espectro en RSA [35]

1.2 SDN

1.2.1 Redes SDN

Las redes de datos han tenido grandes avances desde su invención y primeras versiones capaces de transportar unos pocos bits por segundo, hasta llegar a las redes que se conocen hoy, capaces de transmitir cientos de gigabits de datos en cuestión de un segundo [36]. Al mismo tiempo que aumentaba la capacidad de las redes para comunicarse, también aumentó la complejidad en su operación, pues nuevos dispositivos, técnicas, protocolos, entre otros se hicieron necesarios para romper nuevas barreras, así pues, la inclusión de dispositivos de red capaces de procesar información para su respectiva gestión se hizo necesaria, por lo que

equipos como enrutadores, conmutadores, entre otros, debieron ser dotados de la electrónica capaz de procesar datos y tomar decisiones en base a los criterios de operación de la red, lo que si bien permitió el establecimiento de las comunicaciones modernas, encareció también la fabricación y puesta en operación de dichos dispositivos por lo que el costo de instalación de la red (CAPEX) y mantenimiento (OPEX) se elevó junto con ellos.

Es por lo anterior que el proceso de centralización de las redes se perfila como una técnica capaz de disminuir costos y facilitar la operación de las redes al ubicar en un nodo central todos los procesos de decisión que antes eran tomados por cada uno de los equipos de red. Las redes definidas por software o SDN (Software Defined Networking), buscan entonces redefinir la arquitectura de la red, logrando una gestión centralizada de la red y de sus procesos claves [12], [37].

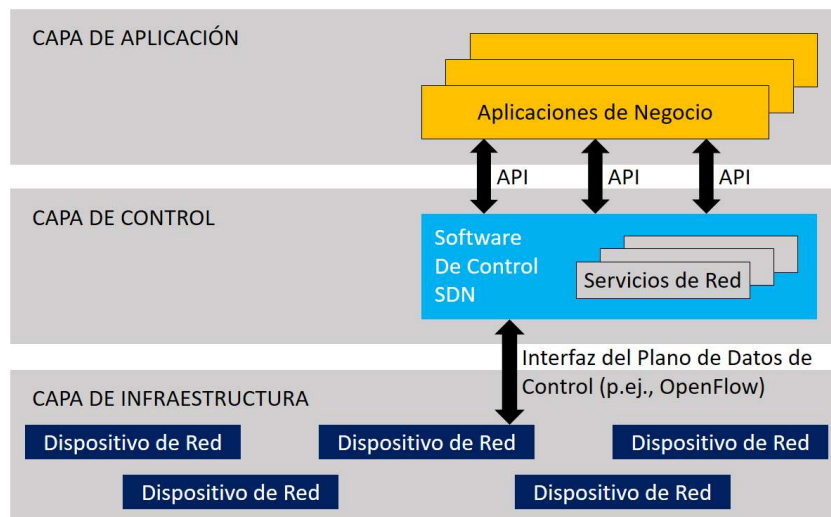


Figura 1.18 - Arquitectura SDN [12]

El paradigma SDN propone una nueva arquitectura de red, separando el plano de control (encargado de manejar el tráfico de la red y decidir cómo se gestionarán los recursos de red) del plano de datos (encargado de reenviar el tráfico según las decisiones del plano de control) que están integrados en los dispositivos de red, lo que disminuye la flexibilidad de la red y limita la evolución de la misma. Al separar los planos de control y datos, el control de la red se hace entonces en un sistema operativo de red ubicado en un nodo central, cabe resaltar que la centralización lógica de la red no significa una centralización a nivel físico de la misma, pues cuestiones como escalabilidad, desempeño y confiabilidad de la red lo impiden [38]. En la figura 1.14, se observa la arquitectura de SDN.

En la figura 1.18, se puede ver la separación vertical entre los planos de datos y control y la capa de aplicación, la gestión de la comunicación entre estas capas está a cargo de un protocolo, siendo OpenFlow el más reconocido y utilizado debido a su alta capacidad de gestión de la información. A continuación, se explicarán con más detalle las diferentes capas que componen la arquitectura SDN.

1.2.2 Capa de aplicación

La capa de aplicación contiene programas encargados de enviar las peticiones de comportamiento y funcionamiento de la red a los controladores de las capas inferiores mediante NBIs (Northbound Interface) que son básicamente APIs (application programming interface) diseñadas para permitir la comunicación “hacia abajo” en el diagrama de infraestructura de SDN. La capa de aplicación está conformada por uno o varias lógicas de aplicación y uno o más NBIs. Es gracias a esta capa que se logra tener una visión holística de toda la red, verificar y controlar su operación.

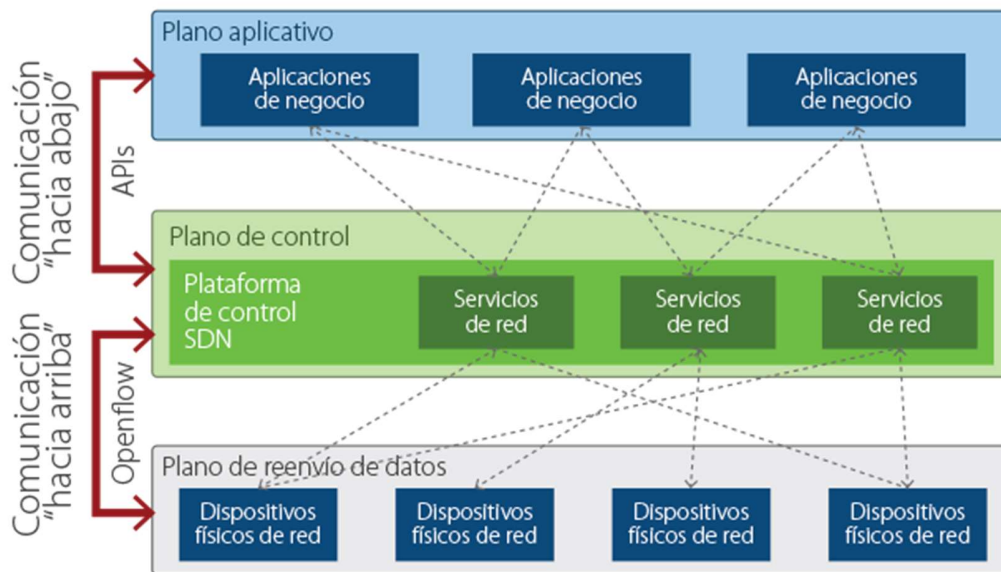


Figura 1.19 - Planos de SDN [39]

Existen diferentes tipos de aplicaciones en esta capa, algunas se enfocan en parámetros de la red como, por ejemplo: enrutamiento de la información, itinerancia de la red, mantenimiento de la red y su visión, seguridad, cloud computing o computación en la nube y virtualización, entre otras [40]. En la figura 1.19, se puede apreciar la división de los planos o capas de SDN y su forma de interacción.

1.2.3 Capa de control

La capa de control es la parte más importante de la arquitectura SDN [40], pues es esta la encargada de la gestión integral de la red. Se encarga de configurar y actualizar los dispositivos de la capa de datos a partir de las peticiones hechas por la capa de aplicación [36, 37]. Esta capa se encarga también de proporcionar una visión global u holística de la red.

Los dispositivos de la capa de datos son los encargados de recolectar y comunicar la información del estado de la red al controlador mediante una interfaz como por ejemplo “openflow”, después, el controlador SDN se encarga de tomar las decisiones sobre la operación de los dispositivos de red, basándose en las reglas definidas en la capa de aplicación y comunicándolas de nuevo mediante la interfaz hacia los dispositivos de la capa de datos [41].

En una red se suele disponer de uno o más controladores para evitar cualquier posible falla debido a congestión o a “bugs” del software, por lo que se dispone de controladores en backup, pero estos no son de uso obligatorio, siendo más un criterio a definir por los administradores de la red. Los controladores de red no están estandarizados, por lo que estos deben ser capaces de manejar un lenguaje común para intercambiar información con los controladores de otras redes.

1.2.4 Capa de datos

En esta capa se pueden encontrar lo que anteriormente eran los dispositivos de red con inteligencia embebida como conmutadores o enrutadores, siendo ahora equipos con la única misión de recibir y enviar el tráfico sin tomar decisiones en el proceso.

La capa de datos, también llamada de infraestructura, es la encargada del reenvío de información a través de la red. Esta capa opera según las instrucciones recibidas desde la capa de control. Se encarga también de recolectar los datos necesarios para que el controlador SDN tome las decisiones correspondientes para operar la red. Entonces, se puede decir que en esta capa se ejecutan dos tareas principales [40], [42]:

- **Control:** Se recoge la información acerca de la operación de la red, para luego ser enviada al controlador [40].
- **Datos:** Los paquetes de datos son enviados a sus respectivos destinos asociando las decisiones del controlador SDN con las cabeceras de cada

paquete de información. En SDN además de poder dirigir paquetes por su respectiva identificación IP o MAC (Media Access Control), también es posible realizar direccionamiento haciendo uso de etiquetas VLAN o puertos TCP/UDP (Transmission Control Protocol - User Datagram Protocol), lo que permite un control aún mayor de la red [36].

1.2.5 SDN en redes ópticas

Las redes ópticas han sido objeto de estudio junto a SDN con el fin de lograr la simplificación de las mismas, al mismo tiempo que se logra explotar los beneficios de la centralización del control. Las redes ópticas definidas por software o SDON (Software-Defined Optical Network), llevan el concepto de SDN a las redes ópticas, abstrayendo el plano de control de los conmutadores ópticos a una entidad central al igual que el SDN convencional, para ello se hace uso de conmutadores ADD-DROP reconfigurables [43], [44].

El uso de Flexgrid junto a la centralización del plano de control óptico permitirán el desarrollo de nuevas redes de datos capaces de soportar de manera eficiente canales de 400 Gbps o incluso de hasta 1000 Gbps, maximizando el uso del espectro óptico. El uso de un elemento de computación del camino o PCE (Path Computation Element) en un nodo central será de gran utilidad para el cálculo de las rutas ópticas, logrando así la transparencia de la red de extremo a extremo, evitando la tarea de convertir longitudes de onda en los nodos intermedios de la ruta escogida, [37]. Una posible desventaja de la implementación de las SDON en el futuro será la alta complejidad para el trazado de las rutas ópticas, pues, la capa óptica cuenta con varias limitantes como los dispositivos de red usados, las longitudes de onda disponibles y las restricciones de transmisión, que deben ser consideradas de manera conjunta por el controlador para la asignación de rutas, al mismo tiempo que evalúa todas las solicitudes de conexión y envío de información sobre toda la red, siendo este un motivo que aun limita el total despliegue de esta tecnología aún en desarrollo [43], [46], [54].

El uso de una capa óptica programable da lugar a lo que se conoce como “servicio óptico de longitud de onda bajo pedido”, en el que la cantidad de espectro requerido es solicitada al controlador, el cuál asigna el espectro necesario para la conexión, cabe resaltar que esto se logra haciendo uso de la tecnología Flexgrid. Si bien los avances en el tema del plano de datos para SDON hacen de esta una tecnología prometedora, aún se encuentran inmaduros los avances respecto al control unificado, planificación, gestión y optimización de la red mediante un controlador centralizado [43], [46].

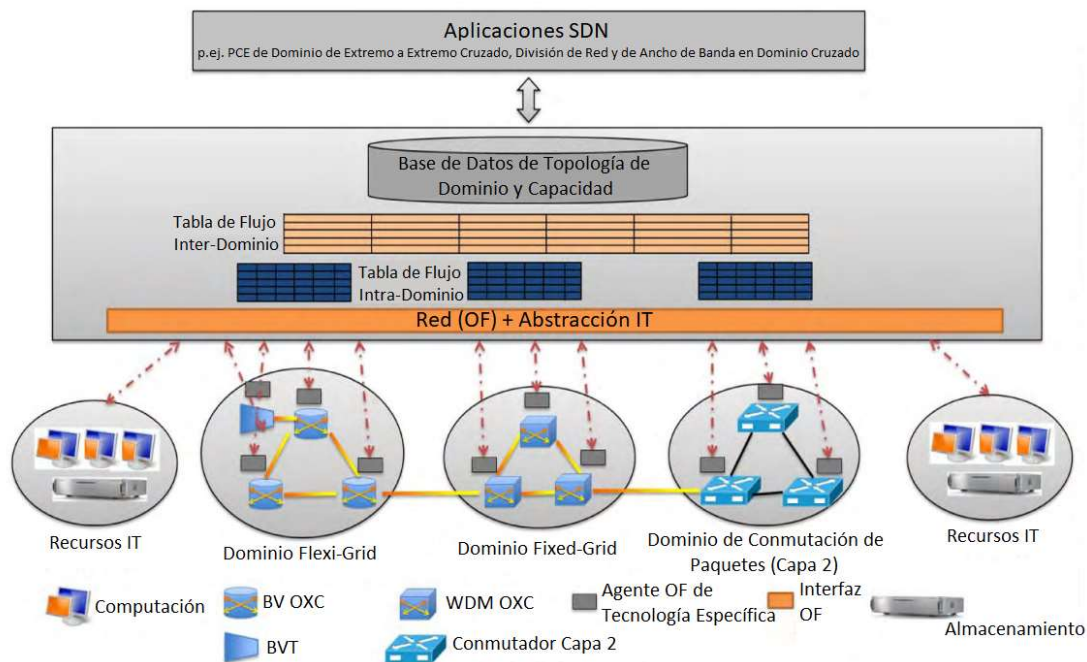


Figura 1.20 - Modelo de arquitectura para SDON [46]

En la figura 1.20 se puede observar la arquitectura propuesta para el establecimiento de SDON. Si bien esta arquitectura conserva mucha similitud con SDN convencional, son necesarias todas las abstracciones del dominio óptico y el establecimiento de parámetros que determinen el funcionamiento del controlador, como por ejemplo el ancho de ranura óptica a implementar, la tecnología (WDM, Flexgrid, etc.), técnica de conmutación, niveles de potencia y otros, lo que hace de SDON una tecnología prometedora a mediano-largo plazo, pero inviable masivamente por el momento [46, 47].

Gracias a los avances en los elementos hardware de las redes ópticas, los transceptores (dispositivos que cuenta con un transmisor y receptor de información), que antes eran dispositivos fijos y no modificables, ahora pueden ser controlados según las necesidades del momento, permitiendo un control mucho mayor sobre la red, esto combinado con SDON, promete maximizar la utilización y desempeño de las redes [48]. En la figura 1.21 se observa un esquema para un transceptor ajustable para redes ópticas y algunos de los parámetros que podrían ajustarse.

Dado que en el plano de control pueden abstraerse muchas de las funciones que se encuentran embebidas en los dispositivos de red actuales, el controlador SDN puede tornarse bastante complejo y extenso, por lo que, para el desarrollo de

este trabajo de grado, se realizará una aproximación del proceso de centralización, como se verá más adelante.

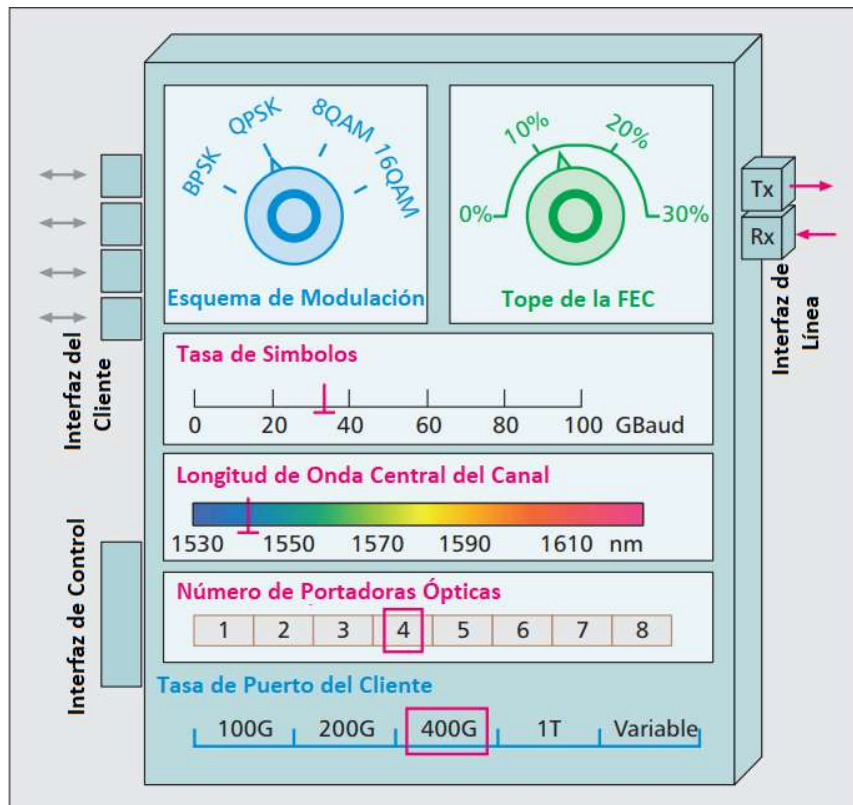


Figura 1.21 - Representación esquemática de un transceptor óptico ajustable [48]

1.3 Metaheurísticas y Machine Learning

1.3.1 Metaheurísticas

La metaheurística puede definirse como el uso de la inteligencia para resolver un problema, haciendo uso óptimo y razonable de los recursos de los que se dispone, especialmente el tiempo. En el campo de la informática este término ha sido acuñado para describir las técnicas utilizadas para la resolución de problemas reales usando de forma óptima los recursos hardware y software de los que se dispone, a fin de hallar una solución adecuada más no necesariamente ideal o exacta para el problema en cuestión, siendo soluciones de tipo no deterministas, por lo que aplicar varias veces la misma técnica al mismo problema puede no resultar en la misma solución. Las metaheurísticas caen dentro de la categoría de machine learning de la que se tratará más adelante en este capítulo [49, 50].

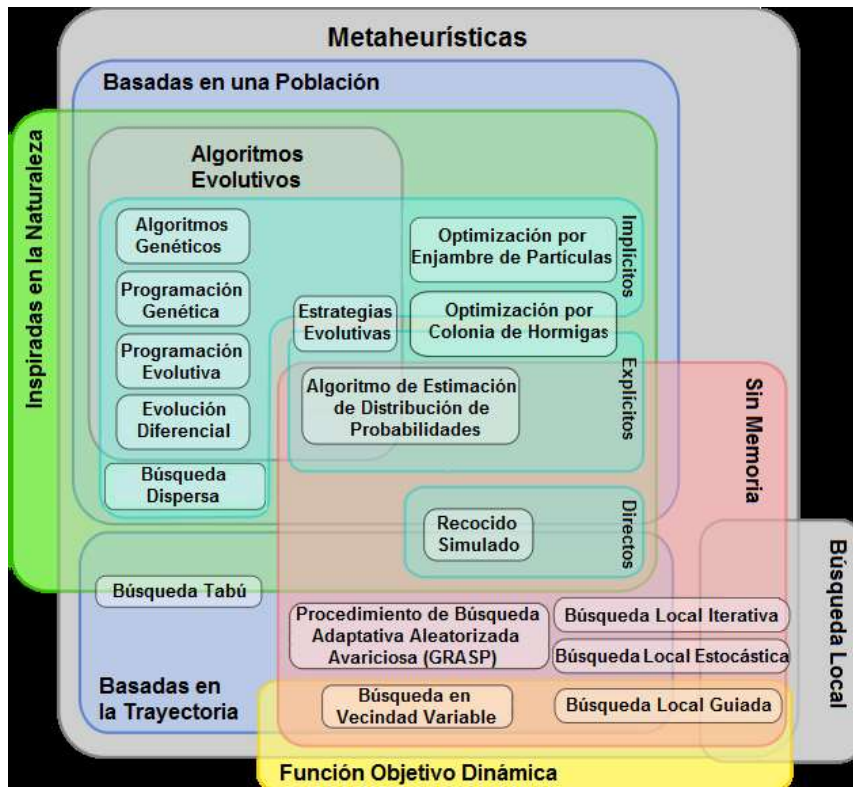


Figura 1.22 - Clasificación de las metaheurísticas más usuales [51]

Algunos de las clasificaciones de metaheurísticas más usuales son, por ejemplo [50]:

- **De relajación:** que buscan descomponer un problema complejo en varios más pequeños y fáciles de resolver.
- **Constructivas:** que buscan hacer una selección paulatina de los elementos que conforman una posible solución al problema tratado.
- **De búsqueda:** guían procedimientos para recorrer el espacio del problema buscando soluciones alternativas.
- **Evolutivas:** Se basan en la evolución de soluciones a partir de posibles soluciones previas al problema.

Existen metaheurísticas que no se encuentran claramente clasificadas dentro de alguna de las categorías anteriores, por utilizar elementos de dos o más de ellas, como por ejemplo las metaheurísticas de colonias de hormigas o redes

neuronales, entre otros [50]. En [52] pueden encontrarse descripciones detalladas de algunas de las metaheurísticas más populares. En la figura 1.22, se pueden observar algunas de las metaheurísticas más usuales, dentro de su categoría. Como se menciona previamente, existen varias formas de clasificar las metaheurísticas, una forma es diferenciar las técnicas que utilizan una única solución para explorar S (espacio de posibles soluciones), llamadas metaheurísticas basadas en trayectoria, mientras que otras técnicas utilizan una población entera de soluciones para establecer diferentes puntos de búsqueda y lograr así una exploración de S más completa. Las técnicas citadas son ampliamente usadas en el campo de las metaheurísticas gracias a su relativa facilidad de implementación y resultados. En la figura 1.23, se puede ver una clasificación de esos tipos de técnicas.

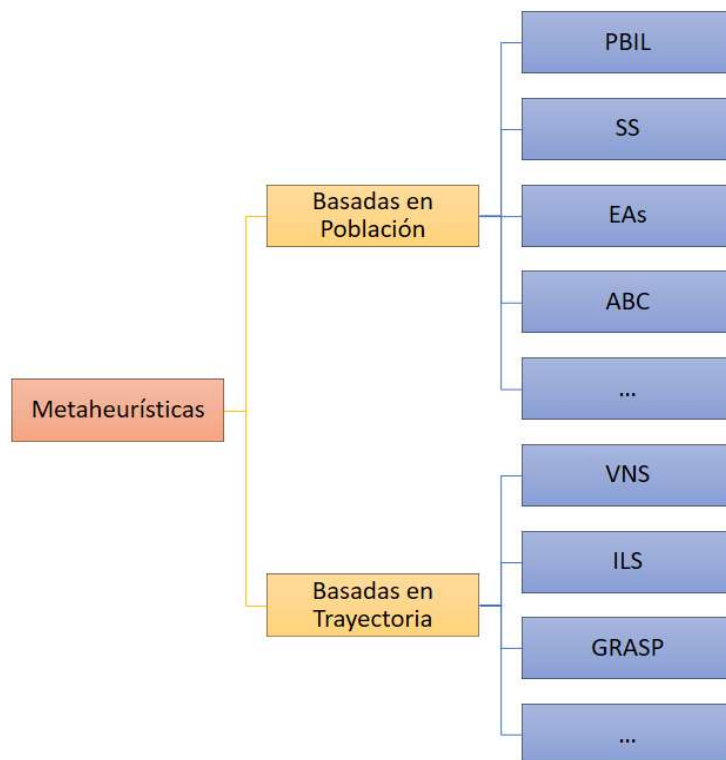


Figura 1.23 - Clasificación de metaheurísticas [53]

Las metaheurísticas son ampliamente usadas para intentar dar una solución a los problemas de tipo NP-completo, es decir aquellos problemas que no poseen una solución en tiempo polinómico, por lo que el gasto computacional para resolverlos crece exponencialmente sin llegar a ser suficiente para dar con la respuesta [49]. La asignación de espectro de forma dinámica será abordada desde la perspectiva de las metaheurísticas, buscando dar una solución óptima al problema.

Las metaheurísticas usualmente implementadas están inspiradas en el entendimiento y abstracción de los procesos llevados a cabo en la naturaleza, como puede verse en [54], lo que ha dado origen a metaheurísticas como las colonias de hormigas, redes neuronales, algoritmos genéticos, colonia de abejas y muchos otros.



Figura 1.24 - Ejemplos de fuentes de inspiración para las metaheurísticas [55]

Para dar solución a los problemas de optimización, como es el caso de este trabajo de grado, son propuestos los algoritmos de evolución que usan un proceso iterativo basado en procesos biológicos para dar con la posible solución al problema. También existen los algoritmos basados en enjambres, que utilizan el comportamiento colectivo de animales como peces, aves, manadas o insectos para dar con la solución al problema [55], como lo muestra la figura 1.24. Será decisión de los autores de este trabajo de grado la elección de la metaheurística a implementar para dar solución al problema de asignación de espectro, pudiendo ser una adaptación de una técnica ya implementada, o bien, una nueva propuesta que se adecúe al problema enunciado.

1.3.2 Machine Learning

El aprendizaje de máquina o machine learning o ML, es un subcampo de la inteligencia artificial, que busca que los programas software “aprendan”, haciendo un estudio y generalización de su entorno. Esta rama de las ciencias de la computación tiene grandes y diferentes campos de acción, entre los que se destacan la medicina, el análisis bursátil, motores de búsqueda y más [56], [57]. También es aplicable al campo de las telecomunicaciones gracias a la capacidad de aprender a cerca del estado de las redes que junto con la cognitividad de la que se hablará más adelante, hacen posible la “auto-gestión”. Puede decirse entonces que: “*el aprendizaje de máquina se define como cualquier algoritmo que mejora su desempeño a través de la experiencia obtenida o ganada durante un periodo de tiempo sin toda la información sobre el entorno en el cual está operando* [13]”. Dentro del aprendizaje de máquina caben las técnicas o

metaheurísticas tratadas en la sección anterior. En [13], se pueden encontrar los tipos de aprendizaje de máquina que se detallan brevemente a continuación:

- **Aprendizaje supervisado:** Permite la búsqueda de patrones en campos de datos históricos para relacionarlos con un “campo objetivo”. Más usado en problemas de clasificación y regresión, requiere entrenamiento previo.
- **Aprendizaje no supervisado:** Se busca la estructura correspondiente a un grupo de campos de datos históricos no etiquetados, a fin de organizarlos. Puede requerir entrenamiento previo.
- **Aprendizaje basado en memoria:** Vecinos más cercanos y modelos de aprendizaje *lazy* y *eager*.
- **Aprendizaje semi-supervisado:** Cuando no es posible etiquetar todos los campos de datos.
- **Aprendizaje activo (con refuerzo):** Se premia a un agente con un refuerzo positivo cuando realiza la acción adecuada o se le castiga sin dar refuerzo cuando se equivoca. En la figura 1.25 se puede ver una representación de esta técnica.
- **Aprendizaje en línea:** Se realiza una pasada sobre un gran volumen de datos y en tiempo real. El objetivo de este tipo de aprendizaje es la extracción de la mayor cantidad de información útil con el menor número de pasadas. Puede ser supervisado o no.

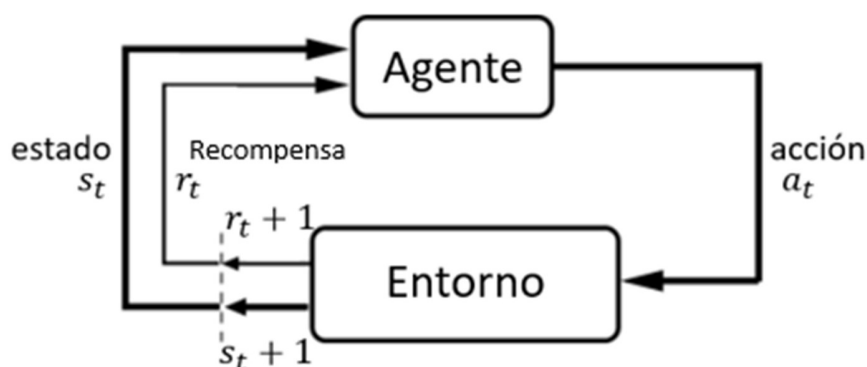


Figura 1.25 - Aprendizaje activo con refuerzo [13]

- **Formulación de problemas dentro del aprendizaje de máquina**

Existen diferentes maneras de formular un problema dentro del aprendizaje de máquina, a continuación, se describen algunas de las maneras más relevantes [13], [57].

- **Aprendizaje por clasificación y regresión:** básicamente el proceso de clasificación involucra asignar un caso de prueba para uno de un conjunto finito de clases, mientras que la regresión permite predecir el caso empleando variables continuas o atributos, como ejemplo tomado de [13] se tiene: *“Para el contexto del diagnóstico de redes un problema de clasificación puede consistir en decidir si la falla en una conexión se debe a que el sitio tomado como objetivo se ha caído, el sitio objetivo se ha sobrecargado, o si el prestador del servicio está fuera de servicio. Por analogía, el problema de regresión puede consistir en predecir el tiempo que toma la conexión en retornar a su estado funcional.”*
- **Aprendizaje para actuar:** Aquí se seleccionan planes de actuación para que sean ejecutados por un agente. Se puede llevar un registro de las acciones pasadas de forma tal que la ejecución de una nueva acción lleve menos tiempo, lo que se conoce como “clonación del comportamiento”.
- **Aprendizaje para interpretación y entendimiento:** Se busca poder interpretar situaciones y eventos. Mediante el entendimiento puede identificarse alguna instancia dentro de una clase, mientras que la interpretación requiere que se usen modelos profundos para explicar los datos del ambiente.

También es posible la implementación del aprendizaje con refuerzo que fue descrito previamente. La creación de conjuntos de soluciones deseables y no deseables es viable, dándole al agente la capacidad de aprender que acciones son adecuadas y cuáles no.

Tabla 1.3 - Resumen de formulación de problemas en aprendizaje de máquinas [13]

Formulación		Tarea desempeñada
Clasificación	y	Predice y dando x
regresión		Predice el resto de x dada una parte de x Predice P(x) dado x
Actuación	y	Selecciona iterativamente una acción a en un estado s
planeación		Selecciona acciones $\langle a_1, \dots, a_n \rangle$ para alcanzar una meta g Encuentra el valor a fijar s para optimizar un objetivo J(s)
Interpretación	y	Empleo de un parser o analizador sintáctico que convierte un flujo de datos en una estructura de árbol de análisis de objetos o eventos.
entendimiento		

1.3.3 Cognitividad en redes de datos

El avance de la informática ha permitido dotar con cierto nivel de “inteligencia” a los programas software o dispositivos existentes mediante la aplicación de técnicas como la “cognitividad” que se enmarca dentro del campo del aprendizaje de máquina, explicado previamente, haciendo que los programas informáticos sean capaces no solo de aprender sino también de “pensar” por sí mismos y tomar decisiones adecuadas según la situación bajo la que se encuentren operando.

La necesidad de evolucionar las redes de telecomunicaciones ha llevado a que estas técnicas sean trasladadas al dominio de las redes de datos, dando lugar a lo que se conoce como “networking cognitivo” o “interconexión cognitiva de redes”, en donde se busca que la red sea capaz de lograr la “auto-gestión”, “auto-conciencia” y “auto-cuidado” (“auto-reparación”), para optimizar la operación de la misma [13], [58].

Según [13] *“una red cognitiva o CN es capaz de percibir las condiciones actuales del entorno, planificar, decidir y actuar sobre esas condiciones, teniendo la opción de aprender de los sucedido y actuado mientras sigue unas directrices de desempeño establecidas previamente, para ello hace uso del concepto de ciclo cognitivo”*. Una red cognitiva va acompañada de un plano de cognición, donde se almacenan las condiciones de red, se realiza razonamiento, emite juicios y reconfigura la red según los parámetros establecidos de extremo a extremo.

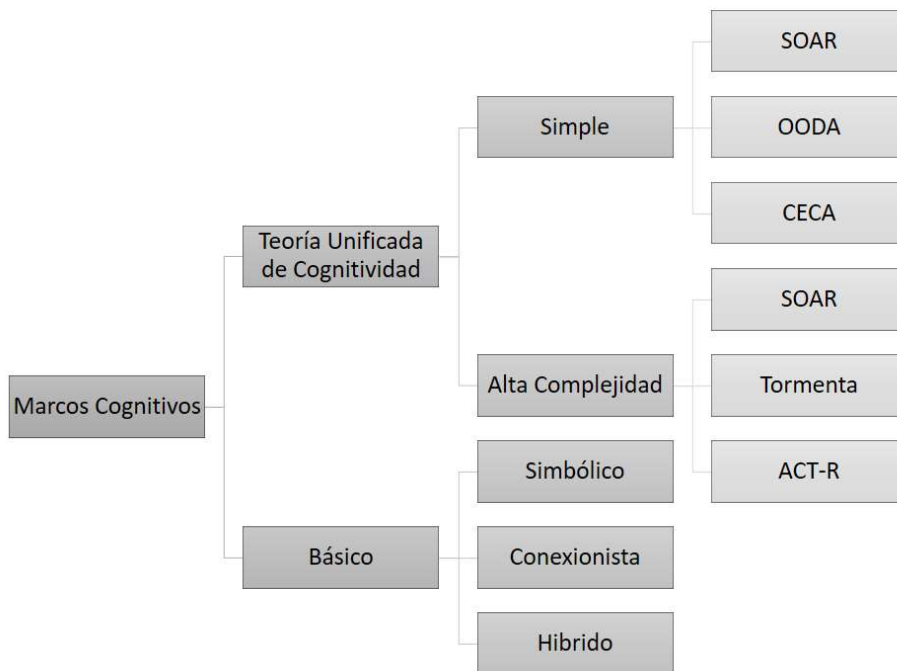


Figura 1.26 - Clasificación de arquitecturas cognitivas [59]

El núcleo de la cognitividad es un algoritmo encargado de observar, analizar y aprender del estado de red, que al combinarse con la técnica SDN, estará en un nodo central desde donde tendrá una visión holística de la red.

Según [59], existen dos tipos de arquitecturas cognitivas, la primera, las basadas en la teoría unificada de la información y la segunda las de naturaleza más simple, esto puede observarse en la figura 1.26. Para la aplicación de alguno de estos métodos de cognitividad, se hace necesario hacer uso de lo que autores como [60] o [61] han denominado “ciclo cognitivo” o “lazo cognitivo”, que consta de seis procesos para lograr la “cognición²” de la red. Si bien este ciclo fue desarrollado originalmente para ser adaptado a la tecnología de radio cognitiva, gracias a sus características y funcionamiento, ha sido acomodado para el desarrollo de nuevas tecnologías de telecomunicaciones, llevándolo así al plano de las redes alambradas como lo sugiere [62]. En la figura 1.27, se observa una imagen del ciclo cognitivo, compuesto por las etapas de observación, orientación, decisión, planeación, acción y aprendizaje, que serán descritas brevemente.

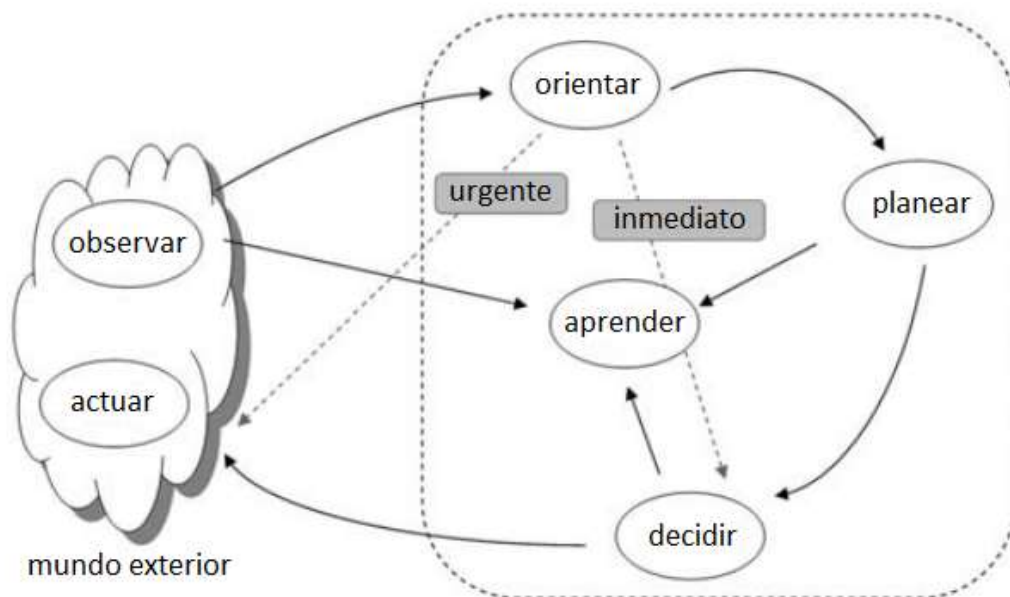


Figura 1.27 - Ciclo cognitivo [60]

- **Observación:** Se recopila información acerca del estado de la red.
- **Orientación o análisis:** Se evalúa la importancia de la información recolectada en el proceso anterior.

² Cognición en este ámbito no se refiere al nivel cognitivo alcanzado por los humanos, sino por una máquina con instrucciones dadas.

- **Decisión:** Se toma en consideración la información y se escoge la mejor³ alternativa para actuar.
- **Planeación:** La información recolectada puede usarse para trazar una línea de acción a más largo plazo.
- **Acción:** Después del proceso de orientación, decisión o planeación, el algoritmo puede pasar a esta fase, donde realiza los ajustes necesarios a la red, en base a lo aprendido durante el ciclo.
- **Aprendizaje:** En esta fase se evalúa el desempeño de la red y se evalúa su cambio respecto a la iteración anterior del ciclo, aprendiendo así cuáles son las “mejores” decisiones en base a la experiencia.

Según [13] y [72] un sistema o algoritmo debe poder realizar algunas de las tareas del ciclo cognitivo, incluyendo observación y acción, para poder ser llamado cognitivo. A medida que se añaden más tareas del ciclo cognitivo, el algoritmo o sistema alcanza un nivel de cognición mayor incrementando su complejidad y alcance.

El ciclo cognitivo ha sido adaptado para su operación sobre redes de datos, en donde un nodo puede tomar una decisión que lo afecte a nivel local (ej., actualizar sus tablas de enrutamiento) o que afecte de forma amplia el desempeño de toda la red (ej., cambiar el esquema de modulación de la red), por lo que la redefinición del ciclo cognitivo para hacer frente a este caso es mostrada en la figura 1.28.

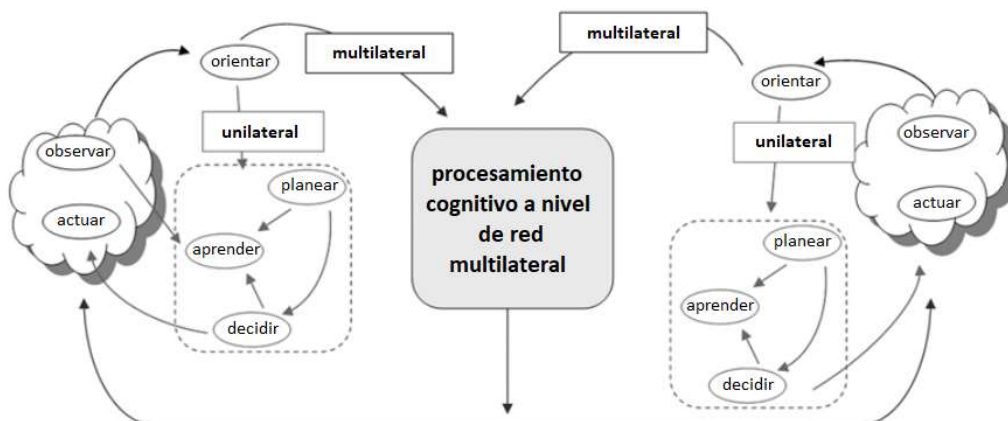


Figura 1.28 - Ciclo cognitivo en redes cognitivas [60]

³ La mejor alternativa se refiere a lo que considere el algoritmo, que puede diferir de la realidad del proceso.

De la figura anterior es importante rescatar el hecho de que existen dos niveles distintos de procesamiento de cognitividad, a nivel de nodo y a nivel de red, en donde se pueden tomar decisiones de tipo unilateral (que afectan únicamente al nodo que las toma) o multilateral (que afectan a la red a una escala mayor a un nodo), siendo esta última, la cognitividad a nivel de red, que presenta mayor interés para el desarrollo de este trabajo de grado.

1.3.4 Elementos básicos en una red cognitiva

Para poder llamar “cognitiva” a una red, debe cumplir con los procesos de observación, análisis (orientar), decisión y acción, en la figura 1.29, estos procesos se ubican dentro de una pirámide, en cuya base se ubican los elementos actuadores y recopiladores de información, la inteligencia de los dispositivos aumenta a medida que se asciende en la pirámide, ubicando el algoritmo de control en la cima de la misma. Cabe mencionar que las tecnologías habilitadoras de las redes cognitivas deben poder ser escalables desde un nodo a toda la red [13].

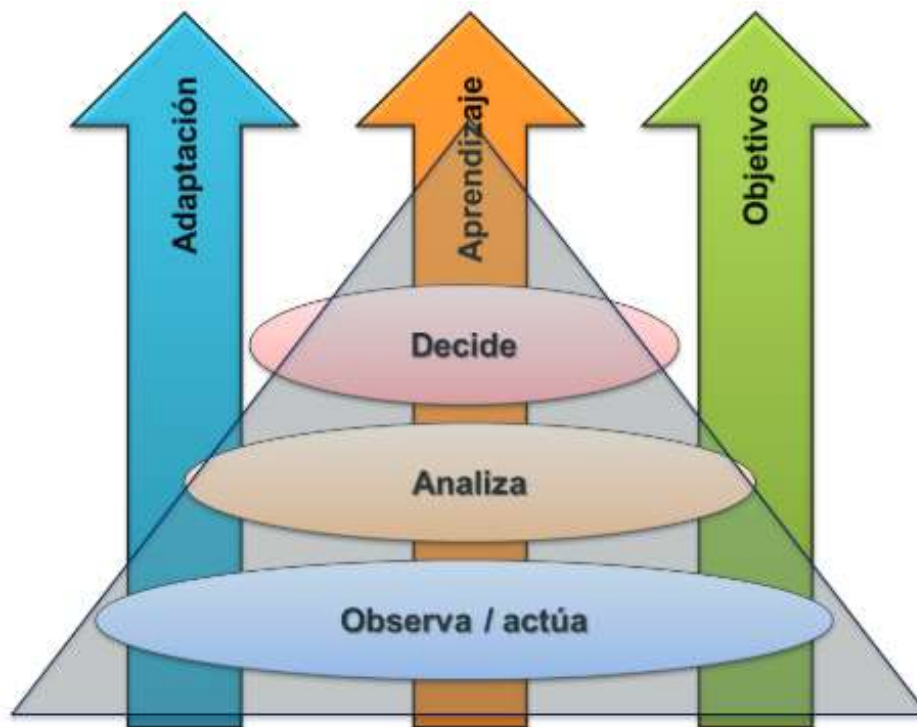


Figura 1.29 - Representación de los procesos cognitivos por planos [13]

De la figura 1.29 se observa que la cognitividad está regida por tres planos, adaptación, aprendizaje y objetivos, los cuales se describen a continuación:

- **Adaptación:** Es una parte esencial de los elementos funcionales y debe responder a los cambios en el entorno de la red.
- **Objetivos:** Contiene el conjunto de metas a alcanzar mediante la aplicación del aprendizaje y la optimización cognitiva en la red. Estos objetivos pueden considerar un comportamiento extremo a extremo y responder a políticas de calidad del servicio.
- **Aprendizaje:** Es el núcleo del sistema cognitivo y se encarga de habilitar a los demás elementos de la red, para llevar a cabo acciones basadas en la experiencia previa o conclusiones almacenadas en el sistema que podrían afectar decisiones futuras.

Es importante la existencia de una vía de realimentación para que el algoritmo cognitivo juzgue si los ajustes hechos llevan la red a operar en la dirección correcta o, por el contrario, la red empeora su funcionamiento tras un cambio, de ser así, el algoritmo deberá retroceder y volver a evaluar la siguiente acción a tomar.

Capítulo 2

ALGORITMOS, HERRAMIENTAS Y METODOLOGÍAS

En este capítulo se describirán algunos de los algoritmos utilizados actualmente para la asignación de espectro en redes ópticas, para posteriormente elegir aquellos que puedan ser de utilidad. También se hará un recuento y descripción de las herramientas software disponibles (Matlab, OptSim, OMNeT) para el desarrollo de este trabajo de grado, finalmente se hará la elección de una de ellas. Para cerrar el capítulo, se hablará acerca de la metodología de trabajo escogida y su aplicación al trabajo de grado.

2.1 Algoritmos Actuales de Redes Ópticas para la Asignación de Recursos de Red

La necesidad de hacer una gestión adecuada de los recursos ha llevado al desarrollo de numerosos algoritmos para controlar parámetros de red como: enrutamiento, anchos de banda, políticas de uso, calidad de servicio, etc. A continuación, se hará un recuento de algunos de los algoritmos de asignación de espectro para redes ópticas RSA Flexigrid implementados, describiendo su modo de operación y características relevantes según la literatura consultada.

Dentro de RSA, se pueden distinguir dos tipos de algoritmos de asignación de espectro, los estáticos u offline y los dinámicos u online [35], los cuales serán detallados a continuación.

2.1.1 RSA Estático

En RSA estático u offline, los requerimientos de tráfico de la red y la topología implementada son conocidos previamente y se mantienen constantes todo el tiempo, por lo que se facilita el establecimiento de rutas y de recursos para la operación de la red [64]. Diversos tipos de algoritmos basados en heurísticas han sido propuestos para abordar el problema RSA, ya sea de manera conjunta (RSA) o de manera secuencial (R+SA), siendo la segunda forma, la que requiere menor cantidad de recursos computacionales, pero sin dar garantías de encontrar la solución óptima al problema.

2.1.2 RSA Dinámico

Aquí, la topología de red es conocida, pero las demandas de tráfico no lo son, lo que hace del problema RSA más complejo de resolver, debido a la inherente naturaleza aleatoria del tráfico de red [64], [65]. La solución dinámica es de suma utilidad para los casos de restauración de una conexión, atención de nuevas solicitudes cuando las demás ya se encuentran establecidas, liberación y reasignación de recursos, por lo que este modo de RSA es el que más se adecúa a las condiciones de red actuales.

El problema RSA puede atacarse según el funcionamiento del algoritmo, ya sea, con algoritmos de una etapa (one-step) o dos etapas (two-steps).

- **Algoritmos de una etapa:** Estos algoritmos solucionan el problema RSA de manera conjunta, es decir, abordan el problema de enrutamiento y de asignación de espectro al mismo tiempo [27], [28]. Como ejemplo se tienen los algoritmos:
 - **Modified Dijkstra's Shortest Path (MSP):** Encuentra la ruta más corta para la conexión y verifica la cantidad de espectro disponible [28].
 - **Spectrum-Constraint Path Vector Searching (SCPVS):** Construye un árbol de vectores de caminos con la restricción de espectro, para buscar la ruta óptima global [28].
- **Algoritmos de dos etapas:** El problema RSA se aborda de manera secuencial, realizando primero el cálculo de la ruta (problema R), para después abordar el problema de asignación de espectro (problema SA) [28]. En [27] sugieren el cálculo de la ruta más corta mediante el algoritmo Dijkstra u otro equivalente, para luego abordar el problema SA con algún otro algoritmo. Esto permite asignar recursos de espectro habiendo

calculado la ruta de ante mano, haciendo más eficiente la gestión de recursos.

2.1.3 Algoritmos RSA

Diferentes algoritmos basados en diversas heurísticas han sido propuestos para abordar el problema RSA ya sea de forma dinámica o estática. Para este trabajo de grado, el algoritmo del “camino más corto” o Dijkstra será usado para resolver el problema de enrutamiento, mientras que uno de los siguientes algoritmos será usado para comparar el algoritmo diseñado para la asignación de espectro:

- **First Fit (FF):** Una vez calculadas las k rutas disponibles para la conexión, todos los slots son enumerados, entonces, el algoritmo, busca de forma ascendente según el índice de slots, los slots consecutivos necesarios para el establecimiento de la conexión y selecciona la primera ruta encontrada y los slots que cumplen con el requisito de conexión. Este algoritmo es la versión modificada el algoritmo first-fit de WDM [28], [28], [65].
- **Random Fit (RF):** Al igual que FF, RF encuentra las rutas disponibles y luego asigna los slots de frecuencia necesarios de forma aleatoria de entre todos aquellos que cumplan los requisitos de conexión a la primera ruta encontrada [28].
- **Smallest Fit (SF):** Se diferencia de los anteriores en que asigna la conexión a la banda espectral disponible más pequeña a lo largo de la primera ruta encontrada [27], [28], [65].
- **Lowest Starting Slot (LSS):** Una vez calculados los posibles caminos o rutas de antemano, el algoritmo busca los slots consecutivos necesarios para la conexión, en orden ascendente según el índice de slots y selecciona el camino que contenga el slot inicial más bajo [27], [28], [65].
- **LAUC-Void Filling (VF):** El camino con la menor cantidad de “huecos” consecutivos es seleccionado para establecer la conexión, escogiendo aquellos slots que se encuentren más cerca temporalmente hablando. Con esto se busca disminuir la fragmentación del espectro [27].

2.2 Herramientas para el Desarrollo del Trabajo

Existen diferentes herramientas o plataformas de simulación destinadas a evaluar el desempeño de las redes de datos. Algunas plataformas están enfocadas en simular eventos en el plano físico, mientras que otras se enfocan más en los planes superiores de la arquitectura OSI, es por esto que la elección de una adecuada plataforma de simulación es de crucial importancia para el desarrollo investigativo como en este trabajo de grado, donde se debe simular el comportamiento de algoritmos RSA en la red NSFNet de la que se hablará más adelante.

Dado que dentro del entorno de la universidad han sido ampliamente utilizadas varias plataformas, a continuación, se hará un recuento de algunas de ellas, listando sus principales características y eligiendo la que se considera más adecuada para desarrollar este trabajo.

2.2.1 MATLAB

MATLAB (Matrix Laboratory) es una herramienta informática y lenguaje de programación propietario (MathWorks) de alto nivel, destinado a la realización de cálculos matemáticos, especialmente operaciones de matrices. Permite la realización de cálculos complejos, creación y manejo de gráficas, creación de interfaces, interacción con otros entornos computacionales, simulación de eventos, procesamiento de señales, etc., siendo ampliamente usado en muchos campos, desde la física, química, hasta prácticamente en todas las ramas de la ingeniería [66], [67].

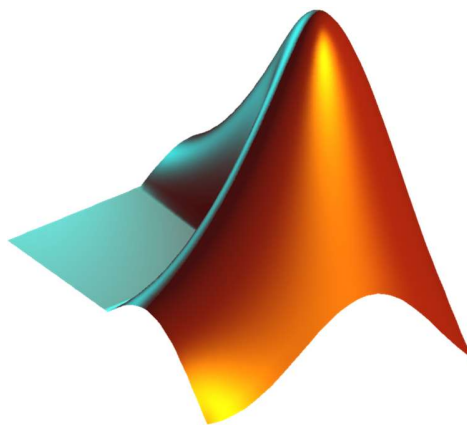


Figura 2.1 - Logo de MATLAB [67]

MATLAB puede ser utilizado mediante la aplicación de comandos o la ejecución de scripts de usuario escritos en lenguaje Matlab, siendo este lenguaje de alto nivel y parecido al lenguaje C [68]. La figura 2.1, muestra el logo de esta herramienta.

Si bien MATLAB es usado en el campo de las telecomunicaciones, la simulación de redes de datos fuera del nivel físico, resulta bastante engorrosa y compleja debido a que la herramienta carece de las características necesarias para simular los eventos que ocurren desde el nivel dos del modelo OSI hacia arriba, por lo que es más usado para simular esquemas de modulación, desempeño de enlaces y todo aquello que involucre la capa física de los sistemas de red.

2.2.2 OptSim

Este entorno es ampliamente usado para la simulación de sistemas de fibra óptica debido a su gran repertorio de componentes y poder de simulación. Permite la simulación a nivel de transmisión o capa uno del modelo OSI, siendo ampliamente usado por la industria y el campo educativo. OptSim (de RSOFT) es ideal para evaluar a nivel físico el desempeño de sistemas WDM, dispositivos ópticos, sistemas totalmente ópticos, etc. [69], [70].

Esta herramienta requiere el uso de una licencia para su operación y a pesar de que su uso se basa en la interconexión de componentes modulares, se requiere de cierto nivel de instrucción y conocimiento para explotar las características de este software. Cabe decir que OptSim puede ser usado en conjunto con MATLAB, expandiendo así los potenciales de cada herramienta, en lo que se conoce como entorno co-simulado.

2.2.3 OMNeT++

OMNeT++ es un entorno de desarrollo integrado o IDE (Integrated Development Environment), que permite la simulación de redes de datos, basado en componentes modulares y de arquitectura abierta. Cuenta con una interfaz gráfica que permite el desarrollo de componentes mediante la aplicación del lenguaje de descripción de red o NED, mientras que la programación de los elementos de red se hace con lenguaje C++, lo que facilita su uso [71], [72].

En OMNeT++ módulos simples pueden conformar módulos más complejos, los cuales se comunican mediante el intercambio de mensajes de control, lo que ha hecho de esta plataforma muy útil para simular redes basadas en colas de espera,

arquitectura de hardware, tecnologías móviles, inalámbricas, redes ATM, TCP/IP, redes ópticas, entre otras [72].

2.2.4 Resumen de herramientas de simulación

En la tabla 2.1 se presenta una comparativa de las herramientas expuestas anteriormente.

Tabla 2.1 - Comparación de herramientas de simulación

	MATLAB	OptSim	OMNeT++
Uso investigativo	Alto	Alto	Alto
Tipo de licencia	Comercial	Comercial	Gratuita
Interfaz gráfica	Alta	Alta	Media
Graficación de resultados	Buena	Buena	Buena
Soporte de niveles dos y tres de OSI	Bajo	Bajo	Alto
Implementación de heurísticas	Permite	No permite	Permite

Con el fin de simular la red NSFnet y la implementación de algoritmos RSA heurísticos, la herramienta escogida para el desarrollo de este trabajo de grado es OMNeT++ debido a su licencia gratuita y la capacidad de simular niveles superiores del modelo OSI al mismo tiempo que permite el uso de heurísticas dentro de sus componentes modulares.

2.3 Metodología de Trabajo

Para el desarrollo del trabajo de grado, se ha optado por hacer uso de la metodología de trabajo denominada lineal secuencial o también llamada modelo en cascada, la cual se considera adecuada para abordar el problema debido a sus fases y procesos claros, además de ser ampliamente usada para el desarrollo de soluciones software.

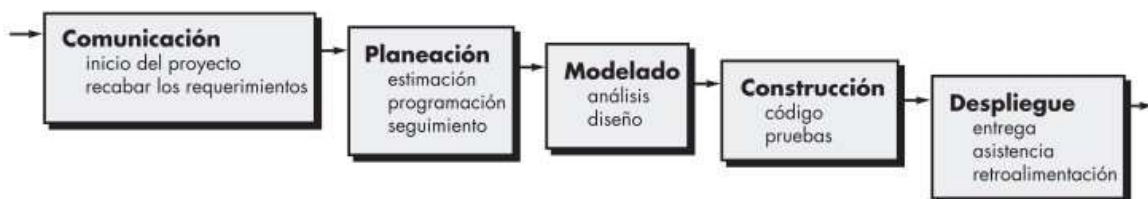


Figura 2.2 - Modelo en cascada [73]

El modelo en cascada está compuesto por cinco fases, las cuales se suceden una tras otra a medida que los requisitos de cada una son completados, permitiendo así avanzar de forma clara y ordenada. Las fases del modelo son: Fase de Recolección de Información o comunicación, Fase de Análisis o planeación, Fase de Diseño o modelado, Fase de pruebas o construcción, y Fase de Entrega o despliegue, las cuales serán brevemente descritas a continuación. [73]. En la figura 2.2, se puede observar el proceso de vida del modelo en cascada.

- **Fase de recolección de información:** En esta fase se busca entender las necesidades finales del cliente y se levantan todos los requisitos funcionales del proyecto de forma clara, de modo que se conozca desde un comienzo que debe poder realizar el software una vez terminado el proyecto.
- **Fase de planeación:** En esta fase se descompone el proyecto en partes más pequeñas que puedan ser desarrolladas por separado o de forma más sencilla.
- **Fase de modelado:** En esta fase se diseña el o los algoritmos que darán la solución al problema de usuario.
- **Fase de construcción:** Aquí se implementa el código desarrollado y se hacen pruebas de funcionamiento del software.
- **Fase de despliegue:** En esta fase se entrega el proyecto y se hace un constante monitoreo del mismo para verificar que cumpla los requisitos de usuario. Se crea un canal de retroalimentación para corregir posibles errores del software.

2.4 Metodología de Simulación

Para el desarrollo de la fase de simulaciones se ha optado por aplicar una metodología con enfoque práctico, basándose en [74]. A continuación, se explicarán con detenimiento las fases comprendidas en este método de simulación.

- **Formulación del problema y definición del sistema:** En esta etapa debe analizarse el problema a resolver y entender cómo interactúan las variables que lo componen entre sí y las posibles restricciones a las que estará sujeto el sistema planteado.

- **Diseño del modelo:** Se debe de diseñar un modelo lógico de la solución cuyos resultados sean los que se esperan obtener de acuerdo a los objetivos. Para esto, debe tenerse claridad acerca de las relaciones lógicas que componen el modelo, sus variables, definiciones y demás parámetros que afecten el comportamiento del sistema a simular.
- **Recolección de información:** Se debe poder identificar y recolectar los datos que sean importantes para la simulación, para después ser analizados.
- **Implementación del modelo:** Una vez hecho lo anterior, se deberá definir herramientas y lenguajes de programación necesarios o adecuados para la implementación del sistema propuesto.
- **Validación:** En esta etapa debe evaluarse el comportamiento del modelo implementado y estudiar si hace lo que se espera, además, verificar si las relaciones y variables planteadas son las adecuadas. Se recomienda si es posible, realizar simulaciones de las que se conozca previamente los resultados para validar el funcionamiento.
- **Análisis:** Una vez realizada la validación, se realizan las simulaciones y se procede a estudiar los resultados para sugerir posibles cambios en el modelo.
- **Documentación:** Se sugiere documentar los aspectos técnicos del modelo propuesto y su modo de operación, en algunos casos es recomendado la creación de un manual de usuario.
- **Implementación:** En esta etapa se considera que es posible poner en funcionamiento el modelo de simulación una vez se hayan efectuado todas las posibles correcciones. La figura 2,3 muestra el diagrama de flujo para el desarrollo de las simulaciones del presente trabajo de grado.

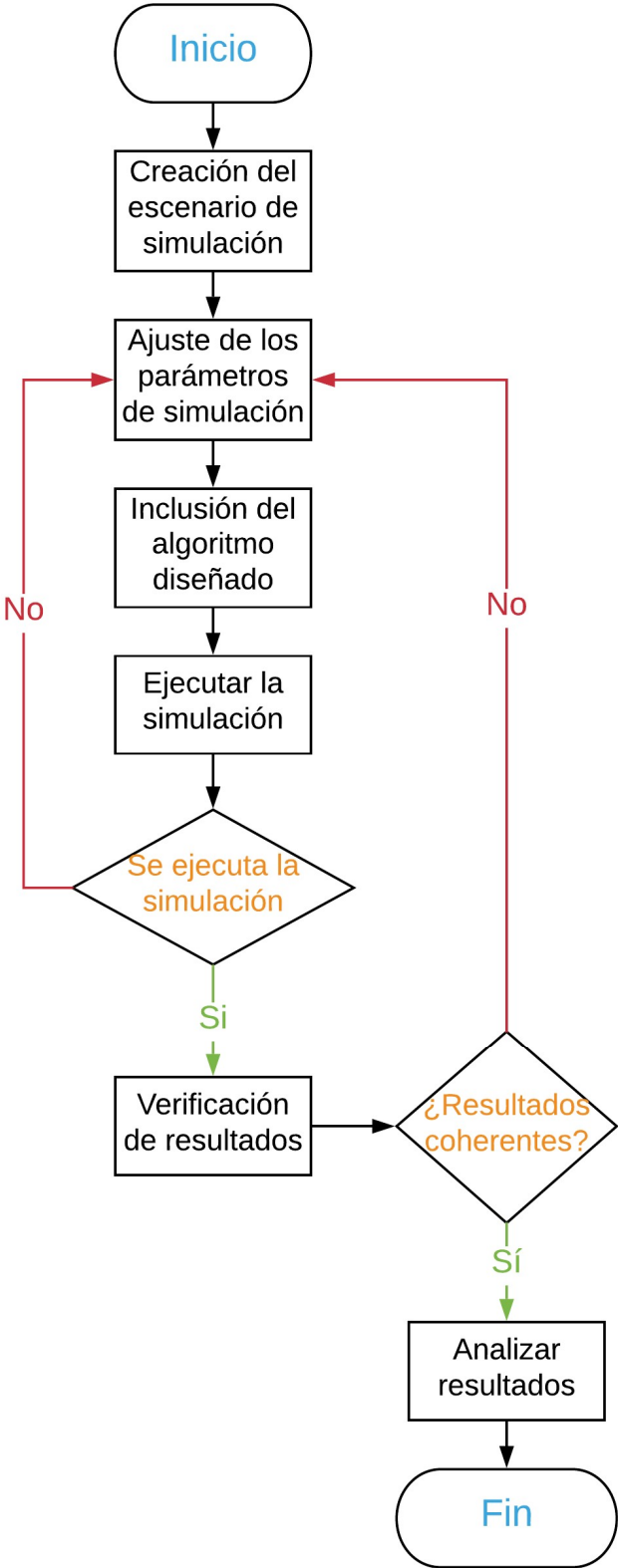


Figura 2.3 - Diagrama de flujo de la simulación

2.5 Definición del Escenario de Simulación

El diseño del escenario de simulación es un proceso principal dentro del desarrollo de este trabajo investigativo, resulta crítico el modelado adecuado de la red y la interconexión de los diferentes elementos que la componen. Es por esto que el diseño y adecuación de la red y sus parámetros lleva varios meses en los cuales se desarrollaron cientos de simulaciones con el fin de verificar la correcta operación de cada uno de los elementos y casos de uso de la red. En los siguientes apartados, se explicarán de forma detallada los módulos que componen la red y la función que cumplen dentro de la misma.

Una red óptica en el IDE OMNeT++ está compuesta por elementos que deben ser definidos dentro del entorno de simulación, siendo los nodos Core o núcleo y los Edge o borde los elementos principales para la operación de la red, por lo que es necesario su definición y construcción a partir de los submódulos que los componen.

Para la creación de los paquetes de datos, su envío a la red, los procesos de conmutación de la información y recepción de la misma, se ha desarrollado un módulo denominado EdgeCore como se observa en la figura 2,4, el cual contiene los nodos Edge y Core. Este módulo es el que representa los centros de datos y conmutación que conforman la red NSFNet, es por esto que se encuentran ubicados en los lugares geográficos como se verá más adelante, los cuales serán detallados en las secciones siguientes.

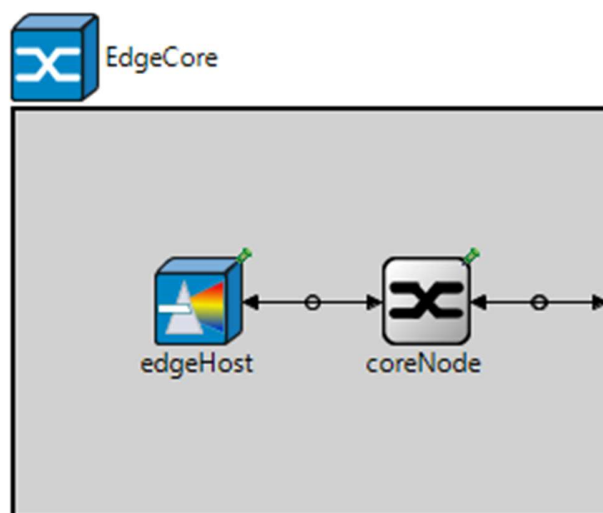


Figura 2.4 - Modulo de nodo Core y Edge

2.5.1 Nodo fuente-destino

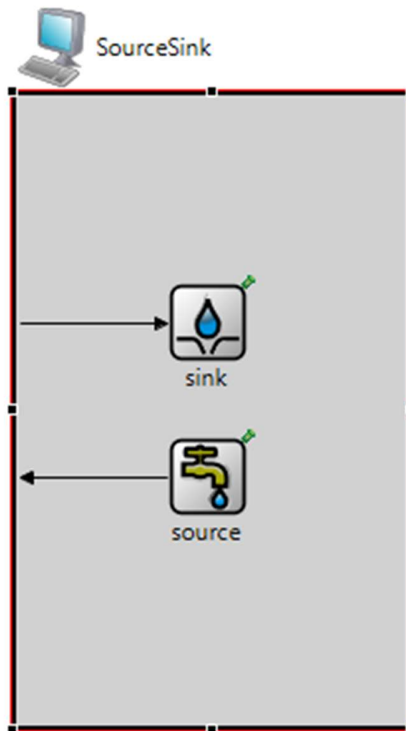


Figura 2.5 - Nodo fuente-destino en OMNeT++

El nodo fuente o source es el encargado de la generación de datos de forma aleatoria siguiendo una distribución uniforme. Los datos generados están compuestos por una dirección de origen, dirección de destino, longitud del paquete de información en bytes, tiempo de generación entre paquetes en segundos y tipo de información. Estos datos son declarados en el archivo de descripción de red (de extensión .NED) y asignados en el archivo de configuración e inicialización (de extensión .ini) del proyecto.

El nodo destino o sink es el encargado de la recepción de los datos provenientes de la red óptica, aquí se realizan los procesos como cálculo de retardos extremo a extremo, cálculos de paquetes perdidos y demás variables que se deseen estudiar. La figura 2,5, muestra el nodo fuente y destino definidos en OMNeT++.

2.5.2 Nodo Edge

El nodo Edge o borde o de frontera es el encargado del armado de paquetes OPS y desarmado de los mismos, agregando o quitando las cabeceras correspondientes (opera de manera equivalente para las redes OBS), por lo que

puede ser de ingreso (paquetes que entran a la red) o de egreso (paquetes que salen de la red y llegan a su destino). Este nodo recibe los paquetes provenientes del nodo Source o fuente, clasifica la información, agrega las cabeceras y se encarga de su envío a través de la red, para ello se ha creado un módulo denominado Assembler o ensamblador. Por otro lado, se ha creado también un módulo desensamblador o Disassembler encargado de remover las cabeceras ópticas y entregar los paquetes al destino correspondiente. La figura 2,6 muestra al nodo Edge definido en OMNeT++.

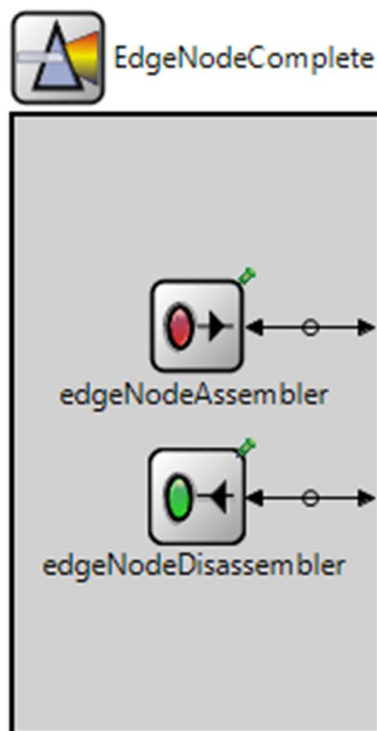


Figura 2.6 - Nodo Edge en OMNeT++

2.5.3 Módulo de ingreso o ensamblador

Este módulo está compuesto de tres entidades encargadas de la clasificación de la información, armado de paquetes y envío de los mismo, estos elementos pueden verse en la figura 2,7, a continuación, se describe la operación de cada sub módulo.

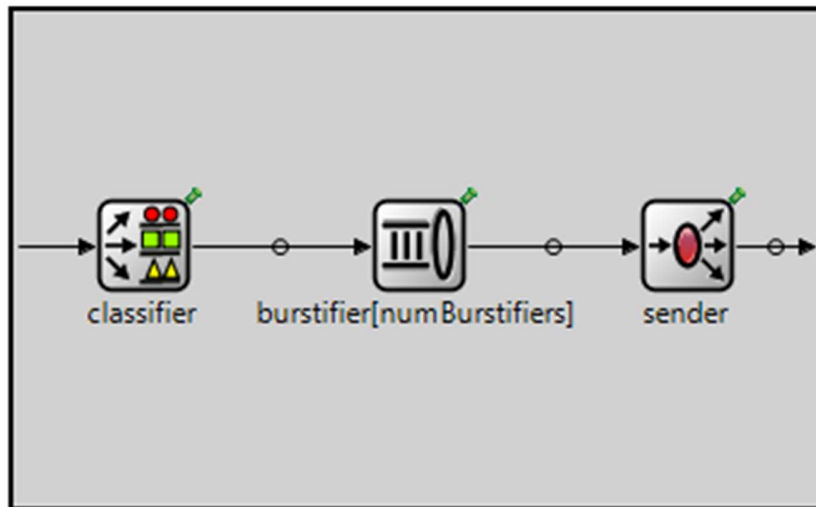


Figura 2.7 - Módulo ensamblador visto en OMNeT++

- **Clasificador:** denominado classifier en la herramienta, se encarga de recibir los paquetes provenientes de la fuente y ordenarlos según su dirección de destino y tipo de servicio.
- **Conformador de ráfagas o burstifier:** es el encargado de el armado de los paquetes con su respectiva cabecera.
- **Planificador de envío o sender:** Este se encarga de la adición de la cabecera OPS y la gestión de recursos para el envío de la información incluyendo la asignación de espectro. Una vez el paquete es conformado se envía a la red la cabecera seguida por el paquete de información. Esta cabecera puede ser modificada de ser necesario cada vez que pasa por un nodo intermedio en la red.

2.5.4 Módulo de egreso o desensamblador

Este módulo está encargado de retirar las cabeceras OPS añadidas a cada paquete y de entregar la información al nodo destino, en donde se calculan los parámetros de funcionamiento de la red como ya fue mencionado.

2.5.5 Nodo Core

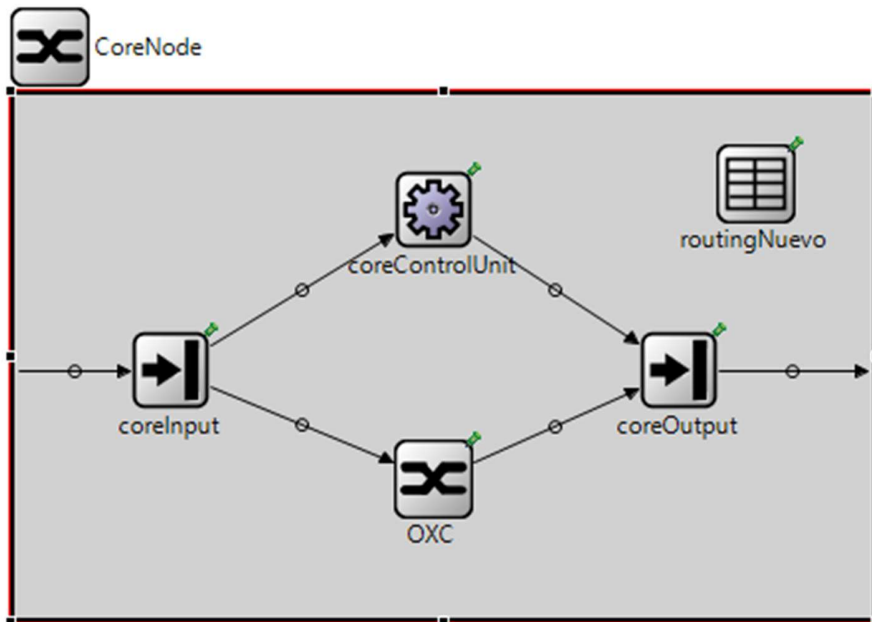


Figura 2.8 - Nodo Core

El nodo core o central está encargado de la conmutación de la información que es enviada a través de la red. El nodo central permite que los paquetes enviados vayan hacia sus destinos sin la necesidad de una conversión óptico-eléctrica, para ello hace uso de una unidad de control encargada de leer las cabeceras ópticas y fijar la matriz de conmutación para el establecimiento de los caminos ópticos. La figura 2,8 muestra el módulo core. Cabe decir que este nodo se encuentra conectado mediante enlaces de fibra óptica al nodo Edge.

- **Interfaz de entrada:** Llamada coreInput, se encarga de la extracción de la cabecera y su envío a la unidad de control, mientras que el paquete es enviado a la matriz de conmutación sin realizar ningún tipo de conversión.
- **Unidad de control:** Denominada coreControlUnit, realiza la lectura de la cabecera óptica, realizando para ello una conversión óptico-eléctrica. Una vez la cabecera se encuentra en el dominio eléctrico, es analizada, permitiendo así que se configuren la matriz de conmutación para permitir el paso del paquete. Después de leída la cabecera, es modificada de ser necesario y vuelve a llevarse al dominio óptico, luego es enviada a la interfaz de salida.
- **Matriz de conmutación o conmutador óptico:** Llamada OXC, permite el paso totalmente óptico del paquete OPS hacia uno de los puertos de que

ha sido establecido previamente por la unidad de control en la interfaz de salida.

- **Interfaz de salida o CoreOutput:** Se encarga del envío de la cabecera y del paquete hacia el próximo nodo de la red a través de uno de los puertos de salida fijado por la unidad de control.

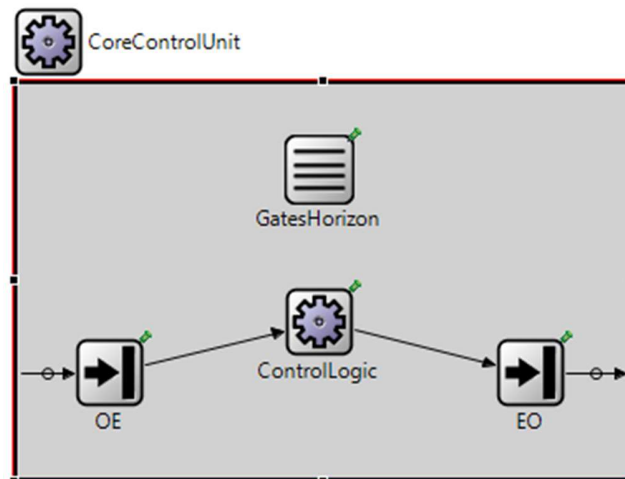


Figura 2.9 - Unidad de control

A parte de los módulos explicados, el nodo central necesita de entidades extra para su operación, es por eso que existe el *módulo de enrutamiento* o *Routing*, que contiene la información de red necesaria para la operación del algoritmo de direccionamiento Dijkstra, explicado previamente. También, un módulo llamado *GatesHorizon* es necesario para la asignación de recursos por parte de la unidad de control. Este módulo contiene información acerca de los canales que dispone la red y su estado de operación. La figura 2,9 muestra cómo está compuesta la unidad de control del nodo central.

2.6 Caracterización de la Red

Ya diseñados e implementados los módulos anteriormente descritos, se procede a interconectarlos siguiendo la topología de la red NSFNet (National Science Foundation Network).

La red NSFNet está compuesta por catorce nodos distribuidos a lo largo y ancho de los Estados Unidos de América, conectando ciudades principales. Esta red ha sido diseñada para la investigación y estudio de las telecomunicaciones.

Posee una topología irregular, donde los nodos tienen una cantidad irregular de conexiones, como puede verse en la figura 2,10, donde se muestra la distancia en kilómetros entre nodos. Esta distancia es utilizada como “peso” de las aristas del grafo de red. Este peso es usado por el algoritmo Dijkstra para el cálculo de la ruta más corta, como se verá en la siguiente sección.

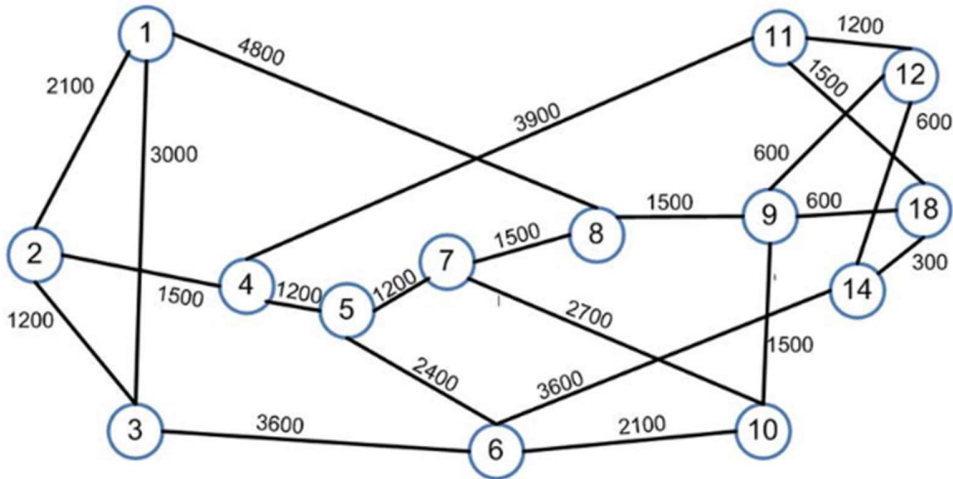


Figura 2.10 - Red NSFNet

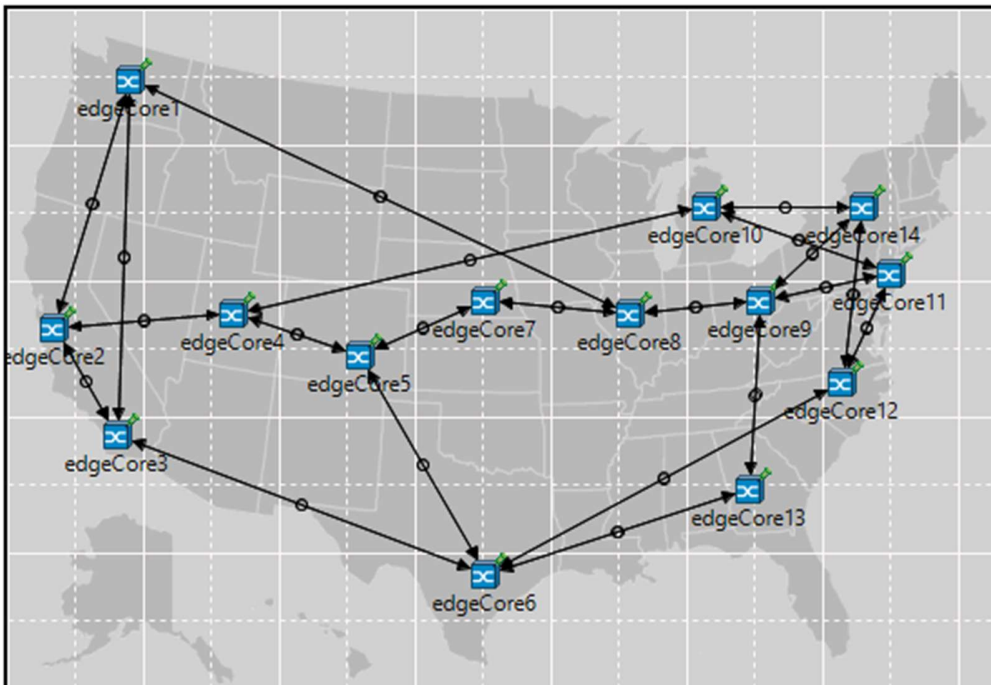


Figura 2.11 - Red NSFNet implementada en OMNeT++

Esta red ha sido modelada en la herramienta de simulación, ubicando cada módulo *EdgeCore* en el lugar geográfico que le corresponde sobre el mapa de Estados Unidos, como puede verse en la figura 2,11. En busca de minimizar el retardo de propagación en la red, se ha determinado que el nodo número nueve, ubicado en Pensilvania es el adecuado para para ejercer el rol de nodo central y así implementar una aproximación a SDN, como lo sugiere [75].

Capítulo 3

DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO PARA LA ASIGNACIÓN DE RECURSOS DE RED

En este capítulo se tratará acerca del proceso de centralización de la red, los algoritmos que operan en la red, el diseño e implementación del algoritmo de asignación de espectro propuesto y los casos de simulación para evaluar su desempeño.

3.1 Algoritmos Implementados

3.1.1 Algoritmo Dijkstra

Como se mencionó anteriormente, el algoritmo de caminos más cortos o Dijkstra, será implementado para solucionar el problema de enrutamiento o problema R de RSA. Este algoritmo es ampliamente usado en toda clase de redes debido a su facilidad de implementación y relativo bajo costo computacional.

El algoritmo de Dijkstra se encarga de encontrar la ruta más corta o de menor peso entre dos nodos de un grafo. Un grafo, es una estructura matemática que consta de nodos y conexiones llamadas aristas, las cuales se encuentran conectadas en sus dos extremos a nodos o posiblemente al mismo nodo en los dos extremos, dichas aristas tienen un valor, peso o costo que es la variable que se busca minimizar mediante la implementación del algoritmo [76].

El algoritmo empieza marcando todos los nodos como no utilizados ubicados a una distancia o peso infinito. Se parte de un nodo inicial conocido visitando sus nodos adyacentes, como Dijkstra es una técnica de metaheurística voraz, para

que un camino sea óptimo utiliza el principio de que todos los caminos contenidos deben ser óptimos, luego entre todos los vértices adyacentes se busca el que se encuentre más cerca (tenga menos peso) al nodo de origen, el cual es tomado como punto intermedio y se evalúa si se puede llegar más rápido desde este nodo a los demás. Después, se escoge el nodo más cercano al nodo inicial y se repite la búsqueda ignorando el nodo del que se partió [76], [77]. A continuación, se muestran los pasos que sigue el algoritmo de forma más detallada.

Partiendo de un grafo de N nodos, donde no hay nodos aislados, se define s como el nodo inicial, un vector D de tamaño $N-1$ guardará al final del algoritmo las distancias desde s al resto de los nodos. En la figura 3,1 se puede observar un ejemplo de un grafo.

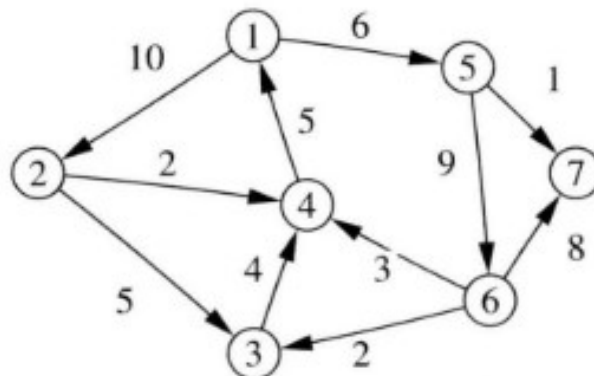


Figura 3.1 - Ejemplo de [77]

1. Inicializar todas las distancias en D con un valor infinito pues son desconocidas al principio, exceptuando la del nodo inicial s que se fija en 0 , debido a que la distancia de s a s sería 0 .
2. Sea $a = s$ (se toma a como nodo actual).
3. Se recorren todos los nodos adyacentes de a , excepto los nodos marcados, se llamarán a estos nodos no marcados vi .
4. Sí la distancia desde s hasta vi guardada 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 encontrada, esto es: sí $(Di > Da + d(a, vi))$ 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 vuelve al paso tres mientras existan nodos no marcados.

7. Una vez terminado al algoritmo, D estará completamente lleno.

A continuación, se muestra el pseudo código correspondiente al algoritmo Dijkstra, como lo sugiere [77].

DIJKSTRA (Grafo G , nodo_fuente s)

Para $u \in V[G]$ **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 $v \in adyacencia[u]$ **hacer**

si $distancia[v] > distancia[u] + peso(u, v)$ **hacer**

$distancia[v] = distancia[u] + peso(u, v)$

$padre[v] = u$

 adicionar($cola, (v, distancia[v])$)

3.1.2 Algoritmo LAUC adaptado

La gestión del espectro será realizada por el algoritmo del “Ultimo Canal no planificado Disponible” o LAUC, también llamado “Horizon” en su versión adaptada para RSA. Este algoritmo es ampliamente usado debido a su relativa sencillez y capacidad de recursos, por lo que será escogido para comparar y evaluar el desempeño del algoritmo diseñado.

LAUC busca minimizar los huecos en la asignación de espectro seleccionando los slots disponibles más recientes para cada paquete de entrada. Dado el tiempo de llegada t de un paquete de datos de duración L , el scheduler o planificador primero encuentra los slots para los de datos de salida que no tengan planificado nada aún (slots a la salida que no tiene ningún paquete planificado después del instante de llegada del paquete de entrada). Si existe al menos un slot que presente estas características, el planificador selecciona el último slot disponible, por ejemplo, el slot que tiene un menor hueco entre t y el final del paquete de datos justo antes de t , para llevar el paquete de datos que llega. El tiempo sin planificar del slot seleccionado (por ejemplo, el tiempo futuro disponible) es actualizado a $t+L$. A continuación, se presentan los pasos que sigue el algoritmo LAUC en la asignación de canales:

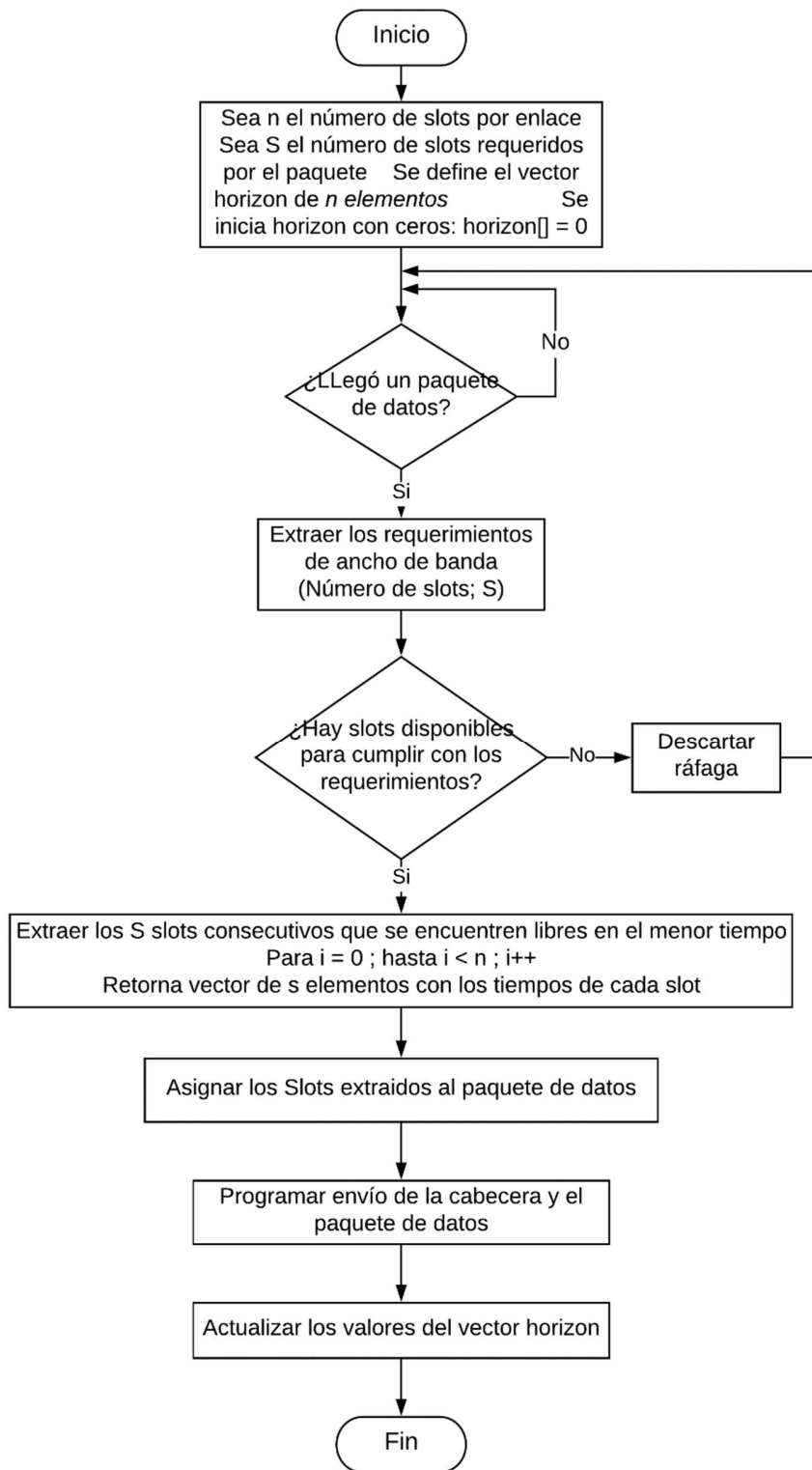


Figura 3.2 - Diagrama de flujo de LAUC adaptado

1. Sea horizon un vector que guarda los tiempos próximos en los que los slots estarán libres.
2. Se Inicializa el vector horizon con ceros.
3. Si llega un paquete de datos, se extraen los slots con el menor tiempo del vector horizon.
4. Se asignan los slots correspondientes a ese tiempo al paquete de datos.
5. Se programa el envío de la cabecera y el paquete de datos.
6. Se actualizan los valores del vector horizon.

A continuación, se muestra el pseudo código correspondiente al algoritmo LAUC adaptado, en el que “*wl*”, corresponde a la frecuencia central a asignar.

```
LAUC (packetSlots, wl)  
  n = numSlots  
  s = packetSlots  
  si no hay slots disponibles hacer  
    descartar paquete  
  de otra forma, hacer  
    si  $wl \geq n - s$  hacer  
       $wl = s - 1$   
    de otra forma, hacer  
       $wl = \text{min} + s$   
  retornar wl
```

En la figura 3,2, se muestra el diagrama de flujo para el algoritmo LAUC adaptado para RSA.

Para el desarrollo de este trabajo de grado, se tendrán en cuenta tres clases de servicio que prestará la red, estos pueden abstraerse como información del tipo: datos, audio y video, donde cada uno cuenta con un tamaño de paquete específico, lo que se ajusta más con los escenarios reales en los que existen paquetes de información de diferente tamaño.

Los paquetes del servicio de video ocuparán un ancho de banda de 37.5 GHz, equivalente a tres slots en la rejilla espectral.

Los paquetes del servicio de audio ocuparán un ancho de banda de 25 GHz, equivalente a dos slots en la rejilla espectral.

Los paquetes del servicio de datos (archivos que no tienen formato de audio o video) ocuparán un ancho de banda de 12,5 GHz, ocupando así un slot de frecuencia en la rejilla espectral.

3.2 Centralización de la Red

Como se mencionó previamente, el nodo número nueve, correspondiente a la ciudad de Pensilvania como lo sugiere [75], será usado para realizar el proceso de centralización de la función de asignación de espectro bajo un algoritmo RSA en la red.

3.2.1 Funcionamiento de la red centralizada

El proceso de centralización de la red, es una de las partes más complejas y demandantes en todo el desarrollo del trabajo de grado, por lo que llevó varias semanas lograr completar esta etapa. Para hacerlo, fue necesario realizar decenas de simulaciones y examinar diversas alternativas para centralizar. Como se verá más adelante, realizar una sola simulación en la herramienta OMNeT++ puede llevar horas, por esto fue necesario hacer uso de varios equipos de cómputo al mismo tiempo, buscando optimizar el proceso.

En los siguientes párrafos, se explicará la forma en la que se realizó el proceso de centralización de la red, cabe decir que existen varias maneras de centralizar los procesos de una red distribuida y la elección de algún método queda a consideración del diseñador de la red, por lo que pueden encontrarse métodos de centralización que difieran al implementado en este trabajo de grado.

Dado que en este trabajo de grado se busca implementar características heredadas de los procesos de centralización y cognitividad, se ha optado por implementar en la red una abstracción del proceso de centralización de una red SDN. En este caso, se realizará una centralización en la que existirán paquetes de control que viajarán de los nodos origen a un nodo central ubicado en Pensilvania y regresarán de vuelta una vez atendidos, proceso que es realizado en una red SDN convencional, con la salvedad de que en este trabajo de grado no se realizará la asignación de recursos en el nodo central, por lo que este proceso servirá como base para el estudio del comportamiento de la red al ser sometida a los retardos y tiempos de procesamiento que existirían en una red SDN.

Como fue explicado con anterioridad, cada paquete creado en la red posee una cabecera de control, además de esta cabecera fue necesario el diseño e implementación de otro paquete para permitir la centralización de la red. Este nuevo paquete ha sido llamado “**Paquete controlador de la centralización**” o PCC de forma abreviada. Los PCC serán creados en el módulo sender de cada nodo cada vez que un nuevo paquete de información salga de la fuente de datos.

En la figura 3,3 se observa el diagrama de flujo correspondiente al proceso de centralización de la red, en él se aprecian las funciones que son realizadas por el nodo origen y el nodo central.

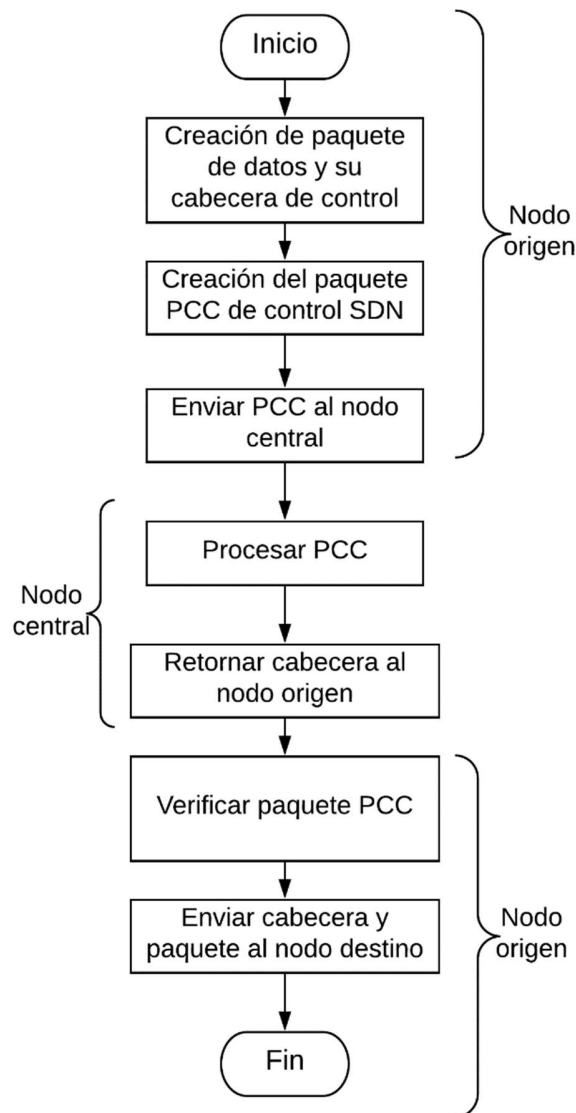


Figura 3.3 - Diagrama de flujo del proceso de centralización de la red

Cuando un PCC es creado, será etiquetado con el número identificador del paquete de datos al que corresponde y será despachado hacia el nodo central de la red o nodo nueve. Una vez el PCC llega al nodo central es atendido y etiquetado indicando que va de regreso al nodo origen donde fue creado, a medida que el PCC avanza por los diferentes nodos de la red, estos verifican el estado del paquete y comprueban si el PCC pertenece o no al nodo en el que se encuentra en ese momento, de ser así el PCC habría llegado al nodo origen, de lo contrario seguirá recorriendo la red hasta alcanzarlo. Cuando el PCC alcanza su nodo de origen, es analizado y entonces la cabecera del paquete es despachada hacia su destino seguida por el paquete de información.

El proceso de centralización está diseñado de tal modo que es así como podría realizarse la centralización de una red convencional, por lo que esto afecta el comportamiento de la red como podrá verse más adelante. La creación, envío, análisis y retorno del PCC desde el nodo origen hasta su regreso agrega un retardo en la operación de la red debido a los tiempos necesarios para cumplir con las etapas mencionadas, además, de un gasto computacional extra.

La cabecera PCC ha sido diseñada de forma tal que contenga la información mínima necesaria para permitir la centralización. Un PCC contiene en sus campos los datos de origen-destino, identificador y etiqueta de dirección que indica si el PCC va hacia el nodo central o, por el contrario, sí regresa de este. Desde el momento en que el PCC es creado se encamina hasta llegar al nodo central. En una red centralizada típica, el paquete de control recibido sería analizado con el fin de extraer la información que contiene y de ese modo asignar los recursos y fijar parámetros en la red de forma tal que el nodo de origen de la información reciba las indicaciones necesarias para el envío de su información, en este trabajo de grado ese proceso no es llevado a cabo, por lo que el PCC recibido en el nodo central es etiquetado indicando que va de regreso hacia el nodo donde fue originado, y entonces es despachado hacia él. Una vez el PCC llega al nodo origen, este realiza la asignación de recursos de la forma habitual y la red continúa con su operación de forma convencional. En otras palabras, el nodo origen asigna espectro, calcula la ruta y realiza las demás tareas necesarias para el envío de su información. En la figura 3,4 puede observarse como está compuesta la cabecera PCC y la explicación de sus campos enseguida.

Lambda	Origen	Destino	Clase de Paquete	Próximo Salto	Número de Saltos	Verificación de Procesamiento
--------	--------	---------	------------------	---------------	------------------	-------------------------------

Figura 3.4 - Composición de la cabecera PCC

- **Lambda:** Frecuencia central asignada al paquete a transmitir.
- **Origen:** Nodo donde fue creado el paquete a transmitir.
- **Destino:** Nodo destino del paquete a transmitir.
- **Clase de paquete:** Indica el tipo de servicio a transmitir (Paquetes de tamaño uno, dos o tres slots de ancho de banda).
- **Próximo salto:** Próximo salto del paquete hasta alcanzar su destino.
- **Número de saltos:** Contador de números de saltos que ha dado el PCC, es útil para estimar el tiempo que le tomará llegar de regreso al origen y entonces realizar la debida comprobación
- **Verificación de procesamiento:** Indica si la cabecera PCC ya ha sido procesada en el nodo central de la red.

3.3 Inclusión del Algoritmo en la Red

Como se mencionó anteriormente, la asignación de espectro se realiza mediante el algoritmo correspondiente dentro del módulo planificador de envío o sender, desde dónde se calculan también los procesos de sincronismo necesarios para el envío de cabeceras y paquetes. Es por esto, que es en este módulo se introducirá el algoritmo diseñado.

Para el diseño del algoritmo fue necesario estudiar la asignación de espectro de los algoritmos convencionales que operan frecuentemente en las redes de este tipo. Se puede decir que la mayoría de los algoritmos hace una búsqueda de la disponibilidad espectral en un solo sentido de la rejilla, moviéndose mayormente desde los slots de menor índice hacia los de mayor o algunos de forma aleatoria.

Diversos modelos para el desarrollo de los algoritmos fueron propuestos a lo largo del trabajo de grado, cada modelo viable fue implementado en la red y posteriormente simulado, para verificar su correcto funcionamiento y eficacia para asignar los recursos, haciendo de esta etapa un proceso lento y complejo que consumió varias semanas y decenas de simulaciones. Cuando una propuesta de algoritmo viable, a la que se denominó como “**algoritmo del péndulo**” fue identificada, se procedió a desarrollar el diagrama de flujo y a evaluar su comportamiento mediante pruebas de escritorio para posteriormente llevarla al simulador y verificar su operación.

El funcionamiento del algoritmo puede resumirse como un movimiento pendular que recorre la rejilla desde sus extremos alternados hacia su centro, en busca de los slots necesarios para atender un paquete. En la figura 3,5 se observa un ejemplo del funcionamiento del algoritmo para atender un paquete que ocupa dos slots de una rejilla que posee 100 GHz de ancho de banda, es decir ocho slots, después se detalla la operación del algoritmo.

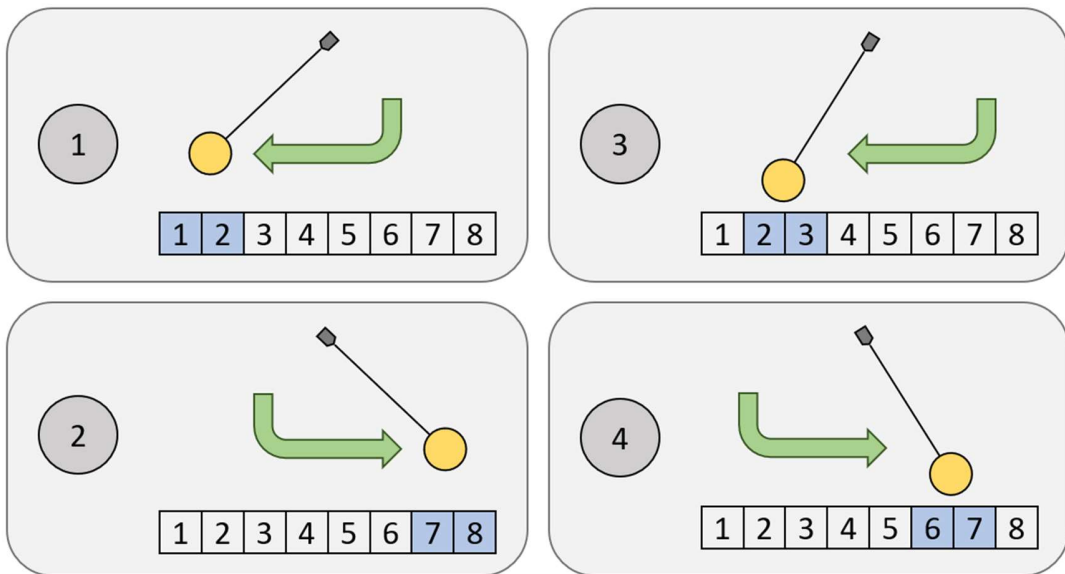


Figura 3.5 - Ejemplo de operación del algoritmo péndulo

1. Cuando un paquete debe ser transmitido, se analiza el tamaño que ocupará en slots, este valor es guardado en la variable n .
2. Dado que la cantidad de slots de la rejilla puede ser ocho o dieciséis, este valor se almacena en S . Una variable T es iniciada en cero, esta variable será utilizada para recorrer la rejilla en forma pendular.
3. Sabiendo el ancho de banda demandado por el paquete a enviar, se verifica la disponibilidad de los $T+1$ hasta $T+n$ slots de la rejilla, siendo este el primer movimiento del péndulo. De hallar disponibles esos slots, el paquete es enviado, finalizando el algoritmo.

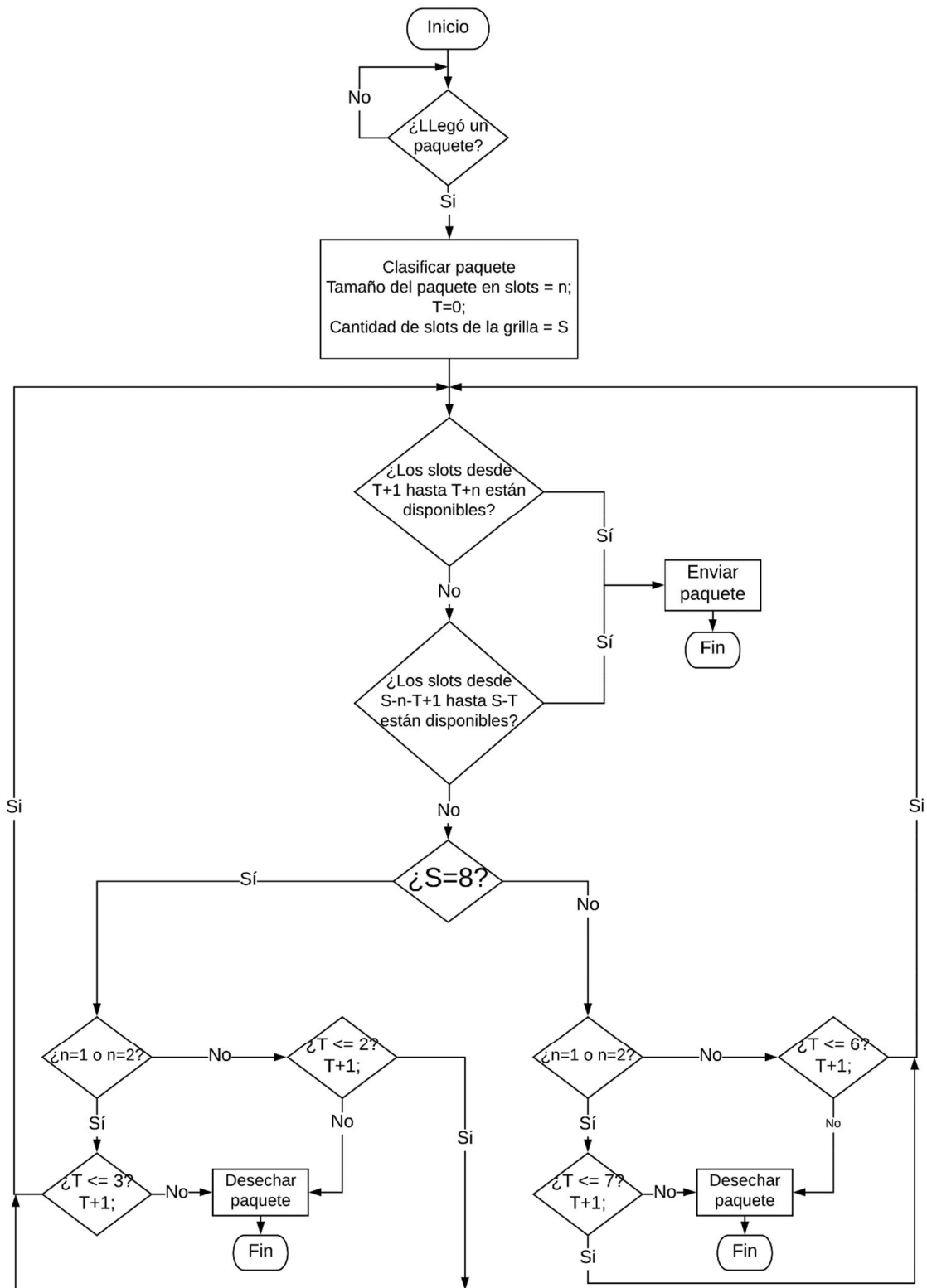


Figura 3.6 - Diagrama de flujo del algoritmo péndulo

4. Si en el paso anterior no se encuentran recursos, se verifica la disponibilidad de los últimos $S-n-T+1$ hasta $S-T$ slots de la rejilla. Si estos slots están disponibles, se enviará el paquete por ahí, finalizando el algoritmo. En este punto, el algoritmo habrá verificado la disponibilidad de los primeros y últimos slots, en un primer ciclo $T=0$ completando su primer movimiento pendular. Si el algoritmo no ha encontrado recursos para enviar el paquete en el primer ciclo, se procede al siguiente paso.
5. Si la rejilla tiene un tamaño de ocho slots se seguirá un camino y de no ser así, se seguirá otro.
6. Según el tamaño del paquete a transmitir se sigue por un camino de acción en el diagrama de flujo u otro según corresponda.

Si los paquetes ocupan uno o dos slots, se verificará que la variable T sea menor o igual a tres si $S = 8$ o que T sea menor o igual a siete para $S = 16$ para continuar al siguiente paso, de ser mayor a tres o siete según el caso, significaría que el algoritmo solaparía los movimientos de izquierda a derecha y de derecha a izquierda, y llevaría a que se exploren zonas de la rejilla revisadas previamente. Si T es mayor al condicional es porque no existen slots disponibles para enviar el paquete, por lo que este es descartado.

7. Se verifica el valor de la variable T y posteriormente se incrementa en una unidad. A continuación, el algoritmo regresa al paso tres, con el fin de explorar los siguientes slots que correspondan según el valor de T .
8. El algoritmo finaliza con el envío del paquete o con su descarte por falta de recursos.

A continuación, se muestra el pseudo código correspondiente al algoritmo "Pendulo", en el que " wl ", corresponde a la frecuencia central a asignar.

PENDULO (*packetSlots*, *slotsDisponibles[numSlots]*)

$wl = 0$

$n = numSlots$

$s = packetSlots$

si no hay slots disponibles hacer

descartar paquete

de otra forma, hacer

si $s = 1$ hacer

```
para  $i = 0; i < n/2; i++$  hacer  
  si  $slotsDisponibles[i] = 0$  hacer  
     $slotsDisponibles[i] = 1$   
     $wl = i$   
    retornar  $wl$   
  de otra forma, si  $slotsDisponibles[n - 1 - i] = 0$  hacer  
     $slotsDisponibles[n - 1 - i] = 1$   
     $wl = n - 1 - i$   
    retornar  $wl$   
si  $s = 2$  hacer  
  para  $i = 0; i < n/2; i++$  hacer  
    si  $slotsDisponibles[i] = 0$  y  $slotsDisponibles[i + 1] = 0$  hacer  
       $slotsDisponibles[i] = 1$   
       $slotsDisponibles[i + 1] = 1$   
       $wl = i + 1$   
      retornar  $wl$   
    de otra forma, si  $slotsDisponibles[n - 1 - i] = 0$  y  
       $slotsDisponibles[n - 2 - i] = 0$  hacer  
         $slotsDisponibles[n - 1 - i] = 1$   
         $slotsDisponibles[n - 2 - i] = 1$   
         $wl = n - 2 - i$   
        retornar  $wl$   
si  $s = 3$  hacer  
  para  $i = 0; i < n/2; i++$  hacer  
    si  $slotsDisponibles[i] = 0$  y  $slotsDisponibles[i + 1] = 0$  y  
       $slotsDisponibles[i + 2]$  hacer  
         $slotsDisponibles[i] = 1$   
         $slotsDisponibles[i + 1] = 1$   
         $slotsDisponibles[i + 2] = 1$   
         $wl = i + 2$   
        retornar  $wl$   
    de otra forma, si  $slotsDisponibles[n - 1 - i] = 0$  y  
       $slotsDisponibles[n - 2 - i] = 0$  y  
         $slotsDisponibles[n - 3 - i]$  hacer  
           $slotsDisponibles[n - 1 - i] = 1$   
           $slotsDisponibles[n - 2 - i] = 1$   
           $slotsDisponibles[n - 3 - i] = 1$   
           $wl = n - 3 - i$   
          retornar  $wl$ 
```

Seguir los pasos anteriores, garantiza que el algoritmo recorra en forma de péndulo el espacio de la rejilla espectral buscando en todas las posiciones posibles los slots necesarios para transmitir los paquetes de información. En la figura 3,6 se observa el diagrama de flujo del algoritmo péndulo para una red con ocho o dieciséis slots de ancho de banda y paquetes de tamaño uno, dos o tres slots. El algoritmo puede ser adaptado para operar sobre redes con una disponibilidad espectral diferente y paquetes de distinto tamaño, permitiendo así que sea altamente escalable y adaptable a las redes del futuro.

Dada que los paquetes mantienen su tamaño una vez creados y la conexión óptica entre nodos existe únicamente por el instante de tiempo en el que el paquete es transmitido, se puede decir entonces que el algoritmo realiza una asignación de espectro fija en el tiempo, pues no varía el ancho espectral asignado a una conexión, como se vio en la sección 1.1.4.

Como se mencionó en el capítulo anterior, para poder catalogar a un algoritmo como cognitivo, este deberá poder cumplir mínimamente con las tareas de observación y acción comprendidas dentro del ciclo cognitivo. El algoritmo de péndulo realiza asignación de espectro cumpliendo con las tareas de observación, análisis y acción, puesto que recorre la grilla espectral buscando espacios dónde ubicar el paquete que solicita ser enviado, tarea que puede entenderse como la observación. A medida que el algoritmo recorre la grilla, estudia los slots que observa y determina su estado (ocupados o disponibles), lo que puede entenderse como la fase de análisis del ciclo cognitivo. Finalmente, cuando un grupo de slots disponibles es encontrado el algoritmo los escoge para realizar el envío de la información por ahí, lo que puede entenderse como la fase de acción del ciclo.

3.3.1 Metaheurística implementada

Para la solución del problema RSA se optó por aplicar la metaheurística de relajación, en la cual, un problema mayor es descompuesto en problemas menores de más fácil solución. Para esto, el problema fue abordado de forma secuencial, es decir el problema fue llevado a la forma R+SA. Donde la parte de enrutamiento fue resuelta por el algoritmo Dijkstra como se explicó previamente y la parte SA fue resuelta por el algoritmo péndulo. El algoritmo propuesto fue desarrollado haciendo uso de su propia metaheurística como se explica en seguida.

Para el diseño del algoritmo péndulo se exploraron diversas metaheurísticas que podrían ser de ayuda para solucionar el problema, simplificándolo y haciéndolo más manejable. Dado que lo que se busca es que el algoritmo explore un conjunto finito de elementos que en este caso es la grilla espectral, en donde

uno o varios de estos elementos son óptimos para la solución del problema de asignación de espectro, fue la metaheurística de los algoritmos voraces, la que, desde nuestro punto de vista, mejor se adecuó al problema.

Un algoritmo voraz recibe un conjunto C finito de elementos, y devuelve un conjunto de S elementos después de aplicar criterios de selección, de modo que S está contenido en C y cumple con las restricciones del problema inicial. Si el conjunto S resuelve el problema de optimización, se dirá entonces que S es una solución óptima. En caso de que S no sea una solución, se descarta y se procede a iterar nuevamente el algoritmo, donde se verificará un nuevo conjunto S en búsqueda de la solución [55].

El objetivo de estos algoritmos es el de encontrar una solución óptima general a un problema mientras realiza búsquedas de soluciones locales. En el caso del problema SA, el problema general sería la asignación de espectro a todos o la mayoría de los paquetes generados en la red, mientras que las soluciones locales serían la asignación de espectro a un paquete a la vez. Los algoritmos voraces son ampliamente utilizados para la solución de problemas de optimización, debido a que no son altamente complejos y permiten una verificación relativamente sencilla de su funcionamiento.

3.4 Escenarios de Simulación

Para la comparación entre el algoritmo clásico LAUC y el algoritmo propuesto anteriormente, se han propuesto varios casos o escenarios de simulación, los cuales servirán para evaluar el desempeño de nuestro algoritmo bajo diversas condiciones de operación.

Básicamente, un escenario de simulación, es un conjunto de parámetros que determinan el comportamiento y operación de un ambiente, en este caso, los parámetros modelarán diversas condiciones en la red NSFNet modelada en el OMNeT++. El análisis de los resultados obtenidos se hará mediante el estudio de los valores de probabilidad de bloqueo (Pb) y el retardo de extremo a extremo.

$$Pb = \frac{\text{Paquetes perdidos}}{\text{Paquetes totales}} \quad \text{Ecuación (3.1)}$$

La probabilidad de bloqueo, como lo muestra la ecuación 3,1 es la relación entre la cantidad de paquetes generados por las fuentes de datos y la cantidad de paquetes que se perdieron sin importar el motivo y que nunca alcanzaron su destino. Por otra parte, el retardo de extremo a extremo, es un valor que mide el

tiempo desde que un paquete es creado en la fuente, hasta que es entregado a su destinatario.

Tabla 3.1 - Escenarios de simulación propuestos

Escenarios de simulación							
Topología de red NSFNet – Red centralizada							
Casos de simulación							
Caso 1				Caso 2			
Red NSFNet RSA centralizada, implementando el algoritmo LAUC				Red NSFNet RSA centralizada, implementando el algoritmo propuesto			
Sub casos de simulación							
Sub caso 1		Sub caso 2		Sub caso 1		Sub caso 2	
8 slots disponibles		16 slots disponibles		8 slots disponibles		16 slots disponibles	
Casos de tercer nivel		Casos de tercer nivel		Casos de tercer nivel		Casos de tercer nivel	
Caso a	Caso b	Caso a	Caso b	Caso a	Caso b	Caso a	Caso b
1,25 Gbps	2,5 Gbps	1,25 Gbps	2,5 Gbps	1,25 Gbps	2,5 Gbps	1,25 Gbps	2,5 Gbps
Los casos de estudio se simularán para cada una de las cargas de tráfico especificadas más adelante.							

Las tablas 3,1 y 3,2 muestran los escenarios de simulación planteados. Como se observa, la red NSFNet distribuida⁴ será también escenario de estudio, esto para comprobar el funcionamiento del algoritmo fuera de un escenario centralizado. Estos casos han sido escogidos de forma tal que la red sea sometida a varios casos de operación y poder así estudiar el comportamiento del algoritmo propuesto. Parámetros como la cantidad de slots de frecuencia disponibles para transmitir, velocidad de transmisión de datos y la cantidad de tráfico han sido modificados y combinados, buscando hacer más completo el estudio del trabajo de grado.

La cantidad de tráfico será variada para cada uno de los casos de tercer nivel, pasando por tráfico muy bajo, bajo, medio, alto y muy alto. Para ello, la variable *SentlaTime*, encargada de producir paquetes de datos de forma aleatoria (ver anexo A) siguiendo una distribución exponencial, será variada en los valores de $4.6 \mu s$, $3.95 \mu s$, $3.3 \mu s$, $2.65 \mu s$ y $2 \mu s$ que corresponden a los tipos de tráfico mencionados. Estos valores han sido escogidos puesto que a lo largo del desarrollo del trabajo de grado se observa que en el valor de $4.6 \mu s$, la red se

⁴ Una red distribuida se entiende como una red que opera sin la necesidad de un nodo central y en la que cada nodo realiza sus operaciones de forma convencional.

comporta bastante bien, presentando bajo retardo y poca Pb, condiciones que indican que la red posee poco volumen de tráfico y, por el contrario, para la carga de 2 μ s la red presenta alto grado de congestión, lo que indica un gran volumen de tráfico y saturación de red, por lo que los valores intermedios entre estos puntos críticos son ideales para el estudio de la misma.

Tabla 3.2 - Escenarios de simulación propuestos

Escenarios de simulación							
Topología de red NSFNet – Red Distribuida							
Casos de simulación							
Caso 1				Caso 2			
Red NSFNet RSA distribuida, implementando el algoritmo LAUC				Red NSFNet RSA distribuida, implementando el algoritmo propuesto			
Sub casos de simulación							
Sub caso 1		Sub caso 2		Sub caso 1		Sub caso 2	
8 slots disponibles		16 slots disponibles		8 slots disponibles		16 slots disponibles	
Casos de tercer nivel		Casos de tercer nivel		Casos de tercer nivel		Casos de tercer nivel	
Caso a	Caso b	Caso a	Caso b	Caso a	Caso b	Caso a	Caso b
1,25 Gbps	2,5 Gbps	1,25 Gbps	2,5 Gbps	1,25 Gbps	2,5 Gbps	1,25 Gbps	2,5 Gbps
Los casos de estudio se simularán para cada una de las cargas de tráfico especificadas anteriormente.							

A continuación, se detallan los casos de simulación propuestos.

3.4.1 Caso de simulación 1

El algoritmo LAUC adaptado a RSA será el encargado de la gestión de espectro dentro de la red, esto servirá como punto de partida para comparar el algoritmo propuesto.

3.4.2 Caso de simulación 2

Aquí, la red operará haciendo uso del algoritmo péndulo diseñado. A partir del caso uno y dos, se desprenden los siguientes sub casos de simulación.

3.4.3 Sub casos de simulación

- **Sub caso 1**
Cada enlace de la red, dispondrá de ocho (8) slots de frecuencia para enviar información, lo que espectralmente corresponde a un ancho de banda de 100 GHz.
- **Sub caso 2**
Los enlaces dispondrán de dieciséis (16) slots para enviar información, lo que da lugar a una disponibilidad de 200 GHz de ancho de banda.

3.4.4 Casos de tercer nivel

- **Caso a**
La velocidad del enlace de datos será de 1,25 GHz.
- **Caso b**
La velocidad del enlace de datos será de 2,5 GHz.

3.4.5 Configuración de los parámetros de simulación

La tabla 3,3 contiene los valores fijados para la realización de los diferentes escenarios de simulación.

Tabla 3.3 - Parámetros de simulación

Parámetro	Variable	Valores
Cantidad de espectro	---	100 GHz
		200 GHz
Número de slots	numSlots	8
		16
Velocidad de transmisión	dataRate	1,25 Gbps
		2,5 Gbps
Longitud del paquete	PacketLenght	Trama E4 (2048 Bytes)
Tiempo de procesamiento de la cabecera	OPSProcessingDelay	10 μ s
Tiempo de procesamiento del conmutador óptico	OXCDelay	10 μ s
Tiempo de guarda	guardTime	1 ns
Tiempo de offset	minOffset	20 μ s
	maxOffset	50 μ s

Para la ejecución de las simulaciones se dispuso de varios equipos de cómputo, con el fin de agilizar el proceso, pues la simulación es un proceso lento, en el cual, por ejemplo, 0.2 segundos de simulación equivalen a cinco horas de ejecución del programa OMNeT++, dependiendo del equipo de cómputo usado, por lo que para trabajos futuros se aconseja el uso de equipos de buena potencia de cálculo. Las características de los equipos usados se detallan en la tabla 3,4.

Tabla 3.4 - Descripción de los equipos de cómputo usados

Equipo	Procesador	Memoria RAM	Disco duro	Sistema operativo	Cantidad
Dell inspiron 620s	Intel Core i5	4 GB	1000 GB	Windows 7 X64 bits	2
Lenovo G410	Intel Core i5	4 GB	500 GB	Windows 10 X64 bits	1
HP	AMD A10	16 GB	1000 GB	Windows 10 X64 bits	1

3.5 Ejecución de las Simulaciones

Después de fijados los valores de los parámetros de simulación, se procede a la ejecución de las simulaciones. Dada la diversidad de los casos y sub casos, se realizaron en total ochenta simulaciones que arrojaron los resultados que se estudian en el siguiente capítulo.

El proceso de simulación es muy lento debido a las características de la simulación, llevando horas cada caso, aunque existen simulaciones que se ejecutan con mayor rapidez, también existen otras que demandan mayor cantidad de tiempo, motivo el que se optó a usar varios equipos al mismo tiempo.

Capítulo 4

COMPARACIÓN, ANÁLISIS, RESULTADOS Y CONCLUSIONES

En este capítulo se estudiarán los resultados obtenidos en la ejecución de las simulaciones (ver anexo B), también se hablará de las conclusiones encontradas al analizar los resultados y finalmente se plantearán algunos posibles trabajos futuros, extendiendo así esta línea de investigación.

4.1 Análisis de los Resultados, Red distribuida.

A continuación, se presentan los resultados obtenidos y el análisis respectivo de las pruebas de simulación de las redes NSFNeT RSA distribuida con el algoritmo LAUC adaptado y el algoritmo péndulo propuesto. Los resultados obtenidos en OMNeT++ fueron llevado a la herramienta Origin Pro 8 con el fin de graficar.

4.1.1 Caso red distribuida con ocho slots a 1.25 Gbps y 2.5 Gbps

Siguiendo los parámetros de la tabla 4,1 se ajustan las simulaciones en la herramienta OMNeT++. Cada gráfica mostrada a continuación, comprende diez casos de simulación, cinco para modelar el comportamiento de la red con el algoritmo LAUC adaptado y otros cinco para ver la operación con el algoritmo propuesto. Cada una de las simulaciones representa un valor de tráfico diferente cómo se mencionó en el capítulo anterior. Los valores de tráfico han sido normalizados dentro de todo el conjunto de valores de cada caso, por lo que varían desde cero hasta uno, medido en erlangs.

Tabla 4.1 - Parámetros de simulación, casos 1 y 2, subcaso 1

Parámetro	NSFNet con LAUC	NSFNet con péndulo
Algoritmo de enrutamiento	Dijkstra	Dijkstra
Número de slots	8	8
Tiempo de simulación	80 ms	80 ms
Tas de bits	1.25 Gbps – 2.5 Gbps	1.25 Gbps – 2.5 Gbps
Algoritmo de asignación de espectro	LAUC adaptado	Péndulo

La figura 4,1 muestra la probabilidad de bloqueo vs el tráfico para la red NSFNet con ambos algoritmos y operando a una velocidad de 1,25 Gbps y con un ancho de grilla de ocho slots.

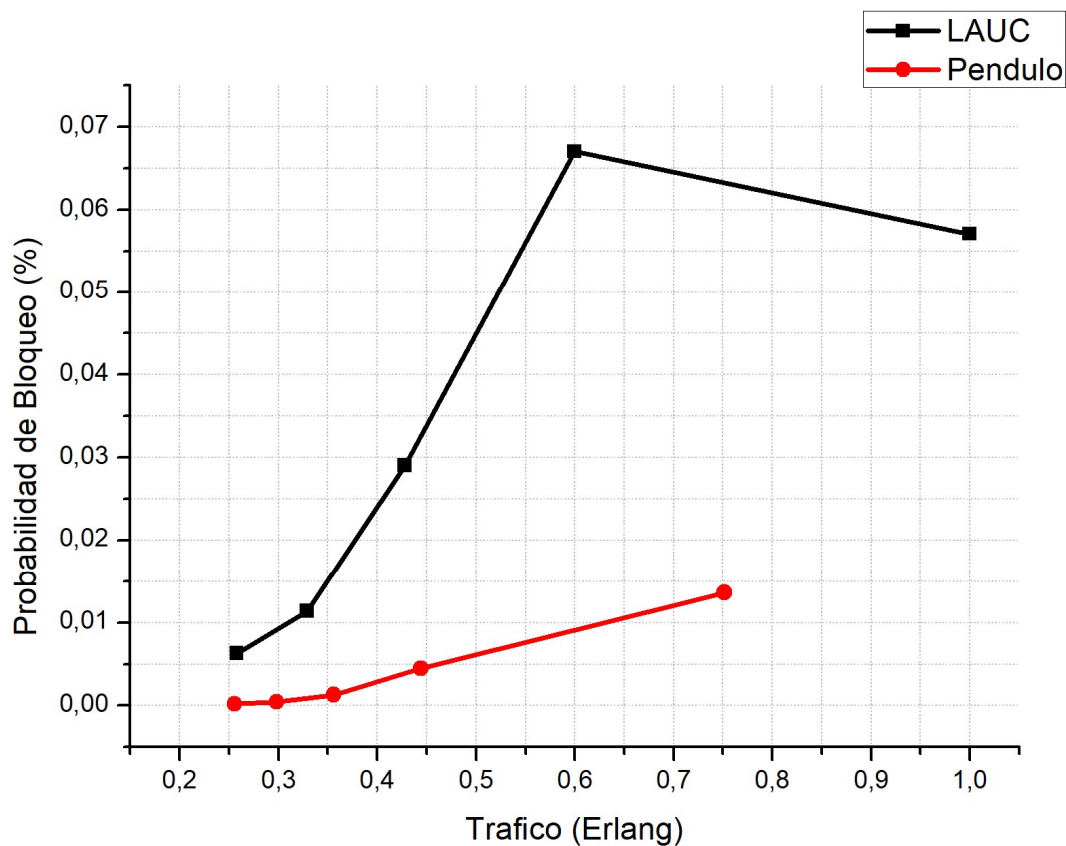


Figura 4.1a - P_b con 8 slots y velocidad de 1.25 Gbps

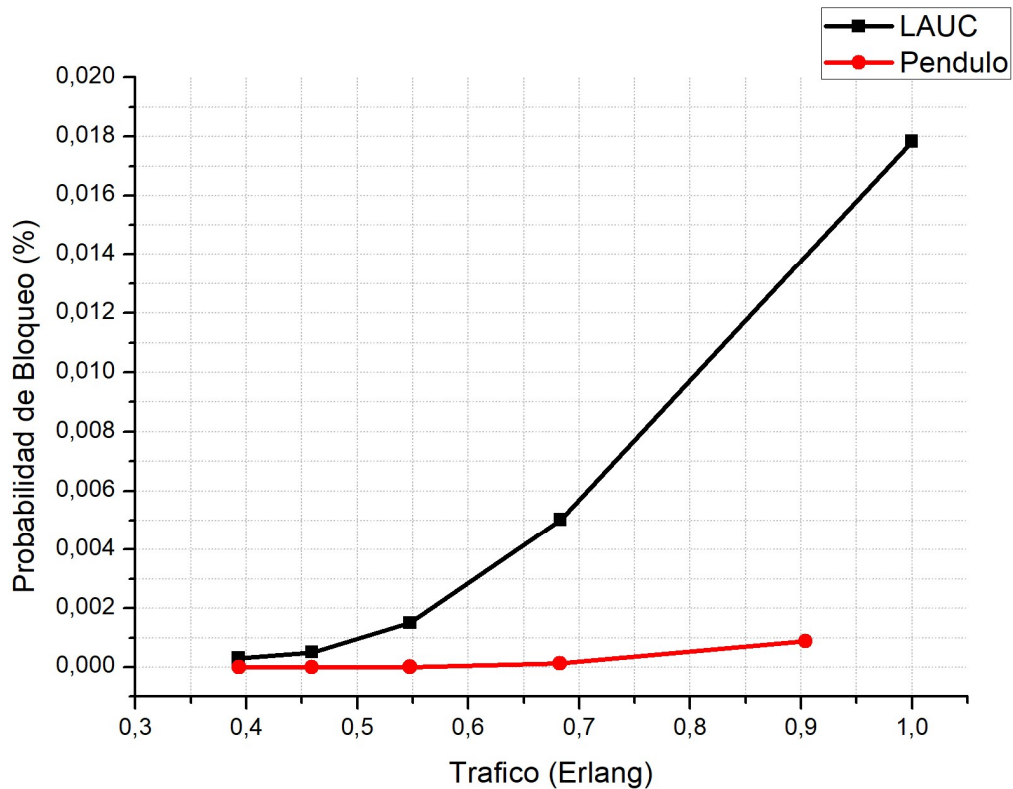


Figura 4.2b - Pb con 8 slots y velocidad de 2.5 Gbps

Cada punto de las líneas graficadas representa una cantidad de tráfico, desde muy bajo hasta muy alto. La no co-linealidad vertical entre gráficas se debe a que la generación de tráfico es de forma aleatoria, como se explicó anteriormente

En la figura anterior caso a, se puede ver como el comportamiento de la red que opera con el algoritmo péndulo, señalada en rojo, es mejor que la que opera con LAUC en negro. Se observa que para cantidades intermedias de tráfico (0.4 – 0.7 erlangs) el comportamiento del algoritmo péndulo es mejor que el LAUC, comportándose de manera más estable y sin presentar pendientes elevadas que lo conduzcan a mayores probabilidades de bloqueo, como si lo hace el LAUC. Mientras que la probabilidad de bloqueo máxima del péndulo es de alrededor de 0.015%, la Pb del LAUC llega hasta cerca de 0.067%. Este resultado muestra que el algoritmo propuesto en este trabajo de grado y bajo estas condiciones de red es mucho más eficiente, pues realiza un barrido de la grilla de forma tal que puede asignar recursos con una mayor probabilidad que su contraparte, por lo que evita que los paquetes sean desechados por falta de recursos.

En el caso b, la velocidad de transmisión de datos es de 2.5 Gbps, este aumento en la velocidad influye notablemente en el comportamiento de la red, pues los slots ocupados en transmitir información serán liberados más rápidamente que a velocidades inferiores, dando espacio para el envío de más paquetes de datos. De nuevo se observa que el comportamiento del algoritmo péndulo es mejor que el LAUC. El algoritmo péndulo es capaz de mantener una tasa de Pb de cero por ciento para valores inferiores de 0.5 erlangs y aumenta solo hasta cerca del 0.005% para tráfico muy alto, mientras que el algoritmo LAUC presenta valores mucho mayores, alcanzando pérdidas de hasta 0.018% como lo indica la gráfica.

4.1.2 Caso red distribuida con dieciséis slots a 1.25 Gbps y 2.5 Gbps

Para estos escenarios de simulación, se dispone de dieciséis slots para asignar, por lo que las capacidades de la red se ven incrementadas, lo que se refleja en la probabilidad de bloqueo. La tabla 4,2 muestra los parámetros fijados para este caso.

Tabla 4.2 - Parámetros de simulación, casos 1 y 2, subcaso 2

Parámetro	NSFNet con LAUC	NSFNet con péndulo
Algoritmo de enrutamiento	Dijkstra	Dijkstra
Número de slots	16	16
Tiempo de simulación	80 ms	80 ms
Tas de bits	1.25 Gbps – 2.5 Gbps	1.25 Gbps – 2.5 Gbps
Algoritmo de asignación de espectro	LAUC adaptado	Péndulo

La figura 4,2 muestra la probabilidad de bloqueo vs el tráfico para la red NSFNet con ambos algoritmos y operando a una velocidad de 1,25 Gbps y con un ancho de grilla de dieciséis slots.

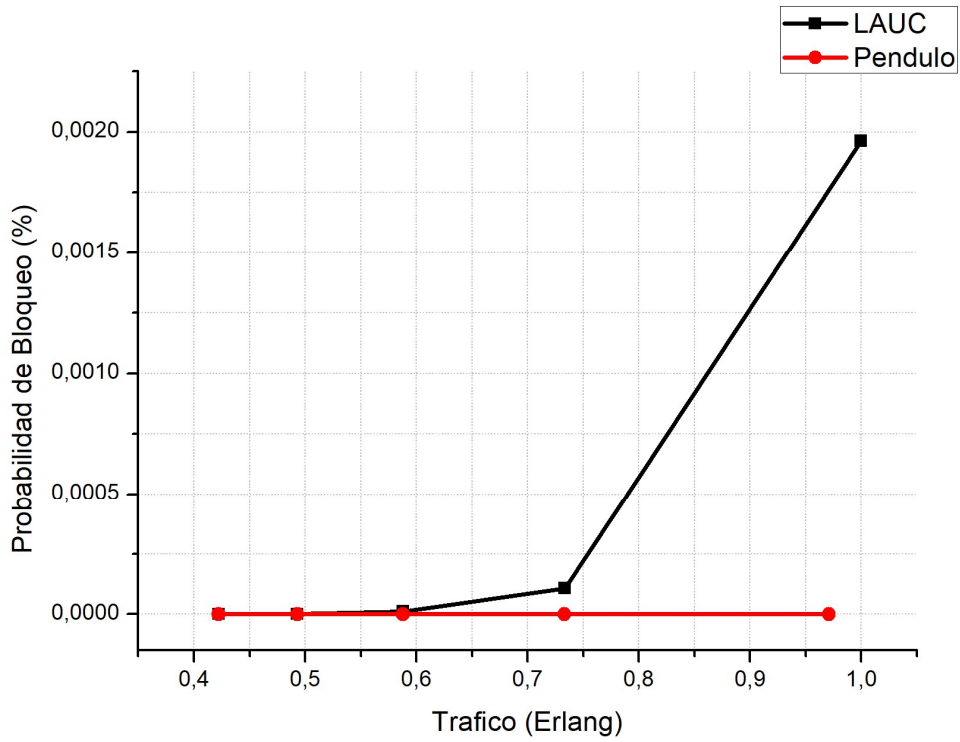


Figura 4.3a - P_b con 16 slots y velocidad de 1.25 Gbps

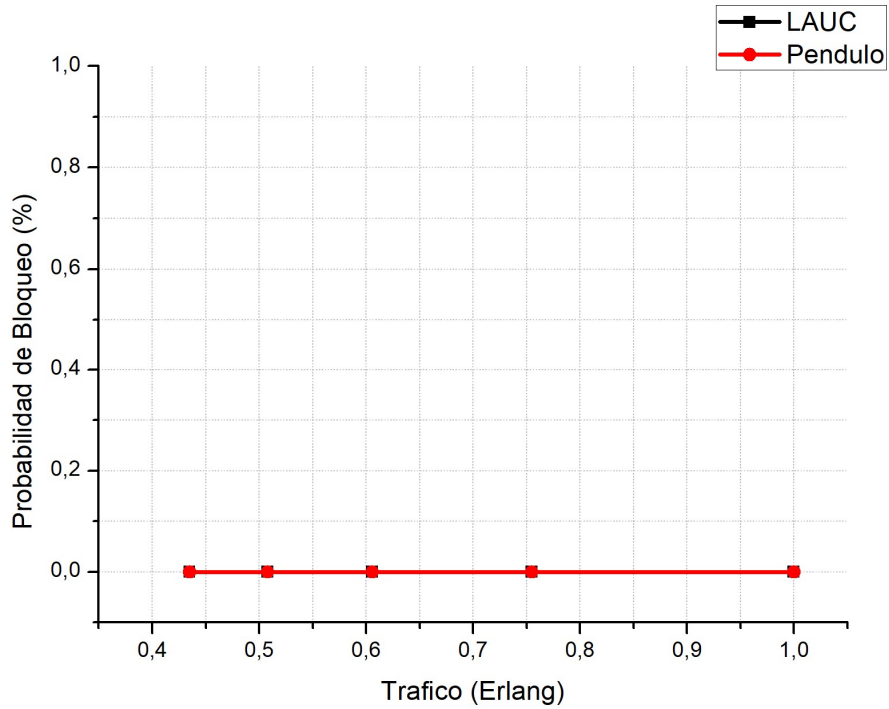


Figura 4.4b - P_b con 16 slots y velocidad de 2.5 Gbps

El aumento de la cantidad de slots genera una clara mejora en la operación de la red, indistintamente del tipo de algoritmo implementado. Para el caso a, el algoritmo péndulo presenta un comportamiento ideal en el que para cualquier carga de tráfico mantiene una probabilidad de bloqueo del cero por ciento, mientras que el algoritmo LAUC, después de 0.6 erlangs empieza a presentar una P_b que está cerca del 0.002%. Aunque este valor de P_b es bastante bajo y cercano a cero, se debe tener en consideración que una red de datos está en capacidad de mover millones de paquetes mientras opera, y valores bajos como 0.002% pueden llegar a tener gran significancia a grandes volúmenes de datos y/o tiempo de operación.

El aumento de la cantidad de slots disponibles y la tasa de velocidad de 2.5 Gbps conducen al escenario b, en el cual el comportamiento de ambos algoritmos es ideal, puesto que disponen de un gran ancho de banda y de una rápida liberación de los recursos, permitiendo que ningún paquete sea desechado.

4.1.3 Caso red distribuida con ocho slots a 1.25 Gbps y 2.5 Gbps, estudio del retardo de extremo a extremo

El otro parámetro analizado para estudiar el comportamiento de la red es el retardo de extremo a extremo que fue explicado previamente. La tabla 4,1 muestra los valores fijados en este caso y la figura 4,3 muestra las gráficas obtenidas como resultados.

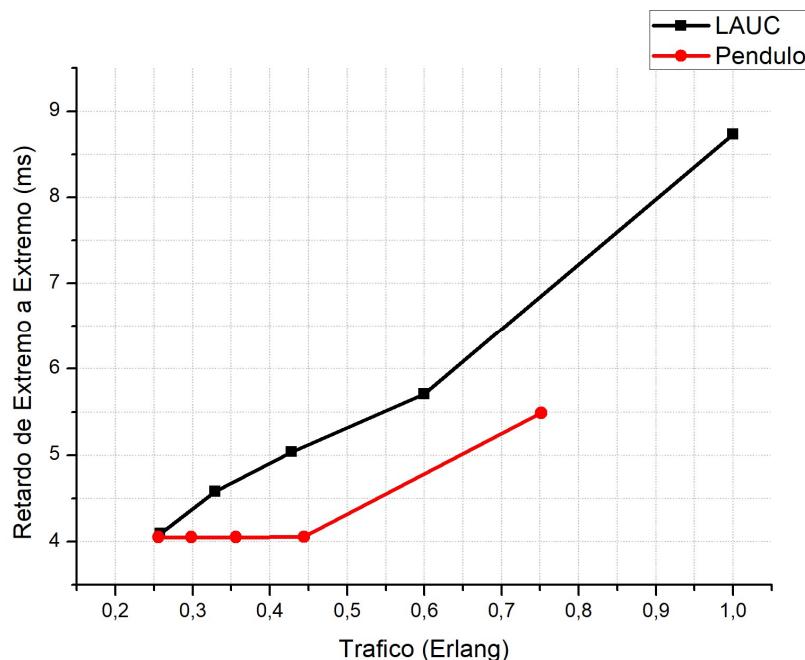


Figura 4.5a - Retardo de extremo a extremo, 8 slots y velocidad de 1.25 Gbps

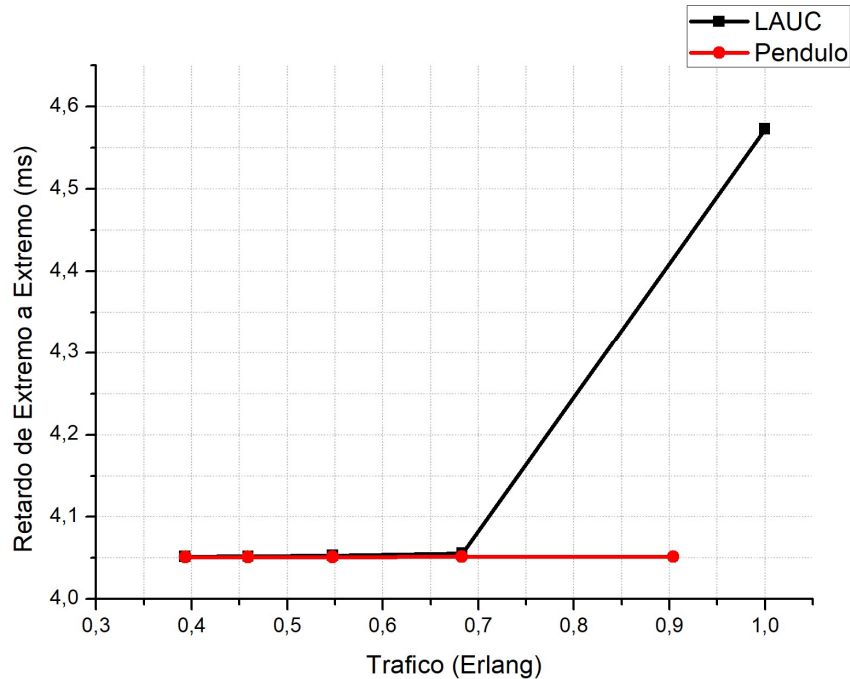


Figura 4.6b - Retardo de extremo a extremo, 8 slots y velocidad de 2.5 Gbps

En el caso a, donde la velocidad de datos es de 1.25 Gbps, se observa que ambas gráficas parten de un valor mínimo de retardo que está dado por las características inherentes a toda red de datos, como, por ejemplo, el tiempo que le toma a un paquete recorrer los enlaces entre nodos y los tiempos de procesamiento y establecimiento de los dispositivos de red.

Puede verse que el retardo de los paquetes es menor cuando se usa el algoritmo péndulo, manteniéndose en el valor mínimo posible hasta cerca de los 4.5 erlangs de tráfico y alcanzando los 5.5 ms de retardo para una cantidad elevada de tráfico. El mejor desempeño del algoritmo péndulo se debe a que es capaz de asignar slots de forma más eficiente que el algoritmo LAUC, por lo que evita la congestión en las colas de paquetes, es decir que, los paquetes pasan menor tiempo en espera de recursos y son atendidos más rápidamente.

En el caso b, la velocidad de datos es de 2.5 Gbps, dado que como fue explicado previamente, los recursos son liberados más rápidamente, los paquetes no deben pasar mayor tiempo en cola esperando por recursos y en lugar de eso, son atendidos de forma inmediata, por lo que no existe un retardo adicional al retardo mínimo de red. Es por esto, que en este caso el retardo generado por el algoritmo péndulo es el mínimo posible, mientras que el algoritmo LAUC llega a presentar un retardo de cerca de 4.55 ms, aunque, para valores medios de tráfico, el comportamiento de ambos algoritmos es prácticamente el mismo.

4.1.4 Caso red distribuida con dieciséis slots a 1.25 Gbps y 2.5 Gbps, estudio del retardo de extremo a extremo

En estos casos de estudio, la cantidad de slots en la grilla ha sido elevada hasta dieciséis, dando a la red más recursos para transmitir. La tabla 4,2 muestra los parámetros fijados en estos escenarios.

La figura 4,4 muestra el retardo de extremo a extremo vs el tráfico para la red NSFNet con ambos algoritmos y operando a una velocidad de 1,25 Gbps y con un ancho de grilla de dieciséis slots.

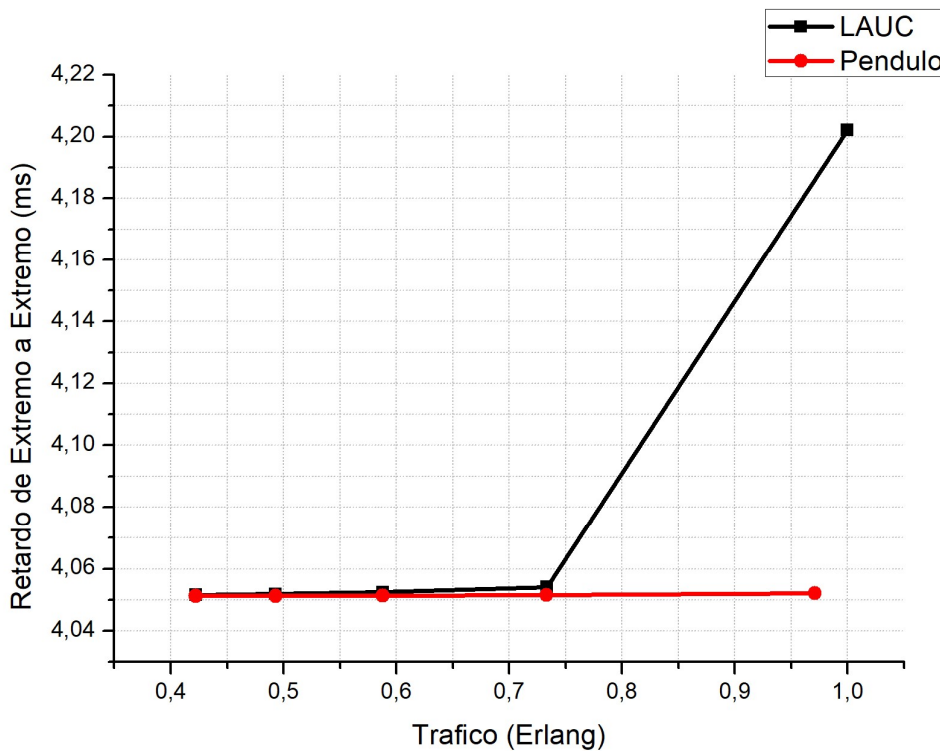


Figura 4.7a - Retardo de extremo a extremo, 16 slots y velocidad de 1.25 Gbps

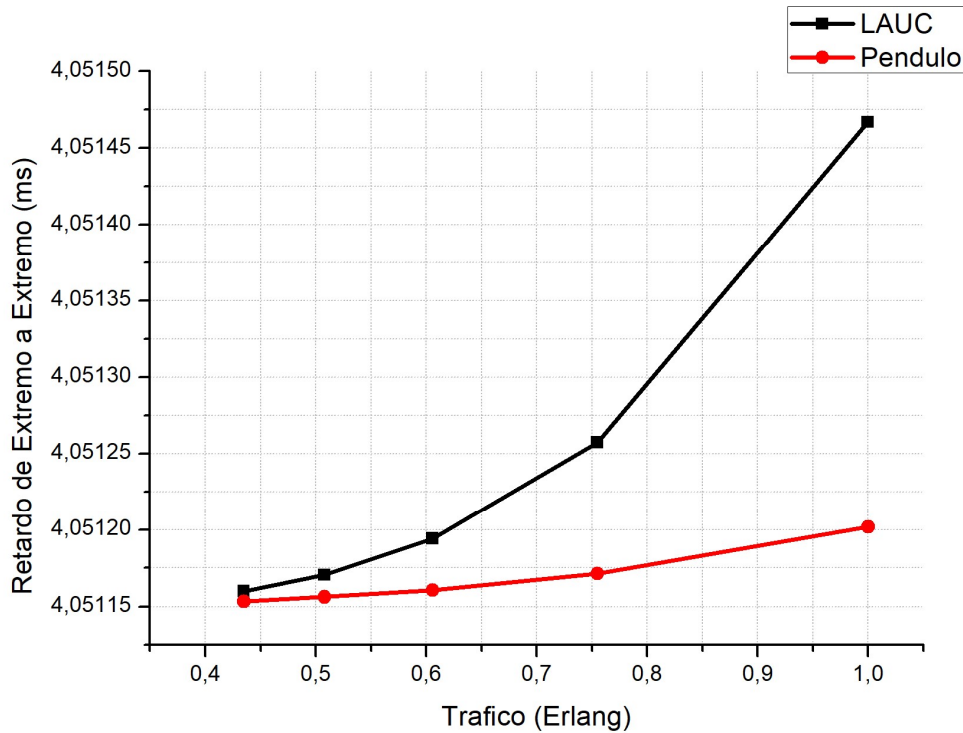


Figura 4.8b - Retardo de extremo a extremo, 16 slots y velocidad de 2.5 Gbps

En el caso a, el algoritmo péndulo no agrega ningún retardo adicional, manteniendo el retardo de extremo a extremo en un valor cercano al mínimo posible, esto es debido a que la grilla cuenta con la cantidad de slots suficientes para garantizar atender a los paquetes entrantes. El algoritmo LAUC se comporta bastante bien para cantidades intermedias de tráfico, pero para valores elevados y a pesar que cuenta con suficientes slots, no es capaz de realizar una gestión eficiente de los recursos por lo que algunos paquetes son puestos en cola, llevando a retardos de cerca de 4.2 ms, mientras que el retardo del algoritmo péndulo se mantiene cerca de 4.05 ms.

Para una velocidad de 2.5 Gbps, la red se comporta aún mejor que en el caso anterior, gracias a la rápida liberación de los recursos, y el retardo se mantiene en valores muy bajos, para ambos algoritmos, pero con un mejor desempeño para el algoritmo péndulo que alcanza un retardo máximo cercano a 4.0512 ms mientras que el LAUC llega hasta cerca de 4.0515 ms.

4.2 Análisis de los Resultados, Red Centralizada.

En esta sección se presentan los resultados obtenidos y el análisis respectivo de las pruebas de simulación de las redes NSFNet RSA centralizada, que opera con el algoritmo LAUC adaptado y el algoritmo péndulo propuesto.

4.2.1 Caso red centralizada con ocho slots a 1.25 Gbps y 2.5 Gbps

A continuación, se muestra la tabla 4,3 en la que se muestran los parámetros fijados para simular la red centralizada y la figura 4,5 muestra la probabilidad de bloqueo vs el tráfico para la red NSFNet centralizada con ambos algoritmos y operando a una velocidad de 1.25 Gbps y con un ancho de grilla de ocho slots.

Tabla 4.3 - Parámetros de simulación, casos 1 y 2, subcaso 1, red centralizada

Parámetro	NSFNet con LAUC	NSFNet con péndulo
Algoritmo de enrutamiento	Dijkstra	Dijkstra
Número de slots	8	8
Tiempo de simulación	80 ms	80 ms
Tas de bits	1.25 Gbps – 2.5 Gbps	1.25 Gbps – 2.5 Gbps
Algoritmo de asignación de espectro	LAUC adaptado	Péndulo

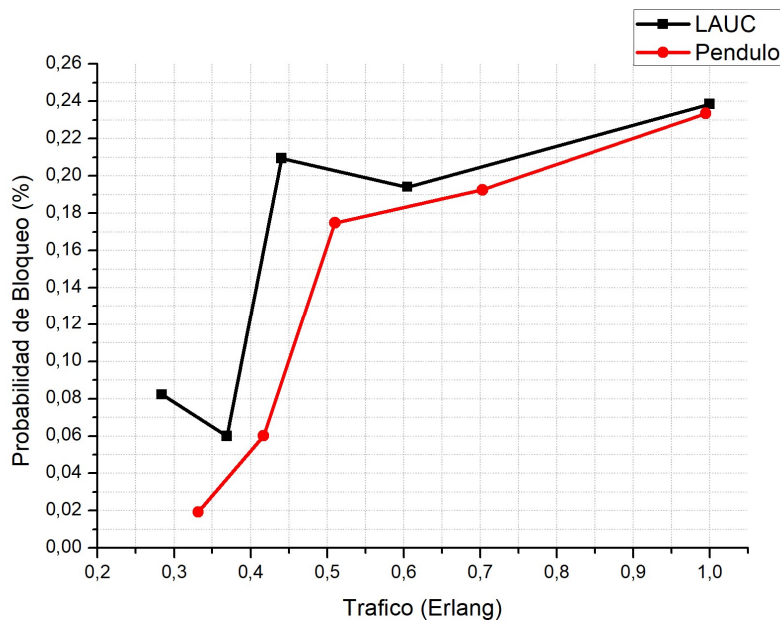


Figura 4.9a - P_b con 8 slots y velocidad de 1.25 Gbps

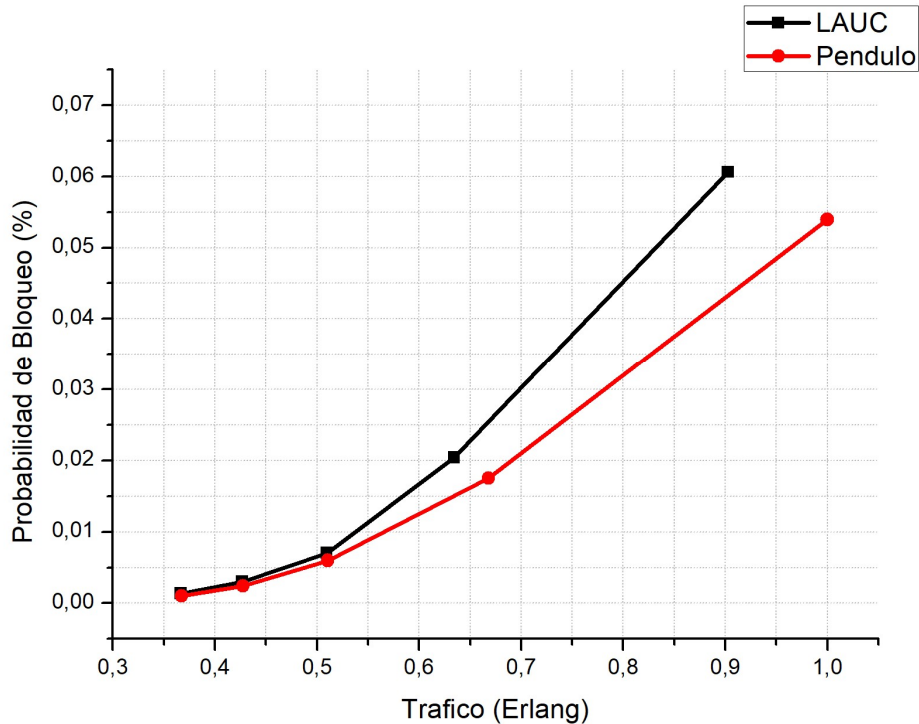


Figura 4.10b - P_b con 8 slots y velocidad de 2.5 Gbps

En el caso a, el algoritmo péndulo presenta un comportamiento bastante similar al LAUC, aunque con un mejor desempeño para valores bajos e intermedios de tráfico, puede decirse que, para valores elevados de tráfico, ambos algoritmos tienen un comportamiento parecido, llevando a probabilidades de bloqueo similares cercanas al 0.24%. Esta alza en la probabilidad de bloqueo, con respecto al caso de estudio anterior se debe al retardo extra que genera el proceso de centralización, pero no es objetivo de estudio de este trabajo de grado comparar los modos de operación distribuido y centralizado.

Para el caso b, la velocidad de datos es de 2.4 Gbps y se observa una merma de la probabilidad de bloqueo, teniendo un máximo cercano a 0.06% para el algoritmo LAUC y de 0.055 para el péndulo. El comportamiento de ambos algoritmos es similar a medida que aumenta el tráfico de la red, sin embargo, la red con el algoritmo péndulo presenta menor probabilidad de bloqueo, lo que indica que el péndulo realiza una mejor gestión de los recursos de red y a grandes volúmenes de datos o tiempo de operación esto puede llegar a ser relevante.

4.2.2 Caso red centralizada con dieciséis slots a 1.25 Gbps y 2.5 Gbps

Para estos escenarios de simulación, se dispone de dieciséis slots para asignar, por lo que las capacidades de la red se ven incrementadas. La tabla 4,4 muestra los parámetros fijados para este caso.

Tabla 4.4 - Parámetros de simulación, casos 1 y 2, subcaso 2, red centralizada

Parámetro	NSFNeT con LAUC	NSFNeT con péndulo
Algoritmo de enrutamiento	Dijkstra	Dijkstra
Número de slots	16	16
Tiempo de simulación	80 ms	80 ms
Tas de bits	1.25 Gbps – 2.5 Gbps	1.25 Gbps – 2.5 Gbps
Algoritmo de asignación de espectro	LAUC adaptado	Péndulo

La figura 4,6 muestra los resultados obtenidos para estos escenarios de simulación para una red con 16 slots y velocidades de 1.25 y 2.5 Gbps.

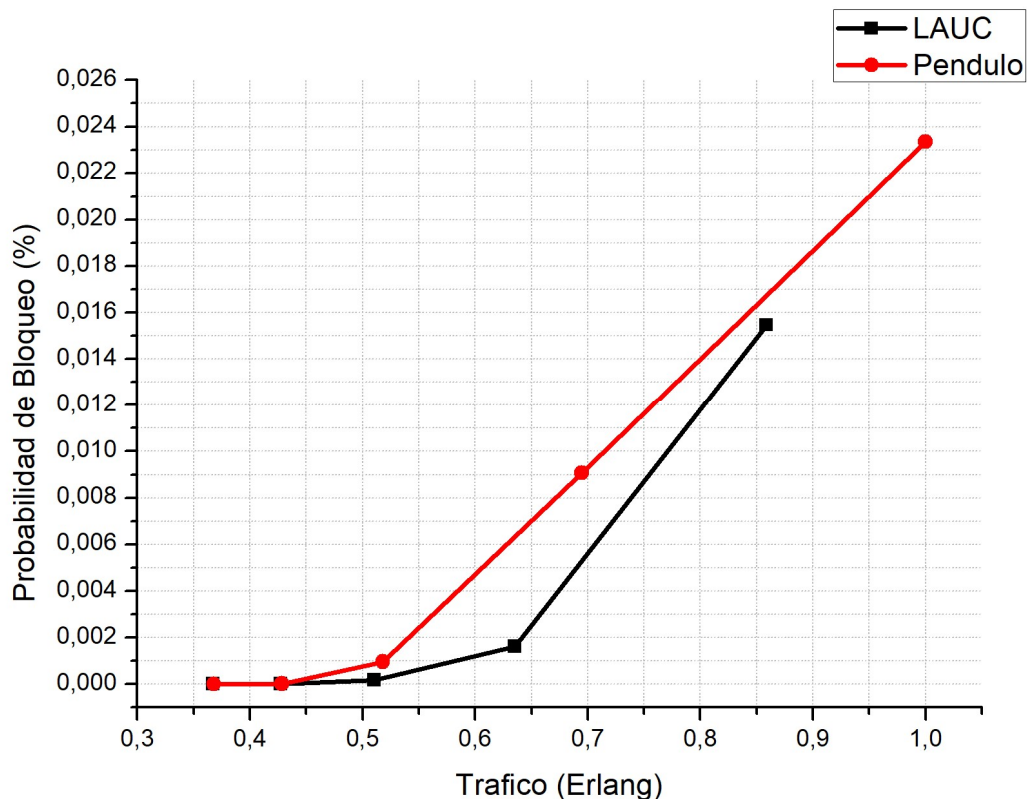


Figura 4.11a - P_b con 16 slots y velocidad de 1.25 Gbps

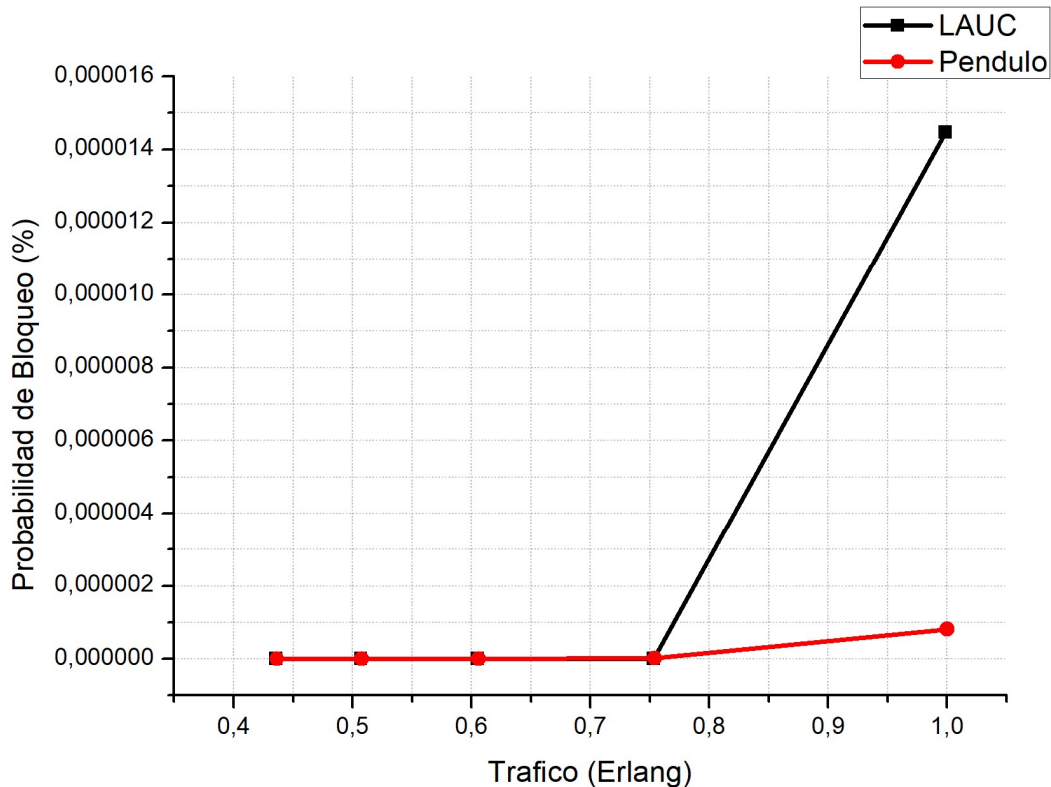


Figura 4.12b - Pb con 16 slots y velocidad de 2.5 Gbps

Se observa del caso a que el algoritmo LAUC tiene un mejor comportamiento que el algoritmo péndulo para valores medios y altos de tráfico, llegando a una Probabilidad de bloqueo máxima cercana al 0.016%, mientras que el algoritmo péndulo alcanza un Pb de hasta 0.023%. Para este escenario de red el algoritmo péndulo no presenta el comportamiento deseado debido a que el retardo extra a causa de la centralización causa demora en la asignación de los slots.

En el caso b, el algoritmo Péndulo presenta un mejor comportamiento que el LAUC, la Pb se mantiene en cero para cargas medias y para cargas altas alcanza un valor cercano a 0.000001% para el péndulo y hasta 0.0000145% para el LAUC. En este caso, la velocidad de datos lleva a un mejor comportamiento del algoritmo propuesto en comparación con el caso anterior.

En la siguiente sección se presentan los resultados obtenidos para el retardo de extremo a extremo de la red centralizada.

4.2.3 Caso red centralizada con ocho slots a 1.25 Gbps y 2.5 Gbps, estudio del retardo extremo a extremo

La tabla 4,3 muestra los parámetros fijados para el análisis de estos casos de estudio. En la figura 4,7 se observan los resultados obtenidos.

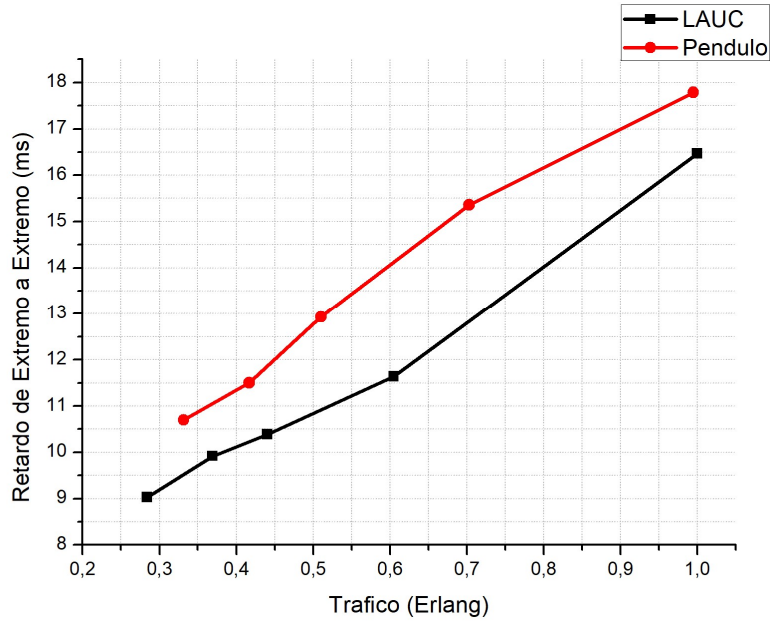


Figura 4.13a - Retardo de extremo a extremo, 8 slots y velocidad de 1.25 Gbps

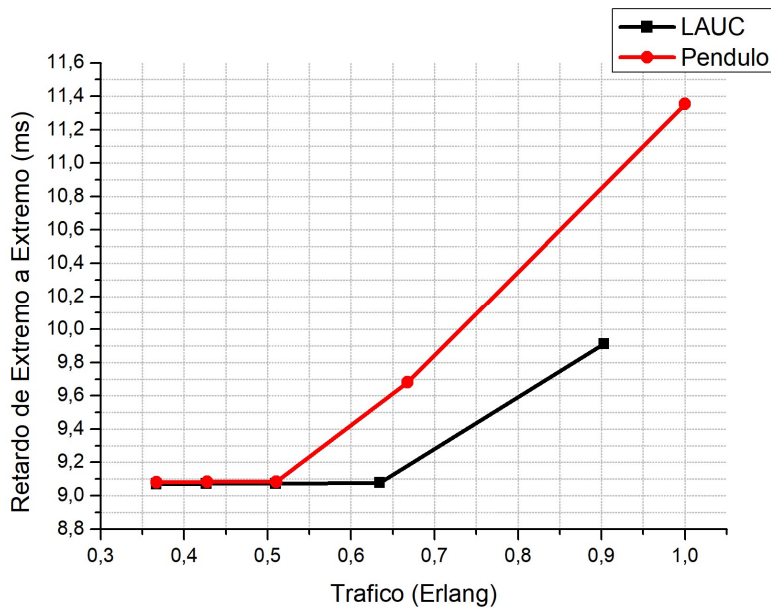


Figura 4.14b - Retardo de extremo a extremo, 8 slots y velocidad de 2.5 Gbps

En el caso a, el algoritmo péndulo presenta mayores retardos que el LAUC, esto se debe al proceso de centralización, indicando que el algoritmo propuesto es sensible a los retardos de la red. Por lo que para este escenario centralizado tiene mejor desempeño el algoritmo tradicional. El caso a, es el más crítico de todos, pues aquí se alcanzan retardos del orden de los 18 ms para el péndulo y de hasta 16 ms para el LAUC. Si bien la diferencia entre los dos no es mucha, puede llegar a ser considerable con servicios basados en el tiempo de respuesta.

En el caso b, el algoritmo péndulo presenta un desempeño inferior que el LAUC, llegando a alcanzar retardos de hasta 11.3 ms mientras que su contraparte clásica llega a retardos cercanos a 9.9 ms. Se observa que el aumento de velocidad entre un caso y otro es benéfico para la red, pues ambos algoritmos redujeron su retardo en varias unidades.

4.2.4 Caso red centralizada con dieciséis slots a 1.25 Gbps y 2.5 Gbps, estudio del retardo extremo a extremo

La tabla 4,4 muestra los parámetros fijados para el análisis de estos casos de estudio. En la figura 4,8 se observan los resultados obtenidos.

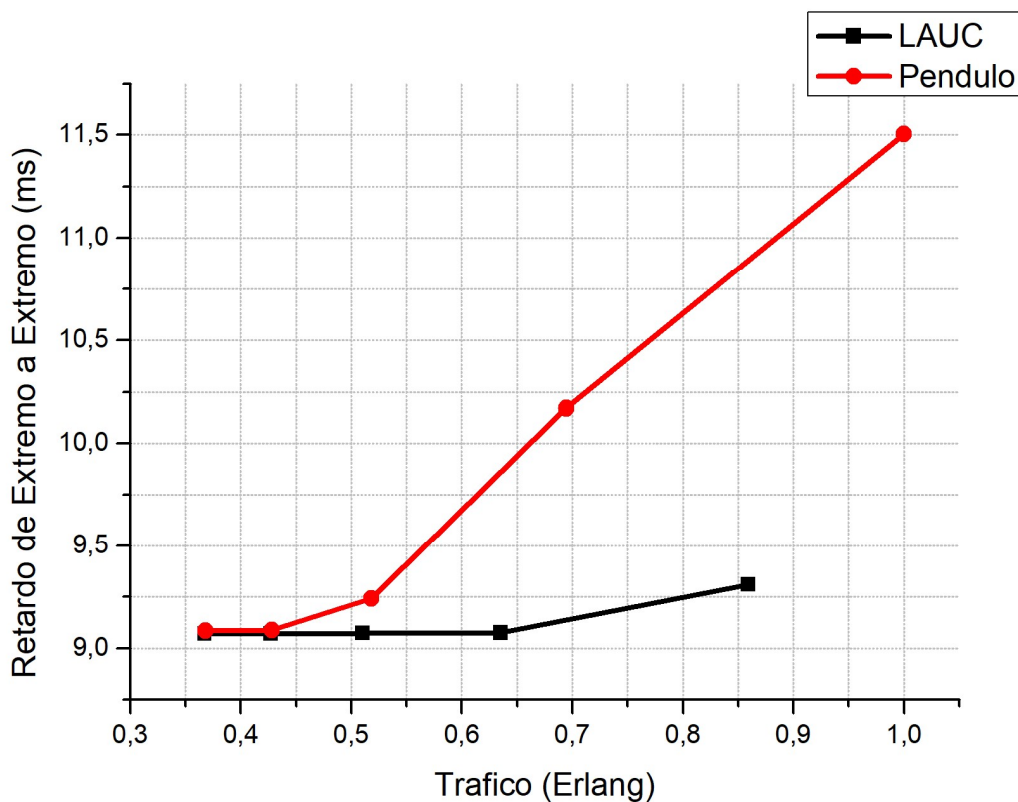


Figura 4.15a - Retardo de extremo a extremo, 16 slots y velocidad de 1.25 Gbps

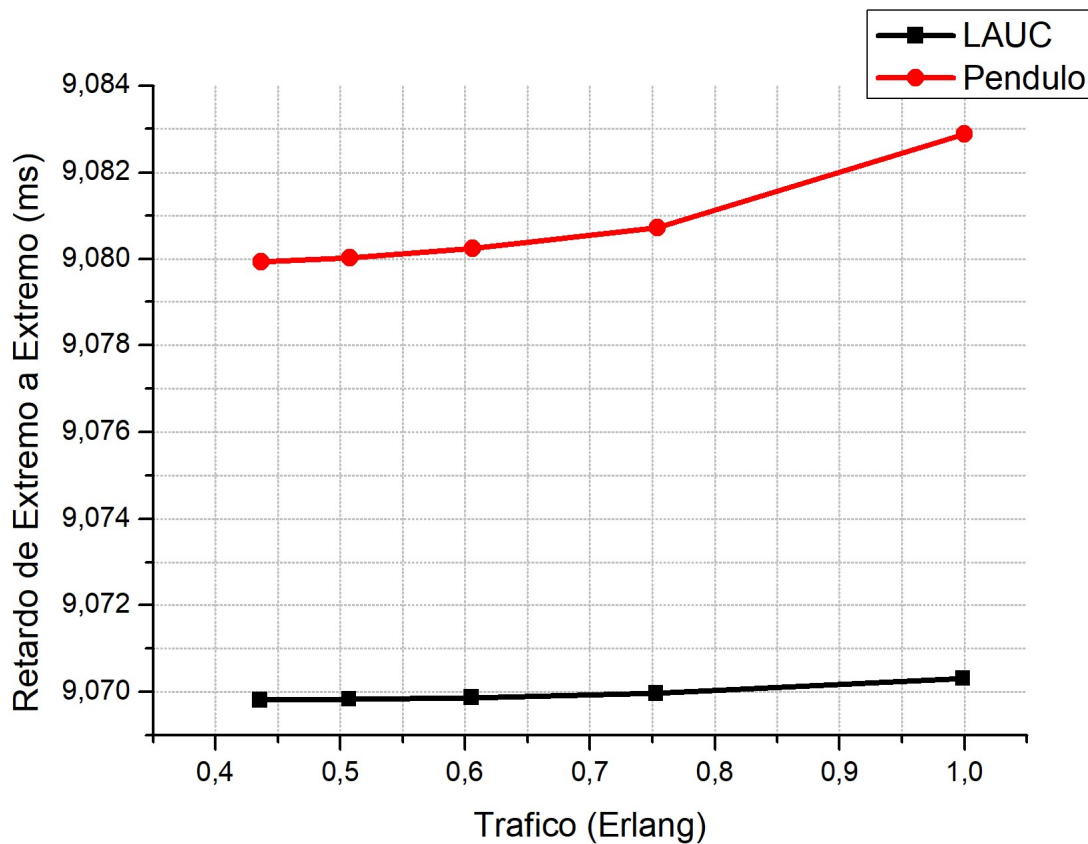


Figura 4.16b - Retardo de extremo a extremo, 16 slots y velocidad de 2.5 Gbps

En el caso a, el algoritmo LAUC presenta un mejor desempeño respecto al retardo presentado por el péndulo, que empeora a medida que el tráfico de red aumenta, llegando hasta 11.5 ms contra 9.35 ms del LAUC. Se hace evidente que el retardo introducido por el proceso de centralización causa una desmejora del algoritmo propuesto, en términos de retardo.

Para el caso b, el aumento a dieciséis slots en la rejilla tiene consecuencias importantes en el desempeño de ambos algoritmos, si bien, el algoritmo péndulo presenta un desempeño menor que el LAUC, la diferencia es de solo 0.013 ms, por lo que, en este punto, el uso de uno u otro algoritmo haría que una red se comporte prácticamente de idéntica manera.

En términos generales puede decirse que el algoritmo propuesto presenta un muy buen desempeño al compararse con un algoritmo típico como LAUC. Para redes distribuidas, el algoritmo péndulo puede ofrecer grandes ventajas de operación reduciendo de forma significativa la cantidad de paquetes perdidos y disminuyendo los retardos de operación de la red. Para redes centralizadas, el comportamiento respecto a la probabilidad de bloqueo es mejor que el LAUC, a costa de un desempeño inferior en el caso del retardo extremo a extremo.

La tabla 4,5 muestra un resumen de los valores promedios obtenidos en la red distribuida, mientras que la tabla 4,6 muestra los datos correspondientes a la red centralizada.

Tabla 4.5 - Síntesis de los resultados de la red distribuida

	Algoritmo LAUC				Algoritmo péndulo			
	8 Slots		16 Slots		8 Slots		16 Slots	
	1.25 Gbps	2.5 Gbps	1.25 Gbps	2.5 Gbps	1.25 Gbps	2.5 Gbps	1.25 Gbps	2.5 Gbps
<i>Probabilidad de bloqueo promedio (%)</i>	0.03416	0.00503	0.00041	0	0.00401	0.00020	0	0
<i>Retardo de extremo a extremo promedio (ms)</i>	5.63114	4.15700	4.08240	4.05124	4.34082	4.05136	4.05149	4.05116

Tabla 4.6 - Síntesis de los resultados de la red centralizada

	Algoritmo LAUC				Algoritmo péndulo			
	8 Slots		16 Slots		8 Slots		16 Slots	
	1.25 Gbps	2.5 Gbps	1.25 Gbps	2.5 Gbps	1.25 Gbps	2.5 Gbps	1.25 Gbps	2.5 Gbps
<i>Probabilidad de bloqueo promedio (%)</i>	0.15680	0.01846	0.00344	28 E -6	0.13605	0.01618	0.00668	16 E -7
<i>Retardo de extremo a extremo promedio (ms)</i>	11.48538	9.24046	9.11975	9.06995	13.65598	9.65688	9.81738	9.08076

La siguiente sección muestra las conclusiones a las que se llegaron a lo largo del trabajo de grado y las posibles recomendaciones para la realización de trabajos similares.

4.3 Conclusiones y Recomendaciones

Aquí se presentan las conclusiones, recomendaciones y trabajos futuros que surgen del desarrollo del trabajo de grado. Las conclusiones se presentan respecto a los resultados de simulación al desarrollo del trabajo de grado y a la herramienta de simulación. Las recomendaciones abarcan aspectos relacionados a la metodología del trabajo y a la herramienta de simulación.

4.3.1 Conclusiones respecto a los resultados obtenidos

- Para redes con poca disponibilidad espectral y baja tasa de bits, el algoritmo péndulo es la mejor alternativa, sin embargo, a medida que los recursos y la velocidad de datos de la red aumentan, la liberación de los slots ocupados por un servicio ocurrirá más rápidamente, por lo que es probable que nuevos paquetes de información encuentren recursos disponibles para ser transmitidos, en cuyo caso, la elección de uno u otro algoritmo es indiferente para garantizar el funcionamiento de la red.
- Para las redes ópticas distribuidas, el algoritmo péndulo, realiza una gestión más eficiente del espectro disponible frente al algoritmo de comparación LAUC, mejorando parámetros como la probabilidad de bloqueo y el retardo extremo a extremo, gracias a que recorre la grilla espectral desde sus extremos hacia el centro, donde es más probable que se encuentren recursos disponibles para transmitir información.
- En el proceso de centralización de la red, el retardo de extremo a extremo aumenta independientemente del algoritmo de asignación de espectro SA empleado, debido, principalmente, a que la carga útil, deben esperar el envío, procesamiento y retorno del paquete de control de centralización PCC, aumentando el tiempo en cola de los paquetes de datos.
- Para servicios sensibles al retardo en redes centralizadas, el algoritmo LAUC se perfila como una mejor opción, dado que, en un escenario centralizado, este algoritmo presenta un retardo de extremo a extremo más bajo que el algoritmo péndulo, gracias a su menor complejidad computacional.
- La aplicación de las técnicas de Machine Learning y redes FlexGrid sobre sistemas DWDM, conducirán a un mejor aprovechamiento de los recursos de red, aumentando la cantidad de paquetes de información que pueden

enviarse sobre un mismo ancho de banda y mejorando la granularidad de la rejilla espectral, del mismo modo, los procesos de centralización, permitirán realizar cambios y adaptaciones de una manera más rápida y eficiente, siendo un requerimiento de los sistemas de próxima generación.

4.3.2 Conclusiones respecto al desarrollo del trabajo de grado

- Al aplicar la aproximación de centralización en la red óptica, el tiempo de simulación de la herramienta se ve disminuido, limitando el estudio de la red para tiempos mayores a los empleados, esto ocurre porque la aplicación de un modelo centralizado incrementa el gasto de recursos computacionales de la herramienta, lo que repercute en la cantidad de tiempo de operación que puede simularse en la misma.
- Algunos parámetros de red en la herramienta OMNeT++ no son susceptibles de control o verificación dado su naturaleza aleatoria.

4.3.3 Conclusiones respecto a las herramientas empleadas

- OMNeT++ se considera como una elección acertada en cuanto a las características que brinda para la construcción de modelos de simulación. Sin embargo, posee una curva de aprendizaje alta, ya que exige un conocimiento en programación orientada a objetos en C++ y requiere de un proceso de familiarización fuerte.
- La herramienta para la realización de las gráficas ORIGIN PRO 8 fue una buena elección, ya que esta presenta gran flexibilidad y facilidad de uso. Además, es una herramienta que permite diferentes formas de representar los datos según las necesidades del usuario.
- La suite de herramientas de Microsoft Office es una excelente alternativa para el desarrollo del documento y el manejo de grandes volúmenes de datos numéricos.

4.3.4 Recomendaciones

- El IDE OMNeT++ es una herramienta robusta que consume gran cantidad de recursos, por lo que se recomienda el uso de equipos de cómputo de alta capacidad, para evitar problemas de bloqueo.

- Antes de emprender un proyecto de alta complejidad en OMNeT++ se recomienda aprender a utilizar sus muchas funciones y a comprender su modo de operación
- Es importante caracterizar la red de trabajo antes del ajuste de los parámetros, puesto que la red limita el rango de operación de los mismo.

4.4 Trabajos Futuros

Con el desarrollo de este trabajo de grado surgen nuevas propuestas de investigación.

- Desarrollo y caracterización de una red óptica RSA totalmente centralizada.
- Adaptación y análisis del algoritmo péndulo para su operación en una red óptica convencional con tecnología RWA.
- Analizar el impacto de velocidades de datos superiores sobre los algoritmos estudiados.
- Diseño e implementación de un módulo cognitivo capaz de elegir entre dos o más algoritmos para la gestión de recursos de red.
- Adaptar el algoritmo de péndulo para la búsqueda y asignación de rutas en una red óptica.
- Analizar el comportamiento de los algoritmos cognitivos para abordar el problema de enrutamiento en redes flexibles.

REFERENCIAS

- [1] ITU-T, "ITU-T G.872," *SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS*. ITU, 2017.
- [2] G. P. Agrawal, *Fiber-Optic Communications Systems*, Third Edit., vol. 6. New York, USA, 2002.
- [3] "CWDM - DWDM - Comparación entre técnicas WDM, página web disponible en: <https://community.fs.com/blog/what-is-the-difference-between-dwdm-and-cwdm-optical-technologies.html/>."
- [4] S. V Kartalopoulos, *Introduction To Dwdm Technology*, First edit., no. 6387. Piscataway, Nj, USA: IEEE Press, 2000.
- [5] ITU-T, "G.694.1 (02/2012), Spectral grids for WDM applications: DWDM frequency grid," *Ser. G.694.1*, pp. 1–16, 2012.
- [6] "Sistema DWDM, página web disponible en: <https://www.pandacomdirekt.com/en/technologies/wdm/what-is-dwdm.html/>."
- [7] L. Steels, "The Evolution of Communication Systems by Adaptive Agents," in *Deploying QoS for Cisco IP and Next Generation Networks*, 1st ed., 2003, pp. 125–140.
- [8] N. Antoniadis, G. Ellinas, and I. Roudas, *WDM Systems and Network, Modeling, Simulation, Desing and Engineering*, First. New York, USA: Springer, 2012.
- [9] "Red óptica típica, página web disponible en: http://faculty.uml.edu/vinod_vokkarane/soon/index.html/."
- [10] S. Verma, H. Chaskar, and R. Ravikanth, "Optical Burst Switching: A Viable Solution for Terabit IP Backbone," New York, USA, 2000.
- [11] R. Ramaswami and K. N. Sivarajan, *Optical networks: a practical perspective*, Second edi. San Francisco, USA: Morgan Kaufmann publishers, 2009.
- [12] V. Lopez and L. Velasco, *Elastic Optical Networks: Architectures, Technologies, and Control*, First edit. New York, USA: Springer, 2016.
- [13] J. G. L. PERAFÁN, "DISEÑO DE MÉTODOS CROSS LAYER COGNITIVOS PARA REDES DE COMUNICACIÓN ÓPTICA DE RÁFAGAS (OBS)," Universidad del Cauca, 2014.
- [14] C. Qiao and M. Yoo, "Optical burst switching (OBS)--a new paradigm for an Optical Internet," *J. high speed networks*, vol. 8, no. 1, pp. 69–84, 1999.
- [15] S. J. Ben Yoo, "16 – Optical burst and packet switching," in *Optical Fiber Telecommunications V B*, First edit., Sacramento, Ca, USA: Springer, 2008, pp. 641–693.
- [16] M. Maier, *Optical Switching Networks*, First edit. Montreal, Canada: Cambridge University Press, 208AD.
- [17] S. Chagas, "Uma abordagem distribuída para o problema de roteamento e alocação de comprimentos de ondas em redes WDM," Universidad de Brasilia, 2010.

- [18] M. A. Aydin, T. Atmaca, O. C. Turna, H. Zaim, and T. Eido, "A performance comparison of OPS with variable-size packets and OBS," *Opt. Switch. Netw.*, vol. 7, no. 2, pp. 85–94, 2010.
- [19] I. SZCZESNIAK, "Overview of optical packet switching," *Projekty.litis.Gliwice.Pl*, vol. 21, no. 3, pp. 167–180, 2009.
- [20] T. S. El-Bawab and J. D. Shin, "Optical packet switching in core networks: Between vision and reality," *IEEE Commun. Mag.*, vol. 40, no. 9, pp. 60–65, 2002.
- [21] G. N. Rouskas and L. Xu, "Chapter 1 OPTICAL PACKET SWITCHING," in *Optical WDM Networks: Past Lessons and Path Ahead*, First edit., Carolina, USA: Raleigh, 2004, pp. 1–19.
- [22] C. Ware, W. Samoud, P. Gravey, and M. Lourdiane, "Recent advances in optical and hybrid packet switching," *Int. Conf. Transparent Opt. Networks*, vol. 2016–August, pp. 7–10, 2016.
- [23] D. Sharma and D. (Col) S. Kumar, "An Overview of Elastic Optical Networks and its Enabling Technologies," *Int. J. Eng. Technol.*, vol. 9, no. 3, pp. 1643–1649, 2017.
- [24] CISCO, "The Zettabyte Era: Trends and Analysis', white paper," 2017.
- [25] P. M. Pereira, "Redes ópticas elásticas," Universidade de São Paulo, 2013.
- [26] M. A. I. Tomkos, E. Palkopoulou, "A Survey of Recent Developments on Flexible/Elastic Optical Networking I.," Athens, Greece, Mo.C1.1, 2012.
- [27] A. Mayoral, "Algoritmo de planificación para redes elásticas," Universidad autónoma de Madrid, 2013.
- [28] P. S. Khodashenas and A, "Dynamic Routing and Spectrum Allocation in Elastic Optical Networks," Universitat Politècnica de Catalunya, 2014.
- [29] A. F. Sevilla Majin and E. C. Zúñiga Quisoboní, "ANÁLISIS COMPARATIVO DEL DESEMPEÑO DE ALGORITMOS RSA Y RWA SOBRE UNA RED ÓPTICA BASADA EN LA TOPOLOGÍA NSFNET," Universidad del Cauca, 2017.
- [30] N. G.-G. Jhon James Granada-Torres, Ana María Cárdenas-Soto, "Redes ópticas elásticas: un nuevo paradigma en las futuras redes de telecomunicaciones," *Connect. tooth-implant a viable Altern. Case Rep. using removable Partial denture Support. teeth Implant.*, vol. 20, pp. 6–22, Dec. 2015.
- [31] S. Talebi, F. Alam, I. Katib, M. Khamis, R. Salama, and G. N. Rouskas, "Spectrum management techniques for elastic optical networks: A survey," *Opt. Switch. Netw.*, vol. 13, no. 2, pp. 34–48, 2014.
- [32] "OFDM, Representación gráfica de una señal OFDM, página web disponible en <http://www.revolutionwifi.net/revolutionwifi/2015/3/how-ofdm-subcarriers-work/>."
- [33] K. Christodouloupoloulos, I. Tomkos, and E. Varvarigos, "Dynamic Bandwidth Allocation in Flexible OFDM-based Networks," Patra, Greece, OSA/OFC/NFOEC 2011, 2011.
- [34] A. Castro, L. Velasco, M. Ruiz, M. Klinkowski, J. P. Fernández-Palacios, and D. Careglio, "Dynamic routing and spectrum (re)allocation in future flexgrid

- optical networks,” *Comput. Networks*, vol. 56, no. 12, pp. 2869–2883, 2012.
- [35] A. A. Garsia, “Elastic spectrum allocation in flexgrid optical networks,” Universitat Politècnica de Catalunya, 2012.
- [36] T. D. Nadeau and K. Gray, *SDN: Software Defined Networks*, First edit. Sebastopol, CA, USA: O’Reilly Media, Inc., 2013.
- [37] A. S. Thyagaturu, A. Mercian, M. P. McGarry, M. Reisslein, and W. Kellerer, “Software Defined Optical Networks (SDONs): A Comprehensive Survey,” *IEEE Commun. Surv. Tutorials*, vol. 18, no. 4, pp. 2738–2786, 2016.
- [38] G. Gomez, “Final Report Dymlec,” *Proc. IEEE*, vol. 103, no. 1, 2015.
- [39] “SDN architecture, planos de interacción de SDN, página web disponible en: <http://www.magazciturum.com.mx/?p=2353#.XBp0pFxKi00/>.”
- [40] D. Andr, S. C. Tutor, J. Carlos, and G. Cebollada, “Redes Definidas por Software (SDN): OpenFlow,” Universidad politécnica de Valencia, 2015.
- [41] M. Ramírez Giraldo and A. M. López Echeverry, “Redes de datos definidas por software - SDN , arquitectura , componentes y funcionamiento,” *J. Cienc. e Ing.*, vol. 10, no. 1, pp. 55–61, 2018.
- [42] H. Kim and N. Feamster, “Improving Network Management with Software Defined Networking,” *IEEE Communications Magazine*, no. February, Georgia, USA, pp. 114–119, 2013.
- [43] J. P. Elbers and A. Autenrieth, “From static to software-defined optical networks,” Martinsried, Germany, 2012.
- [44] B. A. A. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turlitti, “A Survey of Software-Defined Networking: Past , Present , and Future of Programmable Networks,” *IEEE Commun. Surv. TUTORIALS*, vol. 16, no. 3, pp. 1617–1634, 2014.
- [45] Y. Zhao, Y. Wang, W. Wang, and X. Yu, “Software-Defined OpticalNetworking (SDON): Principles and applications,” in *Intech open*, First edit., vol. 2, Beijing, China, 2018, p. 64.
- [46] D. E. Simeonidou, R. Nejabati, and M. Channegowda, “Software Defined Optical Networks Technology and Infrastructure: Enabling Software-Defined Optical Network Operations,” *Opt. Fiber Commun. Conf. Fiber Opt. Eng. Conf. 2013*, vol. 5, no. 10, p. OTh1H.3, 2013.
- [47] B. Collings, “New devices enabling software-defined optical networks,” *IEEE Commun. Mag.*, vol. 51, no. 3, pp. 66–71, 2013.
- [48] S. Gringeri, N. Bitar, and T. J. Xia, “Extending Software Defined Network Principles to Include Optical Transport,” *IEEE Communications Magazine*, New York, USA, pp. 32–40, 2013.
- [49] E.-G. Talbi, *Metaheuristics: From Design to Implementation*, First., vol. 91. Lille, France: Wiley, 2009.
- [50] B. Melián, J. A. Moreno Pérez, J. Marcos, and M. Vega, “Metaheurísticas: una visión global,” *Rev. Iberoam. Intel. Artif.*, vol. 19, pp. 7–28, 2003.
- [51] “Diagrama de metaheurísticas, página web disponible en: <https://es.wikipedia.org/wiki/Metaheur%C3%ADstica>.”
- [52] S. Ólafsson, “Chapter 21 Metaheuristics,” in *Handbooks in Operations Research and Management Science*, First., vol. 13, no. C, Des Moines,

- Iowa, USA: Elsevier, 2006, pp. 633–654.
- [53] J. M. C. Gonzales, “Metaheurísticas y computación paralela para el problema de la planificación de frecuencias en redes reales de telecomunicaciones,” Universidad de Extremadura, 2011.
- [54] J. F. Chicano García, “Metaheurísticas e Ingeniería del Software,” Universidad de Málaga, 2007.
- [55] X.-S. Yang, *Metaheuristic Applications in structures and infrastructures*, First edit. London, U.K: Elsevier, 2013.
- [56] K. P. Murphy, *Machine Learning: A probabilistic perspective*, First edit. Massachusetts, USA: The MIT Press, 2012.
- [57] S. Shalev-Shwartz and S. Ben-David, *Machine learning: From theory to algorithms*, First., vol. 9781107057. Waterloo, Canada: Cambridge University Press, 2013.
- [58] R. W. Thomas, “Cognitive Networking,” Virginia Polytechnic Institute and State University, 2009.
- [59] A. Amanna and J. H. Reed, “Survey of cognitive radio architectures,” Virginia, USA, 978-1-4244-5853-0, 2010.
- [60] Qusay H. Mahmoud, *Cognitive Networks: Towards Self-Aware Networks*, Qusay H. M., vol. 91. Guelph, Canada: Wiley, 2017.
- [61] C. Fortuna and M. Mohorcic, “Trends in the development of communication networks: Cognitive networks,” *Comput. Networks*, vol. 53, no. 9, pp. 1354–1376, 2009.
- [62] R. W. Thomas, D. H. Friend, L. A. DaSilva, and A. B. MacKenzie, “Cognitive networks: Adaptation and learning to achieve end-to-end performance objectives,” *IEEE Commun. Mag.*, vol. 44, no. 12, pp. 51–57, 2006.
- [63] H. Arslan, *Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems*, First. Tampa, FL, U.S.A: Springer, 2007.
- [64] S. Shakya, “Management of Spectral Resources in Elastic Optical Networks,” Georgia State university, 2015.
- [65] S. Ricardo and L. Ribeiro, “Roteamento Multicritério em Redes Ópticas Elásticas Dinâmicas,” Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, 2014.
- [66] M. & JAAR, *Programación en MATLAB*, First edit. USA: MRA & JAAR, 2010.
- [67] T. Philip, “MATLAB - Wikipedia,” *Wikipedia*, 2018. [Online]. Available: <https://en.wikipedia.org/wiki/MATLAB>.
- [68] L. Update, M. Overview, S. Only, S. Only, and L. Mathworks, “MATLAB Overview,” *Tutorialspoint*, 2017. [Online]. Available: <https://www.tutorialspoint.com/matlab/index.htm>. [Accessed: 20-Aug-2012].
- [69] synopsys, “OptSim Product Overview,” *Synopsys*, 2018. [Online]. Available: <http://optics.synopsys.com/rsoft/rsoft-system-network-optsim.html>.
- [70] RSOFTE, “Diseño de Sistemas y Redes – OptSim,” USA, 2016.
- [71] A. Varga and R. Hornig, “AN OVERVIEW OF THE OMNeT++ SIMULATION ENVIRONMENT,” Budapest, Hungary, 2008.
- [72] A. Varga, “A Quick Overview Of The OMNeT ++ IDE,” Budapest, Hungary, 2017.

- [73] Roger S. Pressman, *Ingeniería de software - Un Enfoque practico*. 2010.
- [74] R. M. Beatriz, V. H. Mireya, D. V. Carlos, L. O. Marco A., and M. M. Guillermina, "Simulación digital," Tecnológico de estudios superiores de Jocotitlan, 2008.
- [75] I. de M. Jiménez, "Diseño y modelado de redes ópticas con encaminamiento por longitud de onda y conmutación de ráfagas," Universidad de Valladolid, 2002.
- [76] A. Salas, "Acerca del Algoritmo de Dijkstra," Manizales, Colombia, 2008.
- [77] L. A. Serna, R. M. Unanue, and M. R. Artacho, *Programación y estructura de datos avanzadas*, First edit. Madrid, Spain: UNED.