

ANÁLISIS DEL DESEMPEÑO DE UNA RED ÓPTICA DEFINIDA POR
SOFTWARE EMPLEANDO UN MÉTODO DE ASIGNACIÓN DINÁMICA DE
RUTAS BASADA EN ALGORITMOS GENÉTICOS



BRAYAN SMITH GARCIA ASPRILLA
ESTEBAN ACOSTA VIVAS

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Grupo I+D nuevas tecnologías en telecomunicaciones
Departamento de Telecomunicaciones
Popayán, 2021

ANÁLISIS DEL DESEMPEÑO DE UNA RED ÓPTICA DEFINIDA POR
SOFTWARE EMPLEANDO UN MÉTODO DE ASIGNACIÓN DINÁMICA DE
RUTAS BASADA EN ALGORITMOS GENÉTICOS

BRAYAN SMITH GARCIA ASPRILLA
ESTEBAN ACOSTA VIVAS

Trabajo para optar por el título de
Ingeniero en Electrónica y Telecomunicaciones

Director(a):
MsC. Catalina Muñoz Collazos

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Grupo I+D nuevas tecnologías en telecomunicaciones
Departamento de Telecomunicaciones
Popayán, 2021

*Dedico este logro a mi familia,
Especialmente a mi madre
A mis profesores que me formaron,
Amigos y personas que me han apoyado a lo largo de la vida*

Brayan Smith Garcia.

*Dedico este logro a mis padres y hermanita,
A mis profesores,
A Tony Stark por inspirarme a ser ingeniero
genio, millonario y filántropo.*

Esteban Acosta Vivas.

Agradecimientos

Los autores expresan sus agradecimientos a la ingeniera Catalina Muñoz Collazos, Directora del proyecto, por sus valiosas orientaciones.

Agradecimientos a la Universidad del Cauca, en especial al programa de Ingeniería Electrónica y Telecomunicaciones por sus años de formación.

Agradecimiento muy especial a sus familiares y amigos por su acompañamiento y apoyo incondicional, sin los cuales no podríamos culminar satisfactoriamente esta etapa de nuestras vidas.

Al profesor Dr. Alfonso Ariza Quintana de la Universidad de Málaga y al Dr. Alan Díaz Montiel del Trinity College Dublin por su tiempo y apoyo.

Contenido

Dedicatoria	III
Agradecimientos	IV
Lista de figuras	VII
Lista de tablas	IX
Lista de Acrónimos	X
Resumen	XII
Resumen	XIII
1. REDES ÓPTICAS, REDES ÓPTICAS DEFINIDAS POR SOFTWARE Y ALGORITMOS GENÉTICOS	1
1.1. Redes Ópticas	1
1.1.1. Técnicas de multiplexación en redes ópticas	2
1.1.2. Técnicas de conmutación	4
1.2. Redes ópticas definidas por software	6
1.2.1. SDN	7
1.2.2. SDON	9
1.3. Enrutamiento y Asignación de Longitud de Onda	9
1.4. Algoritmos genéticos	10
2. METODOLOGÍAS	16
2.1. Metodología de trabajo	16
2.2. Metodología de Simulación	17
2.2.1. Herramientas de simulación	18
2.2.2. Definición del escenario de simulación	21
2.2.3. Parámetros de desempeño	28
2.2.4. Diseño del algoritmo adaptado.	28
2.2.5. Descripción de casos de uso	35
3. IMPLEMENTACIÓN Y RESULTADOS	36
3.1. Implementación de la SDON sin el algoritmo adaptado	38
3.1.1. Caso 1: 5 longitudes de onda	38
3.1.2. Caso 2: 10 longitudes de onda	39
3.2. Implementación de la SDON con algoritmo GA	40
3.2.1. Caso 1: 5 longitudes de onda	40
3.2.2. Caso 2: 10 longitudes de onda	41

4. ANÁLISIS DE RESULTADOS	44
4.1. Análisis de resultados escenario 1: SDON sin el método.	44
4.2. Análisis de resultados Escenario 2: SDON con GA	47
4.3. Comparación de desempeño SDON con y sin el método.	49
4.3.1. 5 longitudes de onda, Velocidad 1.25Gbps	49
4.3.2. 5 longitudes de onda, Velocidad 2.5Gbps	51
4.3.3. 10 longitudes de onda, Velocidad 1.25Gbps	53
4.3.4. 10 longitudes de onda, Velocidad 2.5Gbps	55
4.4. Comparación 5L vs 10L	57
4.5. Comparativa promedio parámetros de desempeño.	59
5. CONCLUSIONES Y TRABAJOS FUTUROS	61
5.1. CONCLUSIONES	61
5.1.1. TRABAJOS FUTUROS	62
Bibliografía	62
Anexos	68

Lista de figuras

1.1. Multiplexación por División de Longitud de Onda	3
1.2. Canales de trabajo de WDM	4
1.3. Operación de red bajo OCS	6
1.4. Arquitectura de SDN	7
1.5. Cruce de un punto	13
1.6. Cruce de puntos	13
1.7. Cruce uniforme,	14
1.8. Pseudocódigo del algoritmo genético	15
1.9. Topología NSFNeT	15
2.1. Metodología en cascada	16
2.2. Diagrama de flujo de metodología de simulación	18
2.3. GUI GNS3	19
2.4. GUI NS2	19
2.5. GUI NS3	20
2.6. GUI OMNeT++	21
2.7. Usuario y Nodo Frontera	22
2.8. Nodo frontera	23
2.9. Nodo de control	23
2.10. Selección de ruta	24
2.11. Composición del nodo de control	24
2.12. Composición Control-WRON-Dinámico, capturada en simulación	25
2.13. Tabla de enlaces, capturada en simulación.	27
2.14. Atributos de la información reenviada por la red, capturada en simulación.	27
2.15. Diagrama de flujo algoritmo SP, elaboración propia	30
2.16. Codificación del algoritmo Shortest Path en OMNeT	31
2.17. Diagrama de flujo algoritmo GA, elaboración propia	32
2.18. Codificación para la función de aptitud	34
2.19. Metodo ruleta sesgada.	34
3.1. Estado de tiempos de la red.	37
4.1. Probabilidad de bloqueo método SP.	44
4.2. Retardo extremo a extremo método SP	45
4.3. Throughput método SP	46
4.4. Probabilidad de bloqueo para todas las variaciones en el GA	47
4.5. Retardo extremo a extremo del GA	48
4.6. Throughput del GA	49
4.7. Comparación de probabilidad de bloqueo con y sin el método, 5L a 1.25Gbps	50

4.8. Comparación de retardo extremo a extremo con y sin el método, 5L a 1.25Gbps	51
4.9. Comparación de retardo extremo a extremo con y sin el método, 5L a 1.25Gbps	51
4.10. Comparación de probabilidad de bloqueo con y sin el método, 5L a 2.5Gbps	52
4.11. Comparación de retardo con y sin el método, 5L a 2.5Gbps	52
4.12. Comparación Throughput con y sin el método, 5L a 2.5Gbps	53
4.13. Comparación de probabilidad de bloqueo con y sin el método, 10L a 1.25Gbps	54
4.14. Comparación de retardo extremo a extremo con y sin el método, 10L a 1.25Gbps	54
4.15. Comparación del Throughput con y sin el método, 10L a 1.25Gbps . . .	55
4.16. Comparación de probabilidad de bloqueo con y sin el método, 10L a 2.5Gbps	55
4.17. Comparación de retardo extremo a extremo con y sin el método, 10L a 2.5Gbps	56
4.18. Comparación del Throughput con y sin el método, 10L a 2.5Gbps . . .	56
4.19. Comparación de la probabilidad de bloqueo 5L vs 10Ls	57
4.20. Comparación de retardo extremo a extremo 5L vs 10L	57
4.21. Comparación del throughput 5L vs 10L	58
4.22. Comparativa promedio de la probabilidad de bloqueo	59
4.23. Comparación promedio del retardo extremo a extremo	59
4.24. Comparación promedio del throughput	60
5.1. Algoritmo para encontrar nodos adyacentes	68
5.2. Algoritmo para cálculo de rutas aleatorias	69
5.3. Algoritmo para selección de rutas	70
5.4. Algoritmo para producir rutas previas al cruce	71

Lista de Tablas

1.1. Comparación tipos de WDM	4
1.2. Características de las técnicas de conmutación.	6
2.1. Comparación herramientas de simulación.	21
2.2. Ilustración casos de estudio	35
3.1. Especificaciones técnicas de los equipos empleados para las simulaciones	37
3.2. Parámetros fijos y variables de las simulaciones	38
3.3. Resultados caso 1, velocidad 1.25Gbps.	38
3.4. Resultados caso 1, velocidad 2.5Gbps.	39
3.5. Resultados caso 2, velocidad 1.25Gbps.	39
3.6. Resultados caso 2, velocidad 2.5Gbps.	40
3.7. Resultados caso 1, velocidad 1.25Gbps, 1 Generación	40
3.8. Resultados caso 1, velocidad 1.25Gbps, 4 Generaciones	41
3.9. Resultados caso 1, velocidad 2.5Gbps, 1 Generación	41
3.10. Resultados caso 1, velocidad 2.5Gbps, 4 Generaciones	41
3.11. Resultados caso 2, velocidad 1.25Gbps, 1 Generación	42
3.12. Resultados caso 2, velocidad 1.25Gbps, 4 Generaciones	42
3.13. Resultados caso 2, velocidad 2.5Gbps, 1 Generación	42
3.14. Resultados caso 2, velocidad 2.5Gbps, 4 Generaciones	43

Lista de Acrónimos

AR	Enrutamiento Adaptativo, <i>Adaptative Routing</i>
BCP	Paquete de Control de Ráfaga, <i>Burst Control Packet</i>
CWDM	Multiplexación por División de Longitud de Onda Gruesa, <i>Coarse Wavelegth Division Multiplexing</i>
DWDM	Multiplexación por División de Longitud de onda Densa, <i>Dense Wavelegth Division Multiplexing</i>
EBI	Interfaz con Dirección al Este, <i>EastBound Interface</i>
FAR	Enrutamiento Fijo Alternativo, <i>Fixed Alternate Routing</i>
FF	Primer Ajuste, <i>First Fit</i>
FX	Enrutamiento Fijo, <i>Fixed Routing</i>
GA	Algoritmo Genético, <i>Genetic Algorithm</i>
IETF	Grupo de Trabajo de Ingeniería en Internet, <i>Internet Engineering Task Force</i>
INET	Telemetría Mejorada de Red Integrada, <i>Integrated Network Enhanced Telemetry</i>
IOS	Sistema Operativo de Internet, <i>Internetwork Operating System</i>
NBI	Interfaz con Dirección al Norte, <i>NorthBound Interface</i>
NOS	Sistema Operativo de Red, <i>Network Operative System</i>
OCS	Conmutación Óptica de Circuitos, <i>Optical Circuit Switching</i>
OBS	Conmutación Óptica de Ráfagas, <i>Optical Burst Switching</i>
ONF	Fundación de Redes Abierta, <i>Open Networking Foundation</i>
OPS	Conmutación Óptica de Paquetes, <i>Optical Packet Switching</i>
OTN	Redes de Transporte Óptico, <i>Optical Trasnport Network</i>
SBI	Enrutamiento Adaptativo, <i>Adaptative Routing</i>

SDON	Redes Ópticas Definidas por Software, <i>Software Defined Optical Network</i>
SDM	Multiplexación por División de Espacio, <i>Space Division Multiplexing</i>
SDN	Redes Definidas por Software, <i>Software Defined Networking</i>
SP	Camino Más Corto, <i>Shortest Path</i>
TDM	Multiplexación por División de Tiempo, <i>Time Division Multiplexing</i>
WBI	Interfaz con Dirección al Oeste, <i>WestBound Interface</i>
WDM	Multiplexación por División de Longitud de Onda, <i>Wavelength Division Multiplexing</i>

Introducción.

El tráfico en las redes de transporte de datos cada vez aumenta debido a una mayor demanda por parte de los usuarios finales, esto incrementa de forma exponencial las capacidades que requieren las redes troncales de información donde se concentra todo el flujo. Los sistemas ópticos de transporte de información son los más acertados para conformar estas redes troncales debido a su gran ancho de banda, bajas pérdidas y baja susceptibilidad a interferencias.

Las redes ópticas definidas por software obtienen un máximo desempeño logrando menores retardos, facilidades de programación y adaptación a los cambios en la red. Entre sus funciones se encuentra la posibilidad de definir como se realiza el enrutamiento en la red desde un sistema centralizado, en el cual se gestionan de forma global las rutas que sigue la información para llegar a su destino[1].

Para el enrutamiento de las redes se usan algoritmos con parámetros definidos que buscan encontrar la mejor ruta de conexión entre origen y destino. Existen diversos tipos de algoritmos, algunos orientados a mejorar un parámetro específico en la red como retardo, número de saltos, probabilidad de bloqueo y otros de enfoque global que buscan mejorar los parámetros de forma general.

Apoyado en lo anterior, este trabajo propone un método de enrutamiento para redes ópticas definidas por software basado en un algoritmo genético, el cual se centra en parámetros de la red específicos y hace parte de los algoritmos evolutivos que imitan el comportamiento de los genes en los seres vivos.

El primer capítulo es una recopilación de información acerca de los temas relacionados con el trabajo, como redes definidas por software, redes ópticas y como funcionan los algoritmos genéticos. En el segundo capítulo se realiza una comparación entre las posibles herramientas de simulación disponibles y se escoge la más adecuada para realizar el trabajo, se define la metodología empleada para realizar las simulaciones y se explica el comportamiento de los algoritmos con los que se realizarán.

El tercer capítulo presenta los resultados obtenidos del funcionamiento de la red con y sin el método de enrutamiento en diferentes escenarios de operación con variaciones en el tráfico, número de longitudes de onda por enlace y velocidad de la red. El cuarto capítulo se realiza el análisis de los resultados, se efectúa una comparación del comportamiento de la red cuando emplea los algoritmos y varían los parámetros de la red. Finalmente en el quinto capítulo se presentan las conclusiones obtenidas a partir del análisis del capítulo cuatro.

Abstract.

Traffic in data transport networks is increasing due to greater demand from end users, this exponentially increases the capacities required by information backbones where all the flow is concentrated. Optical information transport systems are the better to build backbone because their high bandwidth, low losses and low susceptibility to interference.

Software-defined optical networks obtain maximum performance by achieving lower delays, programming facilities and adaptation to changes in the network. Among its functions is the possibility of defining how routing is carried out in the network from a centralized system, in which the routes that the information follows to reach its destination are globally managed.[1].

For the routing of the networks algorithms with defined parameters are used that seek to find the best connection route between origin and destination. There are various types of algorithms, some aimed at improving a specific parameter in the network such as delay, number of hops, blocking probability and others with a global approach that seek to improve the parameters in a general way.

Based on the above, this work proposes a routing method for software-defined optical networks based on a genetic algorithm, which focuses on specific network parameters and is part of the evolutionary algorithms that mimic the behavior of genes in living beings.

The first chapter is a compilation of information about work-related topics such as software-defined networks, optical networks, and how genetic algorithms work. In the second chapter, a comparison is made between the possible simulation tools available and the most appropriate one is chosen to carry out the work, the methodology used to carry out the simulations is defined and the behavior of the algorithms with which they will be carried out is explained.

The third chapter presents the results obtained from the operation of the network with and without the routing method in different operating scenarios with variations in traffic, number of wavelengths per link and network speed. The fourth chapter performs the analysis of the results, a comparison of the behavior of the network is made when using the algorithms and the parameters of the network vary. Finally, the fifth chapter presents the conclusions obtained from the analysis of chapter four.

Capítulo 1

REDES ÓPTICAS, REDES ÓPTICAS DEFINIDAS POR SOFTWARE Y ALGORITMOS GENÉTICOS

Las redes de transporte de datos, como es el caso de las redes de fibra óptica han permitido desarrollar diversas tecnologías que tienen como finalidad aprovechar las altas capacidades de transmisión. El uso de la fibra óptica como red de transporte crea la necesidad de abordar las técnicas de conmutación y multiplexación utilizadas, que en consecuencia llevan a la implementación de diferentes arquitecturas y componentes en la red. La arquitectura de cualquier tipo de red se divide en tres planos: plano de datos, plano de control y plano de gestión, donde el plano de datos transmite la información, el plano de control se encarga de tareas de control y enrutamiento, y el plano de gestión ejecuta tareas como reportes y establecer los requerimientos de la red. De acuerdo con lo anterior, el plano de datos y algunas funcionalidades del plano de control de la red óptica pueden ser abstraídas y unificadas con una Red Definida por Software (SDN, Software Defined Networking), para la construcción de una arquitectura con control centralizado

1.1. Redes Ópticas

La fibra óptica es un medio de transmisión que está conformado por un hilo muy fino de materiales transparentes como plásticos o vidrio, sobre los cuales se envían haces o pulsos de luz que generalmente son emitidos por un diodo emisor o un láser. Las redes ópticas han logrado grandes prestaciones debido a las diferentes formas de manejo que se incorporan dentro de ellas, lo cual da lugar a diferentes generaciones de redes ópticas que describen su evolución.

La primera generación se caracterizó por tener únicamente la fibra como reemplazo del cable de cobre en los principales canales de información, incrementando los costos de operación debido a que la fibra se aprovecha únicamente para la transmisión y el resto de los procesos siguen trabajando en el dominio eléctrico.[2]

La segunda generación incorpora métodos de enrutamiento y protocolos propios de la fibra, así como el enrutamiento por longitud de onda y conmutación de circuitos ópticos además de algunas funciones de control, gestión y protección de la red generando un ahorro en equipos de procesamiento electrónico en los nodos.

Las redes ópticas de tercera generación poseen mecanismos más avanzados de enrutamiento e incorporan la conmutación por ráfagas, donde los paquetes IP son convertidos a paquetes IP ópticos en la entrada de un router frontera (conecta la red óptica con otros tipos de redes).

En general los procesos de incorporación de fibra óptica como medio de transporte de información en las redes de comunicaciones han aumentado en gran medida el ancho de banda permitiendo adaptarse a la demanda creciente de flujo de tráfico en las redes. Además incluyen otros beneficios adicionales de fiabilidad, seguridad y confiabilidad. En la actualidad la fibra óptica se encuentra y se sigue incorporando en todas las partes de las redes, desde las troncales hasta el usuario final. [3]

1.1.1. Técnicas de multiplexación en redes ópticas

Las Redes de Transporte Óptico (OTN, *Optical Transport Networks*) son las indicadas para soportar el alto flujo de tráfico que existe actualmente y que se incrementa de forma exponencial, convirtiéndolas en una opción precisa para el transporte de información[4]. Con el fin de aprovechar al máximo los recursos disponibles y optimizar el uso de las capacidades físicas de transmisión de la red, se implementan tres tipos de técnicas de multiplexación: Multiplexación por División de Espacio (SDM, *Space Division Multiplexing*), que consiste en transmitir en forma paralela la información por medio de canales físicos separados, pero genera subutilización y desperdicio de recursos, Multiplexación por División de Tiempo (TDM, *Time Division Multiplexing*) la cual es generalmente empleada para la transmisión de señales digitales y funciona asignando un ancho de banda durante un intervalo de tiempo a cada conexión y Multiplexación por División de Longitud de Onda (WDM, *Wavelength Division Multiplexing*), la cual es una de las más empleadas dado que permite aumentar drásticamente la capacidad de la red de fibra al enviar varias señales de luz independientes sobre un hilo de fibra, en lugar de enviar uno solo[5].

Para compensar la creciente demanda se implementan diferentes técnicas de multiplexación en la fibra óptica, la WDM permite el envío de varias señales ópticas sobre un mismo cable de fibra usando diferentes portadoras, esto se logra mediante el uso de un multiplexor en el emisor y un demultiplexor en el receptor como se presenta en la figura 1.1 Cada señal se convierte en una fibra virtual que tiene una velocidad y un formato (SONET/SDH, ATM, etc) distinto a las demás señales.

De esta técnica surgen los sistemas de Multiplexación por División de Longitud de Onda Gruesa (CDWM, *Coarse Wavelegth Division Multiplexing*) y Multiplexación por División de Longitud de onda Densa (DWDM, *Dense Wavelegth Division Multiplexing*) los cuales se describen como una evolución de WDM permitiendo el aumento de los canales ópticos y reducir limitaciones relacionadas con a la longitud de onda para obtener mayor distancia[6].

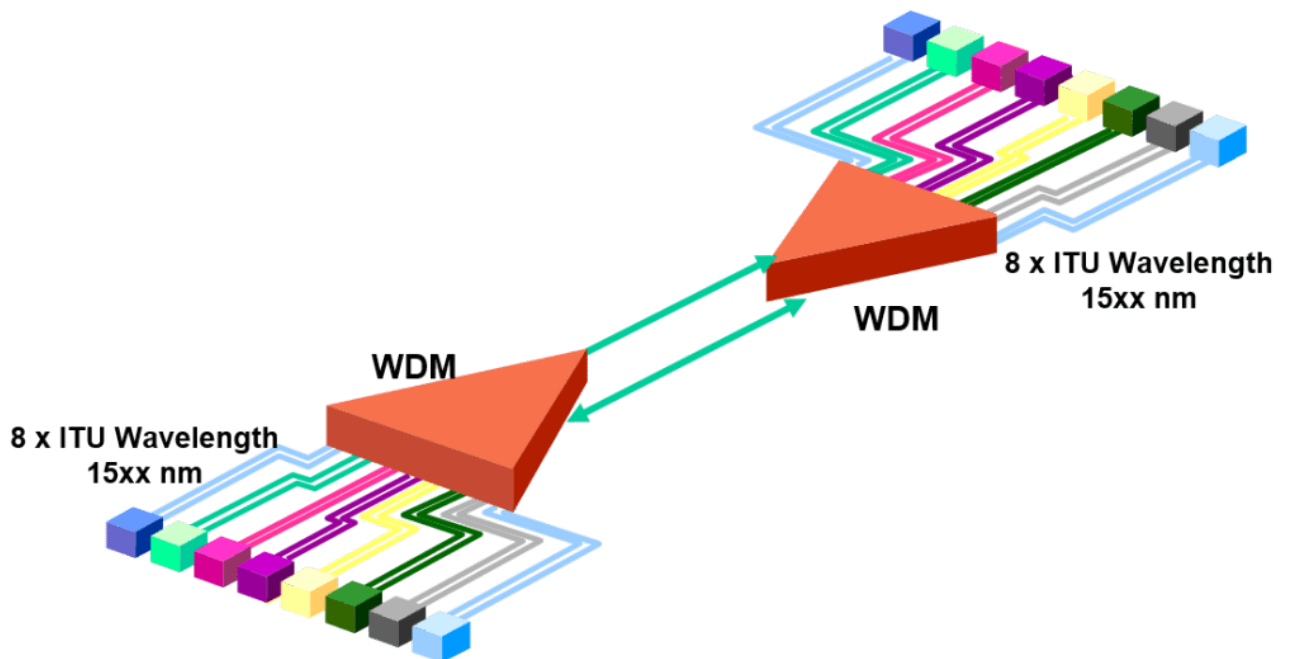


Figura 1.1: Multiplexación por División de Longitud de Onda, tomada de [7]

- **Multiplexación DWDM**

Opera asignando longitudes de onda en la ventana C del espectro de transmisión óptico usando espaciamientos por canal del orden de 100, 50, 25 y 12.5 GHz los cuales son asignados a conveniencia según los requerimientos, este tipo de multiplexación permite la transmisión para múltiples usuarios y múltiples servicios a distancias muy largas y con muy alta velocidad.

Esta variación de WDM a permitido la evolución de las redes globales de datos al permitir que sean heterogéneas y alberguen diferentes servicios. [8].

- **Multiplexación CWDM**

Se usa para redes de costo menos elevado que las DWDM ya que tiene espacios más grandes de separación entre canales aproximadamente de 20 nm y capacidad de transmisión de 8 a 16 canales, hace posible el uso de equipos menos precisos por ende más económicos, lo cual lleva a ser usados en redes que no requieren altas tasas de transmisión ni largas distancias.

En la figura 1.2 se observan los canales en los que funciona cada tecnología especializada de WDM, para entender mejor las características que ofrece cada una, debido a los nanómetros que emplea en su operación[8].

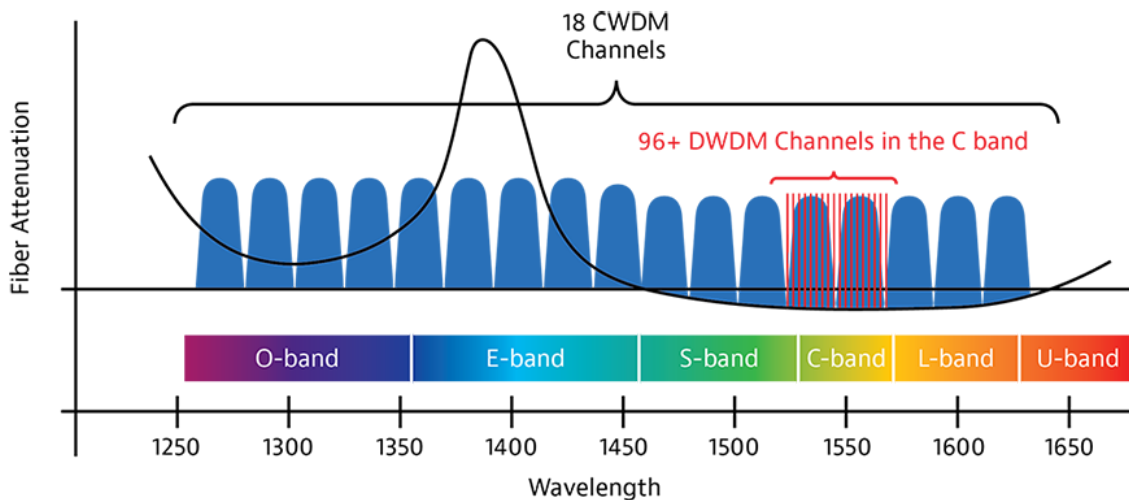


Figura 1.2: Canales de trabajo de WDM, tomada de [9]

En la Tabla 1.1 se encuentran las características de estas multiplexaciones de forma comparativa, debido a que el espaciamiento entre canales, es mucho mayor en los sistemas CWDM, reduce el número de longitudes de onda disponibles para el transporte de información con esta tecnología.

Tabla 1.1: Comparación tipos de WDM

CARACTERISTICAS	CWDM	DWDM MAN/ - WAN	DWDM LARGO ALCANCE
Espaciamiento de los canales	20 nm (2500GHz)	0,8 nm (100GHz)	0,4 nm (50GHz)
Bandas de operación	O, E ,S, C, L	C, L	C, L, S
Canales por fibra	4-16	32-80	80-160
Capacidad por canal	2,2 Gbps	10 Gbps	10 Gbps - 40 Gbps
Distancias típicas	Menor o igual a 80 Km	80 Km - 1000 Km	Mayor a 1000 Km
Costos	Bajo	Medio	Alto

1.1.2. Técnicas de conmutación

La conmutación se define como un proceso de direccionamiento y entrega de información a través de la red haciendo uso eficiente de recursos, a nivel eléctrico este procesamiento introduce mayor retardo en la red, por eso se ha llevado este proceso al dominio óptico y así obtener diseños de red totalmente ópticos, estos procesos describen la forma en la que se dirige la información a través de la red (circuitos o paquetes) y establece la forma en la que trabajan los dispositivos. Se tiene entonces tres casos que

son: Conmutación Óptica de Circuitos (OCS, *Optical Circuit Switching*), Conmutación Óptica de Paquetes (OPS, *Optical Packet Switching*) y la Conmutación Óptica de Ráfagas (OBS, *Optical Burst Switching*)[10].

- **Conmutación Óptica de Ráfagas:** La OBS maneja paquetes más grandes que se generan en un proceso llamado “ensamblaje de ráfagas” en un nodo de entrada, estos paquetes contienen información de diferentes usuarios y servicios, los cuales son desensamblados en cuanto la ráfaga llega a su destino (nodo de salida) y la información es separada para enviarla a su destino correspondiente. La señalización en este tipo de conmutación está conformada por una cabecera llamada Paquete de Control de Ráfaga (BCP, *Burst Control Packet*) que se envía por una longitud de onda diferente con el propósito de configurar los nodos intermedios de la ruta y preparar un camino para la que circule la ráfaga y evitar retardos[11].
- **Conmutación Óptica de Paquetes:** La OPS se caracteriza por transformar los paquetes que vienen de forma eléctrica a paquetes ópticos que llevan una cabecera para el enrutamiento y su carga útil de información, esta cabecera es leída en cada nodo y conmuta la información a un puerto adecuado para su transporte, debido a que actualmente los elementos ópticos no permiten un análisis del paquete o retenerlos mientras son analizados, se debe hacer una conversión óptica a eléctrica en cada nodo para verificar la cabecera, lo que introduce retardos en la transmisión[12].
- **Conmutación Óptica de Circuitos:** La OCS establece rutas a través de nodos OXC las cuales estarán fijas mientras dure la transmisión y son establecidas por una comunicación de control previa. Este tipo de conmutación garantiza el envío de los datos y no introduce retardos durante el viaje de la información ya que no hace conversiones entre dominios óptico y eléctrico. Pero como desventaja, al establecer conexiones fijas de una longitud de onda, ocupa recursos que pueden ser utilizados para otras transmisiones[13].

En la figura 1.3 se observa una red que opera bajo OCS, donde sus enlaces representados de color negro son generalmente de tipo WDM, los caminos ópticos utilizan una longitud de onda que puede ser convertida a otra, debido a que es una funcionalidad que incluyen los nodos OXC de la red.

En la tabla 1.2 se muestran algunas características puntuales correspondientes a las técnicas de conmutación que permiten hacer una comparación entre ellas, donde se puede ver que en la técnica OCS se rechaza la conexión y no hay pérdida de paquetes ni ráfagas, permitiendo que esta información no ingrese a la red de transporte sin garantizar la llegada a su destino y ocupe ancho de banda de forma innecesaria.

La técnica de conmutación OCS al no incorporar elementos de conversión óptica-eléctrica, garantiza un camino establecido y no requiere algún tipo de buffer de almacenamiento óptico durante la transmisión, además en OCS la transmisión tiene más

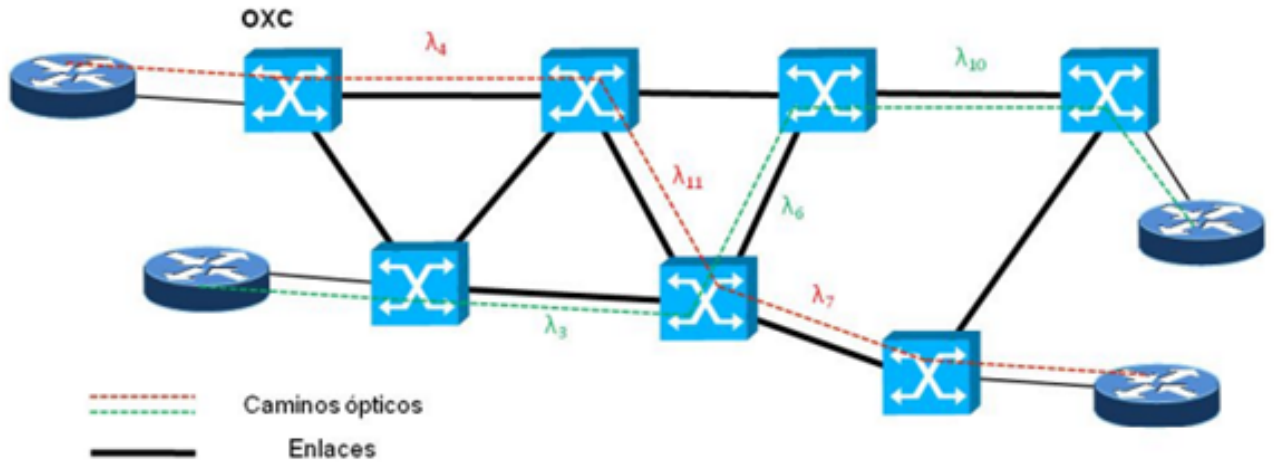


Figura 1.3: Operación de red bajo OCS, tomada de [14]

Tabla 1.2: Características de las técnicas de conmutación.

Técnica de conmutación	OCS	OBS	OPS
Tamaño de datos adecuado	Mayor de Gb	Descenas de Kb	100-1500 B
Garantía de transmisión	Posible	No	No
Tipo de Pérdida	Rechazo de conexión	Pérdida de ráfagas	Pérdida de paquetes
Almacenamiento	No	Si	Si
Latencia dada por	Establecimiento + propagación	Propagación + ensamblaje de ráfaga	Propagación
Desorden en la transmisión	No	No	Posible
Control de cabecera	Establecimiento de conexión	canal de control de ráfaga	Cabecera del paquete

tiempo de conexión y es más probable la presencia de probabilidad de bloqueo, permitiendo evaluar mejor el desempeño de algoritmos de enrutamiento y eficiencia de la red.

1.2. Redes ópticas definidas por software

Las redes ópticas tienen una gran importancia debido a su alta capacidad de transmisión, bajas pérdidas de información y poca susceptibilidad a interferencias, por lo que es necesario el despliegue e incorporación de este tipo de redes y es conveniente impulsar su desarrollo. Debido a las tecnologías que usan la abstracción de redes para simplificar y automatizar la gestión de esta, surgen las SDN, caracterizadas por estructurar la red en tres capas comunicadas por interfaces especializadas brindando a la capacidad de programar las funciones de red y obtener una abstracción de su estado y características. Al combinar la gran capacidad de las redes ópticas y la flexibilidad de gestión de las SDN surgen un nuevo tipo de redes que se conocen como Redes Ópticas Definidas por Software (SDON, *Software Defined Optical Network*) que proveen amplio ancho de

banda y recursos de tecnologías de la información para su gestión. El trabajo con estas redes tiene una amplia similitud con las redes SDN, cambiando algunos factores como la forma de conmutar y de estructura propios de las redes ópticas[15].

1.2.1. SDN

Como lo define el Grupo de Trabajo de Ingeniería en Internet (IETF, *Internet Engineering Task Force*) las SDN son aquellas que permiten programar las redes y gestionarlas, se caracterizan por abstraer y separar el plano de control de la red y el plano de datos permitiendo un mejor y más rápido desarrollo tecnológico en cada uno. Las SDN se componen de tres planos, el plano de aplicación, que hace referencia a las aplicaciones o servicios software que interactúan con el cliente directamente, el plano de control se refiere al controlador junto con sus protocolos y el plano de transporte compuesto por los dispositivos de red. La figura 1.4 permite presenta los tres planos de la arquitectura SDN resaltando sus interfaces y niveles[16].

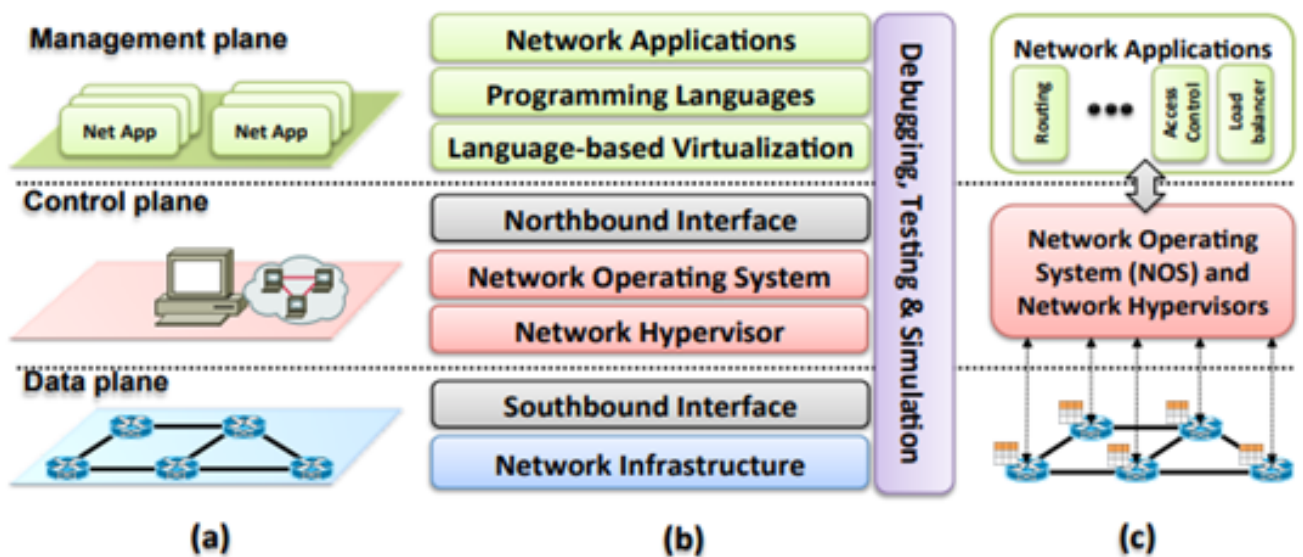


Figura 1.4: Arquitectura de SDN, tomada de [17]

SDN permite la vista simplificada de la infraestructura de la red para el control y monitoreo de esta a través de aplicaciones con la abstracción de la red, sus funciones y cada uno de sus planos de forma independiente las cuales están conectadas por dos interfaces principales, la interface en dirección al sur (SBI, *South-Bound Interface*) que conecta el plano de datos con el plano de control y la interface con dirección al norte (NBI, *North-Bound Interface*) que comunica el plano de aplicación con el plano de control. La Fundación Abierta de Redes (ONF, *Open Networking Foundation*) definió el protocolo OpenFlow como el primero destinado para la SBI. Las arquitecturas SDN pueden coexistir al mismo tiempo con otras, tanto que sean o no SDN y para su comunicación se definen otras interfaces llamadas Interfaz con Dirección al Oeste

(WBI, *WestBound Interface*) e Interfaz con Dirección al Este (EBI, *EastBound Interface*) respectivamente[18].

Planos de SDN

■ Plano de Transporte

Este plano comprende el entorno de transporte de datos, siendo hardware real o virtual, hace referencia a los elementos de red como switch, dispositivos finales y de reenvío, a diferencia de las redes tradicionales, en las cuales el enrutamiento se realizaba en estos dispositivos y no era posible la programación de rutas sin interrumpir la comunicación, en SDN los dispositivos de la capa de infraestructura actúan como simples interruptores que reciben sus órdenes de enrutamiento desde la capa superior permitiendo la mejora en la modificación de paquetes y velocidad de la red[19].

■ Plano de control

Este plano es la entidad lógica responsable de la configuración de los elementos de red presentes en la capa de infraestructura o plano de datos según los requerimientos de la capa de aplicación en otras palabras, toda la lógica de control recae sobre los controladores y aplicaciones que se encuentran en este plano. Para esto los controladores pueden solicitar información del estado de la red, estadísticas de flujo y demás características de infraestructura al plano de datos. El Sistema Operativo de Red (NOS, *Network Operative System*) es una entidad que implementa el controlador SDN que permite y proporciona funciones como virtualización y gestión de base de datos. El NOS permite definir las políticas de la red sobre un lenguaje de alto nivel, la creación de nuevos protocolos y aplicaciones de red de manera sencilla. En resumen, es la clave para la lógica de control ya que debe configurar la red en torno a políticas que defina el operador; abstrae todos los detalles de la red del plano de datos al interactuar con los dispositivos de reenvío por medio de las APIs[19].

La gran mayoría de servicios y aplicaciones de red provienen de distintas plataformas de controlador, por lo que se hizo necesaria la construcción de una tecnología software que permitiera dicha interoperabilidad.

■ Plano de gestión[20].

También llamado plano de aplicación es el que contiene las aplicaciones y servicios de red que usa el plano de control para ejercer las funciones de red. Generalmente el controlador SDN realiza una abstracción de la red para ser presentada a la capa de aplicación que facilita sus funcionalidades y percepción de las necesidades inmediatas de la red, también pueden existir aplicaciones de niveles altos de administración o gestión de red, por ejemplo, para el análisis de datos y funciones especializadas que requieren un gran procesamiento.

1.2.2. SDON

Las redes ópticas debido a su gran ancho de banda ocupan un papel importante en el transporte de información, siendo necesario que evolucionen en cuanto a sus formas de control para automatizar y simplificar su administración, por esto se tiene la idea de trabajar las redes ópticas bajo el concepto de SDN y aprovechar las ventajas que posee la fibra como gran ancho de banda, bajo consumo y grandes distancias de transmisión, junto con la flexibilidad que brinda las SDN[20].

Las redes SDON funcionan con los mismos principios de las SDN, pero con protocolos extendidos y aspectos que se deben tener en cuenta al trabajar con las redes ópticas como el cambio en la conmutación e incorporación de elementos ópticos de transporte. Para llevar una red SDN a una red que use el transporte óptico de información también se debe considerar las formas de cálculo de rutas y dispositivos ópticos que ahora forman parte de la red, así como su compatibilidad con los demás elementos de una red heterogénea[19]. La conmutación en los canales de una red SDON que contienen un espectro amplio permite conmutar puertos y fibras[19].

Existen mecanismos para controlar la capa de infraestructura perteneciente a una comunicación óptica a través de SDN, se deben tener en cuenta los transmisores y receptores que posibilitan controlar las características de la señal óptica, como el formato de modulación[21].

1.3. Enrutamiento y Asignación de Longitud de Onda

El enrutamiento y asignación de longitud de onda se define como el proceso de encaminar la información y asignarle una longitud de onda fija durante ese proceso, durante el evento se establecen caminos de luz entre dos nodos calculando un camino para los datos en una topología física a través de una longitud de onda basándose en la información del estado de la red para poder minimizar recursos[22].

Para el enrutamiento se definen tres formas que se diferencian por la manera en la que determinan el camino que debe seguir la información a través de la red:

- **Enrutamiento fijo**

El Enrutamiento Fijo (FR, *Fixed Routing*) establece una ruta fija o que siempre va a ser la misma entre los nodos de origen y destino, es el método más sencillo de enrutamiento y que menos procesamiento carga a la red, pero presenta inconvenientes como el bloqueo de la solicitud si una ruta falla o los recursos de la red no están disponibles, comúnmente se usan los algoritmos de enrutamiento de camino más corto (SP, *Shortest Path*) como el Dijkstra o Bellman-Ford, estos buscan

realizar el enrutamiento por medio de algoritmos que buscan el menor número de saltos o la menor distancia física entre nodos[23].

- **Enrutamiento fijo alternativo**

El Enrutamiento Fijo Alternativo (FAR, *Fixed Alternate Routing*) emplea unas tablas de rutas en todos los nodos de la red, donde cada una de ellas alberga las rutas alternativas al resto de los nodos de la red. Una ruta alternativa es la que no comparte ningún enlace con la primera ruta de la tabla, que por lo general es la ruta con menos saltos, el algoritmo revisa cada ruta de la tabla de enrutamiento y determina cual tiene los recursos de red necesarios para la transmisión, si no se encuentra una ruta con los recursos disponibles, entonces la transmisión se bloquea, para lo que se trata de asignar de primero, la longitud de onda a las rutas más cortas[24].

- **Enrutamiento adaptativo**

El Enrutamiento Adaptativo (AR, *Adaptive Routing*) es una forma dinámica de asignación de ruta entre los nodos de origen y destino que se realiza tomando en cuenta el estado de todos los nodos en tiempo real en la red, comúnmente se usa la forma de enrutamiento adaptativa de camino más corto, la cual consiste en escoger el camino menos largo, basándose en un valor correspondiente a cada enlace de nodo, donde un enlace libre tiene un valor de 1 y un enlace ocupado un valor de infinito, es decir que se establece la ruta más corta y de menor costo, dado el caso que existan rutas de la misma distancia y costo, se escoge cualquiera aleatoriamente.

Por otra parte, la asignación de la longitud de onda se puede hacer de forma paralela a la asignación de la ruta la cual puede ser realizada de tres formas, la primera es el orden en las que fueron usadas la última vez, por elección aleatoria o por expansión y contracción del espectro de longitudes usadas anteriormente[25].

Existen diferentes algoritmos empleados para la asignación de las longitudes de onda entre los cuales se encuentra el algoritmo de técnica aleatoria, de primer ajuste, de último canal libre disponible, mínima suma, reserva de longitudes de onda, protección de umbral, menos usado y más usado[26].

El Algoritmo de Primer Ajuste (FF, *First-Fit*) enumera las longitudes de onda que existen en cada enlace y cuando se realiza una solicitud de conexión, se escoge la primera longitud disponible entre todas comenzando desde el que tenga asignado el menor número corriendo hacia el de mayor numeración[27].

1.4. Algoritmos genéticos

En la ingeniería existen problemas complejos que comprometen tareas tales como:

- Enrutamiento de paquetes a través de la red.

- Cálculo de rutas cortas u optimizadas.
- Asignación y secuenciación óptima de recursos.

Todos se caracterizan porque agrupan y asignan un conjunto de datos para satisfacer ciertas restricciones, son de gran complejidad computacional y presentan soluciones aproximadas en tiempos razonables. Debido a la gran complejidad de estos problemas, no es óptima su resolución por medio de métodos exactos y se hace uso de técnicas heurísticas o también llamada metaheurística[28].

La metaheurística se refiere a la aplicación de algoritmos y estrategias inteligentes de optimización y búsqueda, que permiten diseñar o mejorar procedimientos heurísticos que tienen una forma genérica, siendo la heurística un método basado en la experimentación, para resolver un problema. Las soluciones ofrecidas por la metaheurística se caracterizan por su alta calidad con un costo computacional moderado, además de ser aproximaciones óptimas y factibles, lo cual las hace ideales para problemas complejos donde no se necesita tener una única solución. Las funciones principales de la metaheurística son la exploración (es decir, la diversificación de la búsqueda en regiones distantes del espacio) y la explotación (es decir, la intensificación de la búsqueda en la región actual), las cuales deben tener un balance para lograr un proceso de búsqueda eficiente.

Los Algoritmos Genéticos (GA, *Genetic Algorithm*) pertenecen a los algoritmos evolutivos, se basan en la genética y en la selección natural, son usados para resolver problemas complejos (dentro de ellos los de tipo no determinísticamente polinómicos o de tipo NP), desarrollando un conjunto de soluciones posibles llamadas población que se reproduce de manera iterativa (generaciones). Se parte de una población inicial elegida al azar que evoluciona al utilizar operadores genéticos inspirados en la selección natural como son: la selección, mutación y el cruce. Los individuos de la población también denominados cromosomas se prueban por medio de una función de aptitud la cual les asigna un puntaje. Este puntaje es la base para la probabilidad de reproducción y determina si la descendencia del cromosoma pasa a la siguiente generación. Idealmente se espera que, con cada generación, la diversidad, aptitud y por ende el rendimiento de los cromosomas descendientes aumente y se mejoren las soluciones[29].

- **Población inicial**

El tamaño de la población inicial de soluciones debe tener un balance para poder cubrir adecuadamente el espacio de búsqueda, a la vez de un moderado costo computacional. La población inicial se genera de manera aleatoria, pero también se puede generar por medio de un proceso de optimización o heurístico. Esto último acelera la convergencia del algoritmo a soluciones dentro de óptimos locales, representando una desventaja para la diversificación del espacio de búsqueda[30].

- **Función aptitud**

La función de aptitud también denominada función objetivo, es fundamental para

establecer el comportamiento del GA, ya que idealmente determina si los individuos de la población representan soluciones óptimas. Consiste en establecer una medida numérica que indica el nivel de aptitud de una solución, es decir; ajusta la probabilidad de supervivencia de ésta para que pueda reproducirse. Esta medición permitirá ejercer control sobre el número de selecciones, cruces y mutaciones que se ejecuten. La función de aptitud se personaliza de acuerdo al tipo de problema, ya que en unos casos se requiere la maximización de ésta y en otros su minimización, por ejemplo, en una población de colonia de hormigas la aptitud dependerá de cuantas piezas de comida puedan llevar al hormiguero, por lo que las más aptas serán las que puedan llevar un mayor número de piezas (maximización), por otro lado se tienen problemas donde se requiera determinar la ruta más corta entre dos puntos, siendo más aptos los individuos con el camino codificado más corto (minimización)[30].

■ Operadores genéticos (Selección, cruce y mutación)

Para el paso de una generación a otra, los GA emplean operadores genéticos que idealmente deben garantizar la adaptación de las posibles soluciones, haciendo que algunos individuos tengan una mayor probabilidad de reproducirse y evolucionar hacia la siguiente generación[31].

El operador de selección se encarga de escoger a los cromosomas que tendrán oportunidad de reproducirse, por lo que esta selección, está relacionada con el valor de ajuste de cada uno. Los métodos de selección se dividen en probabilísticos y determinísticos[32]. En el caso del primero, las probabilidades de escogencia se realizan al azar, teniendo en cuenta que cada cromosoma tiene un valor de ajuste que le otorga un rango de probabilidad menor o mayor; en este grupo se encuentran los métodos de selección por ruleta y torneo. En el segundo método, dado el valor de ajuste de cada individuo, se asigna la cantidad de veces que puede reproducirse, evitando problemas de predominancia y homogeneidad; en este grupo se tiene los métodos de escalamiento sigma, selección por jerarquías, estado uniforme, entre otros[33].

El operador de cruce, que actúa después de la selección de los cromosomas realiza la reproducción o cruce entre dos de ellos, haciendo que intercambien su información genética, con el objetivo de mejorar la aptitud del descendiente en comparación a sus cromosomas padres[34]. Los diferentes métodos de cruce operan en dos formas diferentes:

- Estrategia destructiva: en este caso los cromosomas descendientes serán incluidos en la población temporal sin importar que tengan un valor de ajuste menor que el de sus cromosomas padres.
- Estrategia constructiva: en este caso los cromosomas descendientes serán

incluidos en la siguiente generación sólo si tienen un valor de ajuste mayor al de sus cromosomas padres.

El objetivo del cruce es combinar las mejores características de dos cromosomas padre en un cromosoma hijo, aumentando su valor de ajuste en comparación a sus padres. Los métodos más empleados de cruce son:

- Cruce de puntos.
- Cruce de dos puntos.
- Cruce uniforme.

Para el cruce de un punto, una vez seleccionados dos cromosomas se cortan en un punto seleccionado de manera aleatoria, generando dos segmentos diferentes en cada uno, y luego se intercambia el segundo segmento de cada individuo entre éstos como se muestra en la figura 1.5[35].



Figura 1.5: Cruce de un punto, tomada de [36]

En la figura 1.6 se observa el proceso de cruce de dos puntos en donde el corte se produce en dos puntos, generando tres segmentos. En el cruce uniforme los genes del



Figura 1.6: Cruce de puntos, tomada de[36]

cromosoma hijo tienen la misma posibilidad de pertenecer a cualquiera de los padres. Esta técnica implica el uso de una máscara binaria, en donde los 1's indican los genes que hereda el cromosoma hijo del padre 1, y los 0's corresponden a los genes del padre 2, cada alelo en su respectiva posición. Este proceso se observa en la figura 1.7[35].

El operador de mutación puede ejecutarse después, o de forma paralela al cruce., esto contribuye a la diversidad de la población y previene que las soluciones se queden estancadas en los óptimos locales[37]. El operador consiste en la modificación de ciertos genes del cromosoma (suelen ser un gen) de manera aleatoria en base a la probabilidad

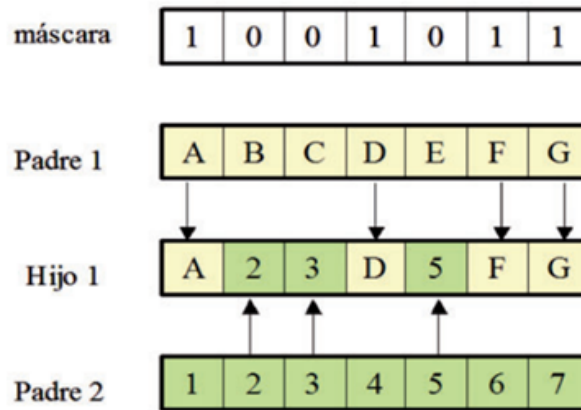


Figura 1.7: Cruce uniforme, tomada de[36]

de mutación que se haya establecido con anterioridad. Sin embargo, es posible que se incremente el tiempo de convergencia y costos computacionales, en consecuencia, este operador no será utilizado en este trabajo de grado[38].

Para el proceso de ejecución del GA, inicialmente se genera una población inicial la cual se evalúa por medio de la función de aptitud, se aplican los operadores genéticos (selección, cruce y mutación) y se realiza el mismo proceso para la siguiente generación. La ejecución de éste termina con la convergencia de la población o con el número de iteraciones establecidas para el proceso[39].

Se dice entonces que los algoritmos genéticos buscan una solución óptima a un problema mediante la aplicación de una función de aptitud y operadores genéticos. Estos operadores genéticos pueden ser usados en diferentes combinaciones e incluso no es imperativo hacer uso de todos dentro de la solución a un problema, estas decisiones se toman durante su diseño. A continuación, se muestra en la figura 1.8 el pseudocódigo de un algoritmo genético que ilustra el proceso mencionado anteriormente[40].

El presente trabajo de grado se basa en una SDON que aplica OCS, debido a que ésta garantiza la transmisión de datos, presenta baja pérdida de información, no requiere buffer de almacenamiento óptico, y la información que viaja por la red con establecimiento previo de conexión va a permitir analizar el uso de los recursos que realiza el método de asignación de ruta. Así mismo se usa la tecnología WDM, que permitirá transmitir información de manera independiente sobre cada una de las longitudes de onda en un enlace. El proceso de enrutamiento se llevará a cabo con y sin el método GA, es decir, con el método se implementa el algoritmo propuesto GA y sin éste se implementa el algoritmo *Shortest Path*, el cual es estándar en las redes tradicionales y será detallado en la siguiente sección. La asignación de longitud de onda será resuelta por el algoritmo *First Fit*[42].

```

BEGIN /* Algoritmo Genetico Simple */
  Generar una poblacion inicial.
  Computar la funcion de evaluacion de cada individuo.
  WHILE NOT Terminado DO
    BEGIN /* Producir nueva generacion */
      FOR Tamaño poblacion/2 DO
        BEGIN /*Ciclo Reproductivo */
          Seleccionar dos individuos de la anterior generacion,
          para el cruce (probabilidad de seleccion proporcional
          a la funcion de evaluacion del individuo).
          Cruzar con cierta probabilidad los dos
          individuos obteniendo dos descendientes.
          Mutar los dos descendientes con cierta probabilidad.
          Computar la funcion de evaluacion de los dos
          descendientes mutados.
          Insertar los dos descendientes mutados en la nueva generacion.
        END
      IF la poblacion ha convergido THEN
        Terminado := TRUE
    END
  END
END

```

Figura 1.8: Pseudocódigo del algoritmo genético, tomada de [41]

Las tecnologías y algoritmos mencionados anteriormente serán adaptados sobre la Red de la Fundación Nacional de Ciencia (NSFNeT, *National Science Foundation Network*), la cual es usada en el desarrollo de muchos trabajos de investigación debido a la variedad de enlaces y características reales que posee, está compuesta por 14 nodos con 21 enlaces bidireccionales, fue desplegada en Estados Unidos en el año 1986 y desde entonces ha tenido constantes actualizaciones en sus componentes y enlaces para cumplir con las demandas de flujo de tráfico. Es una red de tipo irregular ya que no se puede catalogar como una red de estrella, árbol u otro tipo de red convencional, en la figura 1.9 se muestra la red con sus nodos y distancias en kilómetros entre ellos.

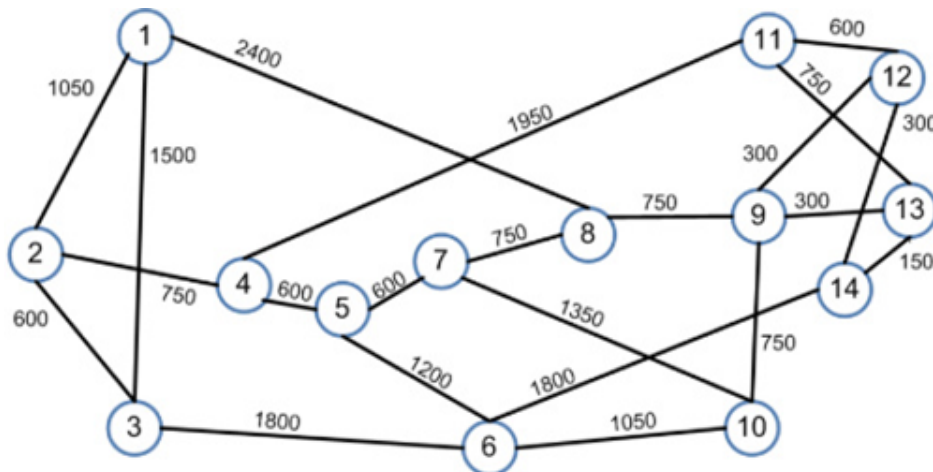


Figura 1.9: Topología NSFNeT, tomada de [43]

Capítulo 2

METODOLOGÍAS

2.1. Metodología de trabajo

Para el desarrollo del trabajo de grado fue propuesta la metodología en cascada, mostrada en la figura 2.1, ya que permite dividir el trabajo en fases secuenciales bajo la implementación de un modelo lineal, es decir, que se ejecutan una tras otra [44]. Las fases de esta metodología se describen a continuación.

- **Requerimientos:** Se analizan los conceptos que teóricos pertinentes para el desarrollo del trabajo, así como las herramientas de simulación.
- **Diseño:** Se realiza la adaptación y diseño del mecanismo de asignación dinámico de ruta basado en algoritmos genéticos, su codificación, así como el diseño del escenario de simulación.
- **Implementación:** Incorporación de los algoritmos de enrutamiento en la red, sobre la herramienta de simulación.
- **Pruebas:** Definición de escenarios, elaboración de plan de pruebas y su ejecución, siguiendo el modelo de desempeño y simulación en la herramienta.
- **Mantenimiento:** Análisis del desempeño de la red según los parámetros definidos, conclusiones de los resultados y definición de mejoras que pudieran realizarse al trabajo propuesto.

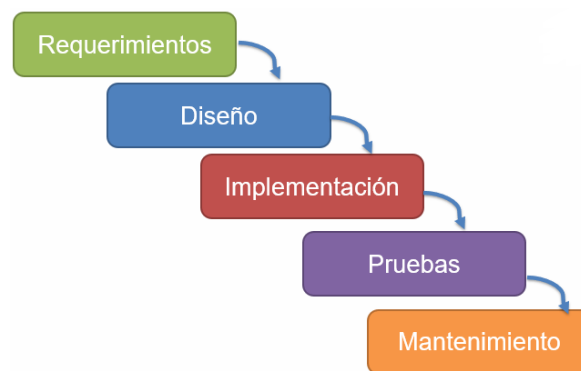


Figura 2.1: Metodología en cascada

2.2. Metodología de Simulación

Para el desarrollo del presente trabajo de grado, en el cual se busca responder a la pregunta de investigación, y cumplir los objetivos propuestos también fue necesaria la implementación de una metodología de simulación. En este sentido, la metodología usada fue Perspectiva - Práctica[45] ya que define los lineamientos de simulación y análisis del desempeño necesarios. A continuación se detallan las etapas que comprende dicha metodología.

- **Fase 1. Formulación y definición del sistema:** Es el punto de inicio del proyecto donde se define el propósito de la evaluación de la red, identificando el problema a resolver, así como las condiciones, delimitaciones y ajustes realizados en el sistema.
- **Fase 2. Selección de métricas:** Hace referencia a los parámetros empleados para medir el desempeño en la red, los cuales cuentan con rangos de valores típicos aceptados y permiten observar el comportamiento del sistema bajo determinados escenarios, donde se puede variar un parámetro y observar su influencia sobre otros.
- **Fase 3. Modelado y recolección de datos:** Una vez definidos los elementos que conforman las etapas previas se procede con el desarrollo de un modelo de desempeño y posteriormente uno de simulación, que van a permitir establecer si los comportamientos que arrojan son los esperados, así como la recolección de los datos necesarios para la simulación y su posterior análisis.
- **Fase 4. Selección ambiente de simulación, e implementación del modelo:** En esta etapa se define la herramienta de simulación que permita implementar el sistema o modelo con las características deseadas.
- **Fase 5. Validación:** Se valida el modelo de desempeño para determinar que según las métricas y condiciones establecidas, se obtenga una representación correcta del sistema, la cual corresponde a la de modelos validados previamente o modelos estadístico-analíticos que sirvan como punto de referencia.
- **Fase 6. Experimentación, análisis y presentación:** Finalmente, se ejecutan las simulaciones, se obtienen los resultados y se representan gráficamente con la ayuda de herramientas como excel. La generación de dichos resultados puede soportarse en la teoría de la estimación para obtener resultados de mayor confiabilidad.

En la definición de la metodología implementada el autor especifica que el contenido de las fases puede ser ajustado, ya que no pretende ser una metodología rígida. En este sentido la selección de la herramienta de simulación descrita en la fase 4 se realizó durante la fase 1, ya que en la definición del sistema está inmerso su uso.

Se define un algoritmo de simulación que realiza la evaluación del desempeño de la red a través de un modelo de simulación, y se muestra en la figura 2.2, adaptando las fases mencionadas anteriormente.

Aunque el desarrollo de las fases de la metodología sea de manera lineal, es posible que sea necesario regresar a fases anteriores, debido a cambios que se puedan presentar en la fase actual, relacionados con alternativas de simulación, cambios en la red o en los algoritmos que implementa. También es importante mencionar que el desarrollo del análisis de resultados y las conclusiones se presenta en los siguientes capítulos, con el objetivo de ordenar la presentación del presente trabajo investigativo.

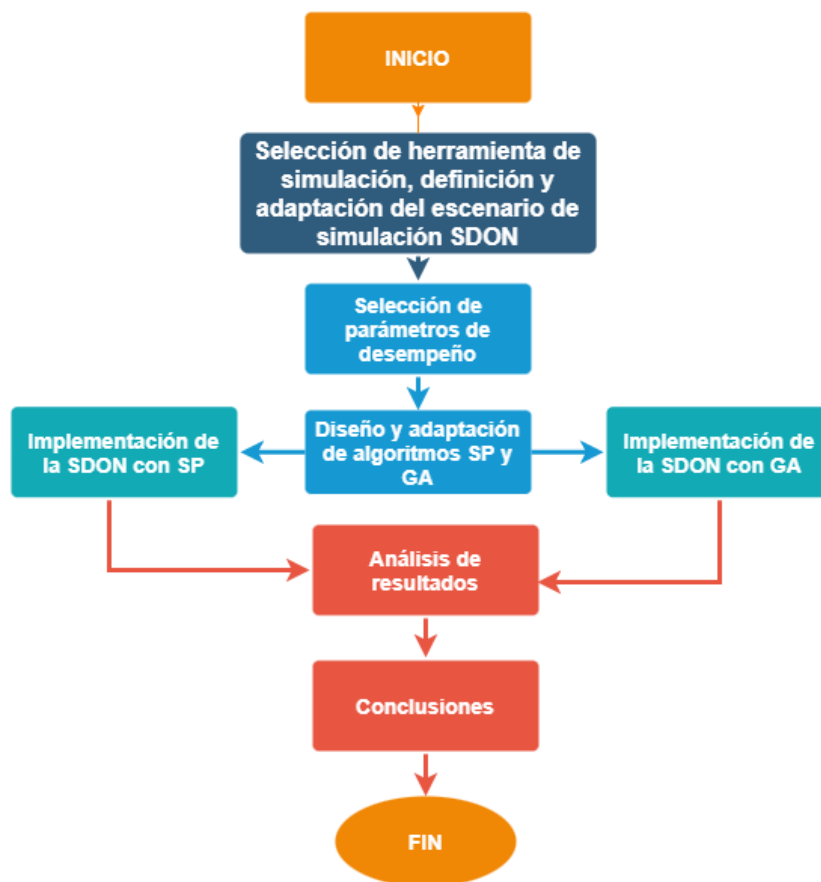


Figura 2.2: Diagrama de flujo de metodología de simulación, elaboración propia.

2.2.1. Herramientas de simulación

Se estudian algunas de las herramientas de simulación en el área de telecomunicaciones que permiten ver el comportamiento de redes según varios parámetros y características asignadas a ellas, se tiene en cuenta que este trabajo se desarrolla sobre la red NSFNeT para hacer una correcta selección de la herramienta.

- **GNS3**

Es una herramienta gratuita para emulación de redes gráficamente, como se observa en la figura 2.3, que simula el Sistema Operativo de Internet (IOS, *Inter-network Operating System*) que es el software que usan los dispositivos Cisco de forma similar a lo que se puede lograr con Vmware y Virtualbox. Funciona en las plataformas Windows, Linux y Mac, a pesar de que la descarga es gratuita, requiere de elementos como imágenes de los dispositivos y componentes de la red que no vienen por defecto y se deben descargar desde el proveedor y ser comprados.

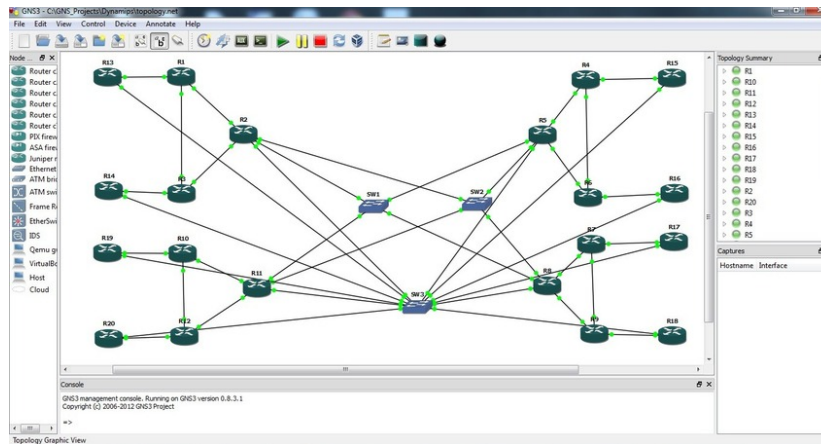


Figura 2.3: GNS3, tomada de [46]

- **NS-2**

Es una herramienta de código abierto basada en código C++ orientado a objetos para la creación de los espacios de simulación, se vincula con herramientas animadoras de red para realizar análisis visual del comportamiento de la red, no tiene interfaz gráfica y tiene lenguajes propios de modelado, lo que requiere más tiempo para hacer que funcione. En este sentido se ha desarrollado el paquete "nam", que funciona como GUI, y se muestra en la figura 2.4.

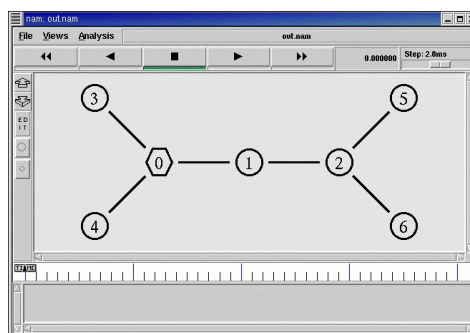


Figura 2.4: NS-2, tomada de [46]

- **NS-3**

Esta herramienta se centra en las capas del transporte y enlace del modelo OSI, es de código abierto, no hace uso del lenguaje orientado a objetos para controlarlo y se compone de 24 módulos compuestos de elementos de red y protocolos reales. Tampoco posee una GUI para animación, pero permite implementar el método NetAdmin como se observa en la figura 2.5.

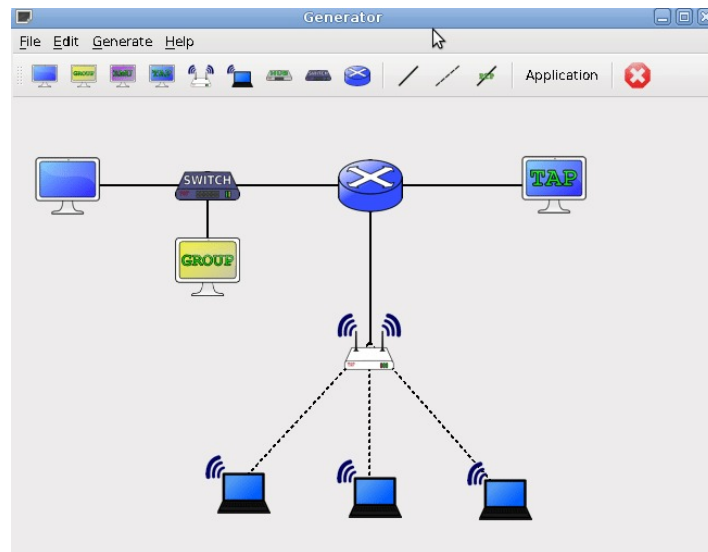


Figura 2.5: NS-3, tomada de [47]

- **OMNeT ++IDE**

Esta herramienta para plataformas Windows, Linux, Unix y Mac, se basa en módulos simples y compuestos desarrollados en lenguaje NED y programados con POO C++, permite simular redes en cualquier ambiente y habilita la simulación de cualquier red o suceso que se pueda representar por medio de eventos discretos. También ofrece el paquete de Telemetría Mejorada de Red Integrada (INET, *Integrated Network Enhanced Telemetry*) que proporciona paquetes con protocolos de internet.

La conexión entre los módulos se utiliza para generar características propias de un enlace como el retardo, velocidad de datos, tasa de error de bits, etc. Posee una interfaz gráfica interactiva llamada tkenv, mostrada en la figura 2.6 y una forma de ejecución de la simulación que permite visualizar el comportamiento de la red durante su ejecución y acciones que se generan en cada instante.

En la tabla 2.1 se presenta comparativamente para visualizar mejor las principales características de las herramientas de simulación.

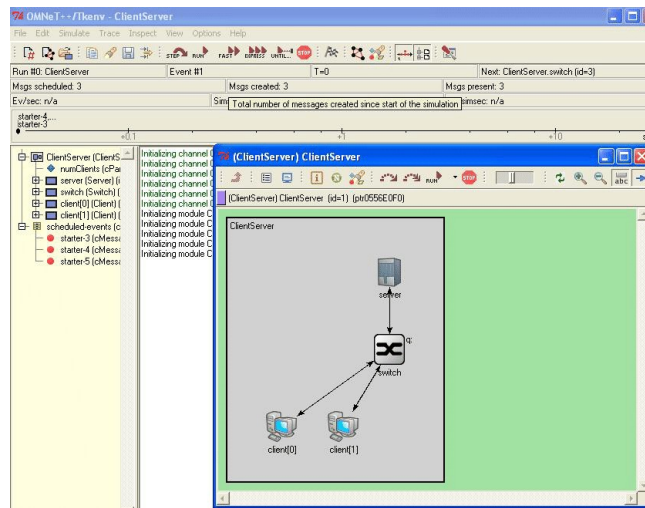


Figura 2.6: OMNeT++, tomada de [48]

Tabla 2.1: Comparación herramientas de simulación.

	GNS3	NS-2	NS-3	OMNeT++
Uso investigativo	Bajo	Alto	Medio	Alto
Tipo de licencia	Libre/Comercial	Libre	Libre	Libre
Interfaz gráfica de usuario	Alto	Bajo	Medio	Medio
Graficación de resultados	Limitada	No permite	Aceptable	Buena
Tecnologías nivel 2 y 3 que soporta	Bajo	Alto	Medio	Alto
Implementación heurística	No permite	No permite	No permite	Permite
Flexibilidad	Baja	Bajo	Baja	Alta

La herramienta OMNeT++ es la más conveniente para desarrollar el presente trabajo, debido a que permite la implementación de heurísticas, la versatilidad, facilidad de manejo y capacidad de reutilización que brinda su separación por módulos, además de contar con interfaz gráfica y la posibilidad de agregar distintas formas para la recolección de datos arrojados durante el proceso y su posterior visualización.

2.2.2. Definición del escenario de simulación

Precisar el escenario de simulación es esencial para la elaboración de las pruebas y garantizar la obtención de resultados coherentes para su posterior análisis, por lo cual la herramienta OMNeT brinda un gran número de posibilidades para la elaboración de redes y configuración de sus características, pero así mismo se requiere un gran conocimiento en su manejo e invertir tiempo en la creación de una red desde cero.

El escenario de simulación debe contar con todas las características de red mencionadas en el capítulo 1, por lo cual se emplea el un proyecto llamado “Simulador WRON” creado en la Universidad de Valladolid, que se encuentra disponible en su repositorio

como recurso abierto con fines educativos[49]. Este escenario representa un gran aporte, debido a que contiene gran parte de los elementos que permitirán el desarrollo del trabajo de grado, así como una guía para su entendimiento. Consta de dos topologías predefinidas, una red simple compuesta de seis nodos y la topología de la NSFNet que será la empleada en este trabajo.

Se tiene un nodo de control centralizado que gestiona las solicitudes de los usuarios en la red. Así mismo, la tecnología de conmutación y multiplexación corresponde a OCS y WDM respectivamente y se emplea RWA. Durante este proceso los usuarios realizan solicitudes de conexión las cuales van al nodo de control quien realiza lo necesario para procesar la solicitud y retornar una respuesta para establecer o denegar conexiones. Los usuarios no tienen información del estado de la red ni su topología, toda esta información la posee únicamente el nodo de control. Además, el simulador está construido de forma modular, donde existen módulos sencillos y compuestos que se forman a partir de los primeros, permitiendo el reuso de módulos y facilitando la construcción de la red.

Fuente de peticiones: En este caso la fuente de peticiones hace referencia al usuario que requiere tener una conexión con otro usuario y está conectado a los nodos frontera. Para la generación del tiempo entre peticiones de los usuarios se emplea la ecuación 2.1.

$$T_{Peticiones} = \frac{T_{Ligthpath}}{(N - 1) * load} \quad (2.1)$$

Donde N es el número de nodos, load es la carga de la red y $T_{Ligthpath}$ es el tiempo promedio que dura una conexión.

Las fuentes de peticiones están conectadas a los nodos frontera para tener acceso a la red como muestra la figura 2.7.

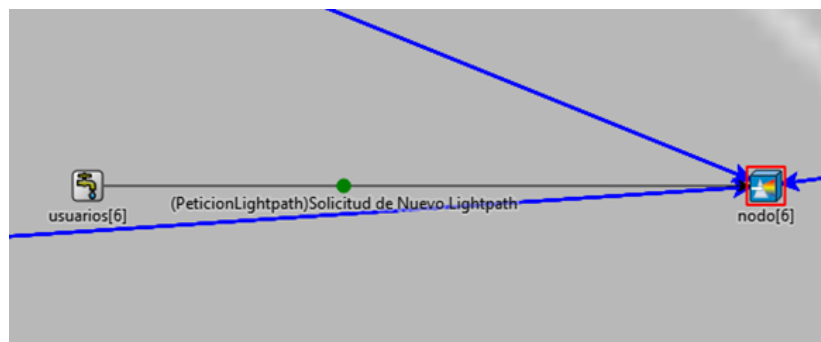


Figura 2.7: Usuario y Nodo Frontera, capturada en simulación

Nodo Frontera: El nodo frontera conecta a los usuarios con la red trocal de transporte óptico y se encarga de reenviar la información de solicitudes de conexión generadas por los usuarios al nodo de control. La petición de enrutamiento es enviada a este nodo, los nodos frontera están conectados entre sí y cada uno de ellos tiene conexión directa

con el nodo de control. Este nodo es un módulo sencillo ya que no está compuesto de más módulos internamente, que de manera directa reenvía la petición hacia el nodo de control como se muestra en la figura 2.8

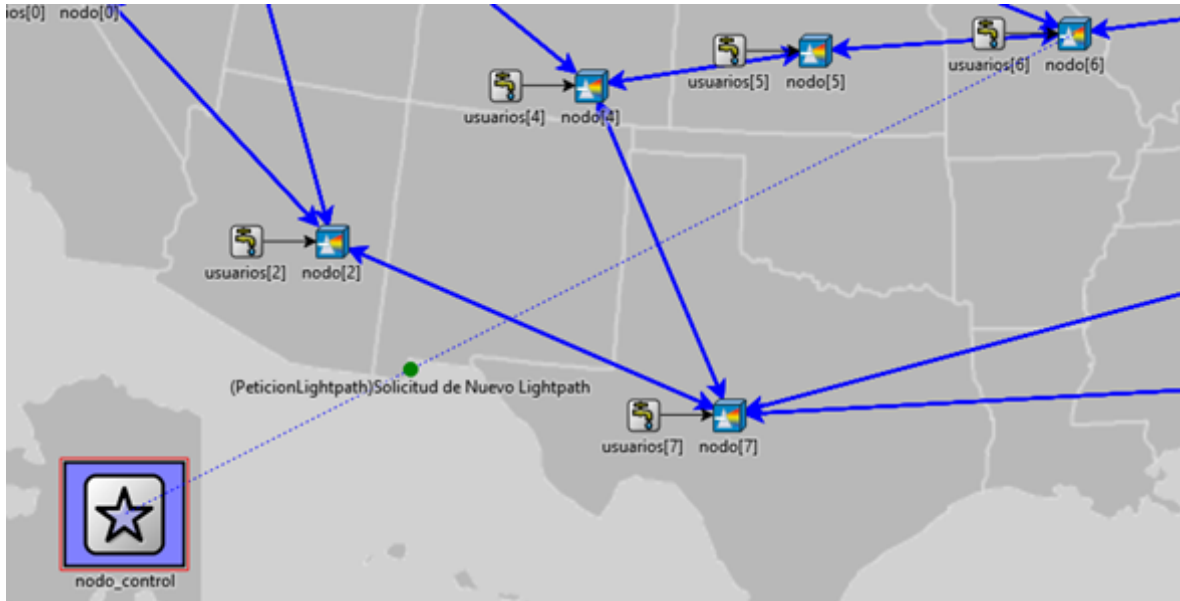


Figura 2.8: Nodo frontera, capturada en simulación

Nodo de control: Es el encargado de la gestión de la red, donde se almacena información como el estado actual, enlaces y topología, en este nodo se realizan todos los procesos de análisis y enrutamiento designados, en la interfaz de la red, el nodo de control se puede distinguir con la figura 2.9. El proceso dentro de la red comienza



Figura 2.9: Nodo de control, capturada en simulación

en la fuente de peticiones que genera una solicitud de conexión con otro usuario, la cual es enviada al nodo frontera que lo comunica con la red, posteriormente el nodo frontera reenvía la solicitud al nodo de control donde aplicará los métodos de RWA, hará los cálculos oportunos y decidirá si es posible establecer la conexión. En caso de

que sea posible, se configurarán todos los nodos del camino y se asignan los recursos necesarios durante un periodo definido por la red para el envío de la información y luego se deshabilita para liberar los recursos y ser usados en otras conexiones como se observa en la figura 2.10. El nodo de control es un módulo compuesto por los submódulos

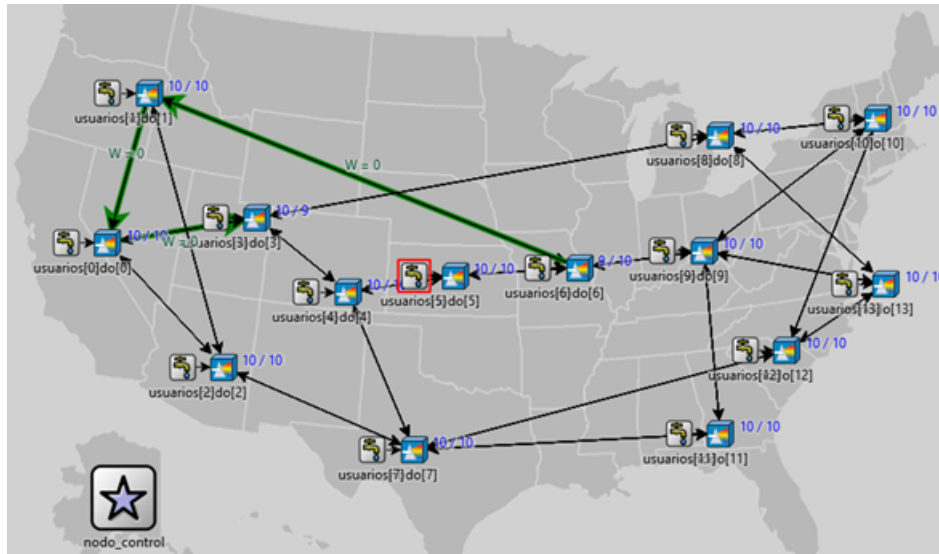


Figura 2.10: Selección de ruta, capturada en simulación

“Estado”, “WRONDinamica” y el módulo “WRONSemiEstatica” como se muestra en la figura 2.11.

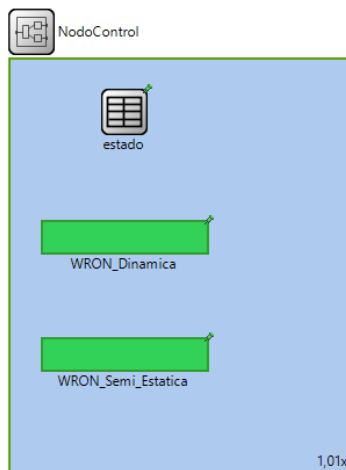


Figura 2.11: Composición del nodo de control, capturada en simulación

Módulo Estado: Dentro de este módulo se almacena la información de la red y de su estado, por lo cual los otros módulos acceden a él para obtener información del estado de los recursos de la red y actualizarlos. Los datos correspondientes a las características de la red almacenados en este módulo se encuentran en un arreglo de vectores

y matrices que contienen los datos del estado de los enlaces, con sus longitudes de onda libres, transmisores y receptores.

Módulo WRONDinámica: este módulo se encarga de establecer los recursos y liberarlos luego de cumplirse el tiempo de conexión. También, da solución al problema de asignación de recursos respectivo a cada solicitud usando el algoritmo o parámetros definidos. Está compuesto por el módulo “NodoControlDynamicWRON”, el cual se divide en tres submódulos:

- **Método RWA Dinámico:** Por defecto viene integrado con el algoritmo “AurExhaustive” diseñado para solucionar el problema de RWA en una red dinámica, el cual puede ser reemplazado por el algoritmo que se quiera implementar.
- **Método RWA Protección:** Este módulo añade protección a la red cuando tiene cargas de tráfico muy altas, las conexiones no pueden ser establecidas o se rompe un enlace.
- **Control RWA Dinámico:** En la figura 2.12 se muestra la composición interna del módulo ControlRWADinámico, que tiene una conexión directa y continua con el módulo “Estado” y el “MétodoRWAdinamico” debido a que en este módulo se establecen las conexiones, se liberan o se bloquean las solicitudes de lightpath. Este módulo modifica el estado de la red almacenado en la información del módulo “Estado” dentro de las matrices y vectores, garantizando que estén actualizados.

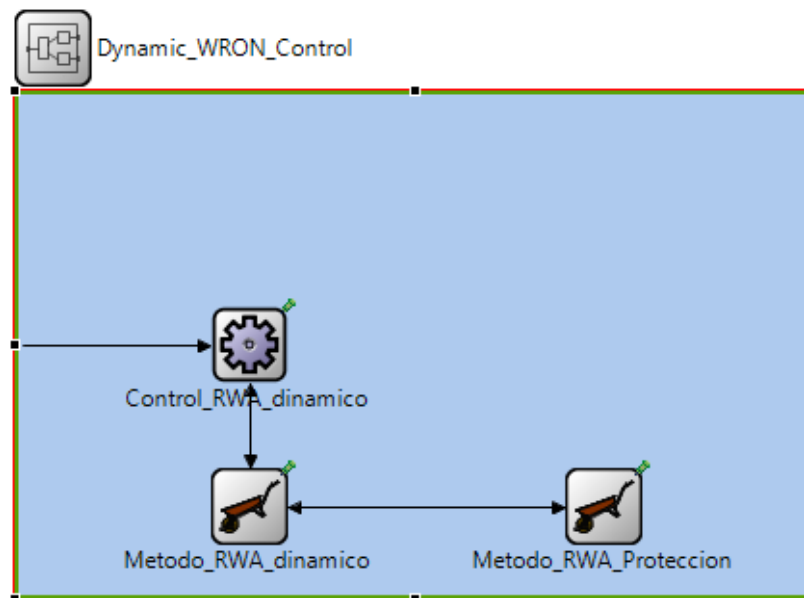


Figura 2.12: Composición Control-WRON-Dinámico, capturada en simulación

Módulo WRON Semi-Estática: Este módulo fue diseñado dentro del simulador para que trabajara con redes estáticas o semiestáticas.

Los módulos mencionados anteriormente son los que componen el diseño del simulador WRON, sin embargo, en este trabajo no se hace uso del módulo WRON-Semi-Estática ni del módulo “Metodo-RWA-Protección”, debido a que se implementa enrutamiento dinámico y no se requiere un método para longitudes de onda adicionales a las definidas, pues se busca analizar el comportamiento de la red con limitación de recursos.

Para poder cumplir a cabalidad con las características de red fueron necesarias algunas modificaciones en el simulador, que serán detalladas a continuación. La primera modificación se realizó en la versión del simulador, ya que venía por defecto en la 4.2.2 y no presentaba operabilidad con la versión más reciente 5.6.2. Estos cambios estuvieron relacionados con la forma en que se programaban los eventos discretos en ” *scheduledz* el manejo de los llamados “*namespace*” [50].

Una característica importante del escenario de simulación especificado para el presente trabajo de grado es el enrutamiento dinámico, ya que se realiza bajo demanda y difiere del enrutamiento estático en la frecuencia y forma con la que se actualizan las rutas en los nodos. Por ello se eliminó el módulo WRON Semi-Estática y su funcionalidad. Así mismo, fue eliminado el módulo de protección, que se encontraba dentro de WRON Dinámica, cuya funcionalidad era habilitar una longitud de onda especial en los enlaces totalmente ocupados.

El módulo estado poseía funciones relacionadas al módulo WRON Semi-Estática y el de protección, por lo que se redefinió para guardar la información necesaria del enrutamiento dinámico y se creó una tabla de enlaces, una de su estado y otra del tiempo de ocupación.

Para poder implementar WDM, se analizó la NSFNeT y se observó un solo enlace bidireccional entre los nodos frontera, por lo cual no sería posible multiplexar la fibra ni reenviar la información a través de la red, entonces se agregaron nuevas conexiones entre los dispositivos de reenvío, donde cada una representa una longitud de onda. Seguidamente, se creó una matriz en el módulo estado mostrada en la figura 2.13, donde se tienen 10 longitudes de onda por enlace cuya disponibilidad se muestra con 0 y, se encarga de simular la multiplexación, es decir, cada fila representa el número de enlace correspondiente y ésta posee n columnas que representan las longitudes.

Para verificar como se realizaba el reenvío de información a través de la red, se ejecutó el simulador y en el momento en que se establece la conexión, se notó que no se estaba reenviando la información desde el origen al destino, por lo que fue necesario añadir dicha funcionalidad a los nodos frontera. En consecuencia, se creó la clase paquete que se muestra en la figura 2.14 que especifica el nodo de origen, destino, el vector ruta y la puerta de salida en cada nodo. Una de las adaptaciones más importantes fue la creación de las clases que contienen al GA y SP junto con el algoritmo de asignación de longitud de onda FF. El simulador implementaba por defecto un método RWA llama-

ESTADO DE LOS CANALES										
	0	1	2	3	4	5	6	7	8	9
0->1	0	0	0	0	0	0	0	0	0	0
0->2	0	0	0	0	0	0	0	0	0	0
0->3	0	0	0	0	0	0	0	0	0	0
1->0	0	0	0	0	0	0	0	0	0	0
1->2	0	0	0	0	0	0	0	0	0	0
1->6	0	0	0	0	0	0	0	0	0	0
2->0	0	0	0	0	0	0	0	0	0	0
2->1	0	0	0	0	0	0	0	0	0	0
2->7	0	0	0	0	0	0	0	0	0	0
3->0	0	0	0	0	0	0	0	0	0	0
3->4	0	0	0	0	0	0	0	0	0	0
3->8	0	0	0	0	0	0	0	0	0	0
4->3	0	0	0	0	0	0	0	0	0	0
4->5	0	0	0	0	0	0	0	0	0	0
4->7	0	0	0	0	0	0	0	0	0	0
5->4	0	0	0	0	0	0	0	0	0	0
5->6	0	0	0	0	0	0	0	0	0	0
6->1	0	0	0	0	0	0	0	0	0	0
6->5	0	0	0	0	0	0	0	0	0	0
6->9	0	0	0	0	0	0	0	0	0	0
7->2	0	0	0	0	0	0	0	0	0	0
7->4	0	0	0	0	0	0	0	0	0	0
7->12	0	0	0	0	0	0	0	0	0	0
7->11	0	0	0	0	0	0	0	0	0	0
8->3	0	0	0	0	0	0	0	0	0	0
8->10	0	0	0	0	0	0	0	0	0	0
8->13	0	0	0	0	0	0	0	0	0	0
9->6	0	0	0	0	0	0	0	0	0	0
9->10	0	0	0	0	0	0	0	0	0	0
9->11	0	0	0	0	0	0	0	0	0	0
9->13	0	0	0	0	0	0	0	0	0	0
10->8	0	0	0	0	0	0	0	0	0	0
10->9	0	0	0	0	0	0	0	0	0	0
10->12	0	0	0	0	0	0	0	0	0	0

Figura 2.13: Tabla de enlaces, capturada en simulación.

do “Aur-Exhaustive”, que para calcular la ruta entre origen y destino creaba grafos y asignaba una longitud de onda específica para cada uno. Una vez se realizaron las adaptaciones anteriores se implementaron los métodos estadísticos ofrecidos por OMNeT en el nodo de frontera, para obtener parámetros como el retardo extremo a extremo, la probabilidad de bloqueo de peticiones y el throughput que serán usados en el análisis de desempeño de la SDON.

```

packet PacketOptico
{
    int dirOrigen;
    int dirDestino;
    int puertos_salida[];
    int rutaEnlaces[];
    int long_onda[];
}

```

Figura 2.14: Atributos de la información reenviada por la red, capturada en simulación.

2.2.3. Parámetros de desempeño

Para la evaluación de la SDON deben implementarse métricas que midan el rendimiento en diferentes niveles como tiempo, capacidad y respuesta ante fallos o congestión. Se mide el desempeño en función del retardo extremo a extremo y la probabilidad de bloqueo de peticiones. Adicionalmente se mide el throughput promedio de la red para analizar a detalle la transferencia de datos efectiva y su correlación con el retardo extremo a extremo.

- **Probabilidad de bloqueo**

Se determina a partir de relacionar las peticiones totales y la cantidad de solicitudes rechazadas de acuerdo con la ecuación 2.2.

$$Pb = \frac{P_{peticionesrechazadas}}{P_{peticionestotales}} \quad (2.2)$$

- **Retardo extremo a extremo**

Se determina a partir de relacionar las peticiones totales y la cantidad de solicitudes rechazadas de acuerdo con la ecuación 2.3.

$$R_{Total} = R_{transmisión} + R_{procesamiento} + R_{Propagación} \quad (2.3)$$

Donde el retardo de procesamiento ($R_{Procesamiento}$) es la suma del tiempo que toma cada nodo frontera en procesar los datos y reenviarlos a los siguientes nodos, el retardo de propagación ($R_{Propagación}$) es el tiempo que se tarda en viajar la información por la fibra óptica desde el origen hasta el destino y el retardo de transmisión ($R_{Transmisión}$) es el tiempo que invierte el usuario emisor en poner los datos en el enlace y depende de la velocidad de transmisión.

- **Throughput**

Se determina a partir de la razón entre los datos a transmitir y el retardo extremo a extremo. Según[51], la evaluación de este parámetro puede definirse dependiendo del objeto de interés en particular. En este caso se mide sobre el valor promedio del retardo extremo a extremo de todos los trayectos establecidos en la red, por lo que ésta medida hace referencia a los Mbps que atraviesan un trayecto promedio para una carga de tráfico determinada, según la ecuación 2.4.

$$T_{Throughput} = \frac{Datos}{RetardoExtremo a Extremo} \quad (2.4)$$

2.2.4. Diseño del algoritmo adaptado.

Algoritmo de asignación de longitud de onda First Fit

El algoritmo se usa en la selección de la longitud de onda que será asignada para atender la petición de conexión, se tomará la primera disponible en orden ascendente, siguiendo una numeración establecida para cada una en los enlaces. Este algoritmo es escogido

debido a su bajo costo computacional, eficiencia y acople con la asignación de recursos.

Para el diseño del algoritmo FF dentro del módulo “Método-RWA-Protección”, primero se consultó en el módulo estado la disponibilidad actual de longitudes de onda por enlace y se entregó dicha información al algoritmo de enrutamiento correspondiente el cual devuelve la ruta de origen a destino. Ya conociendo los enlaces que componen la ruta, el algoritmo FF procede a asignar la primera longitud de onda disponible. Es pertinente recordar que este algoritmo se implementa con y sin el método GA, es decir, con un algoritmo estándar, SP y el propuesto GA.

Algoritmo de enrutamiento estándar: Shortest Path

Luego de recibir la información del estado de los enlaces por parte del algoritmo FF, se procede a calcular la ruta más corta hacia el destino por medio de la función “*calculateUnweightedSingleShortestPathsTo*” que viene integrada con la clase `netTopology` de OMNet. En caso de que la ruta exista, se asigna junto a las longitudes de onda correspondientes y se envía un mensaje de regreso al módulo “Control-RWA-Dinámico” para que se comunique con el nodo de origen, pueda iniciarse la conexión y se deshabiliten las longitudes de onda seleccionadas, durante el tiempo que dure la transmisión. En el caso de que la ruta no exista se habilita una bandera que indica “conexión rechazada” y se envía un mensaje al nodo control para que rechace la conexión.

En la figura 2.15 se representa mediante un diagrama de flujo, los pasos para el enrutamiento y asignación de longitud de onda al combinar los algoritmos SP y FF. Así mismo en la figura 2.16, se presenta la codificación del algoritmo en la herramienta de simulación OMNeT.

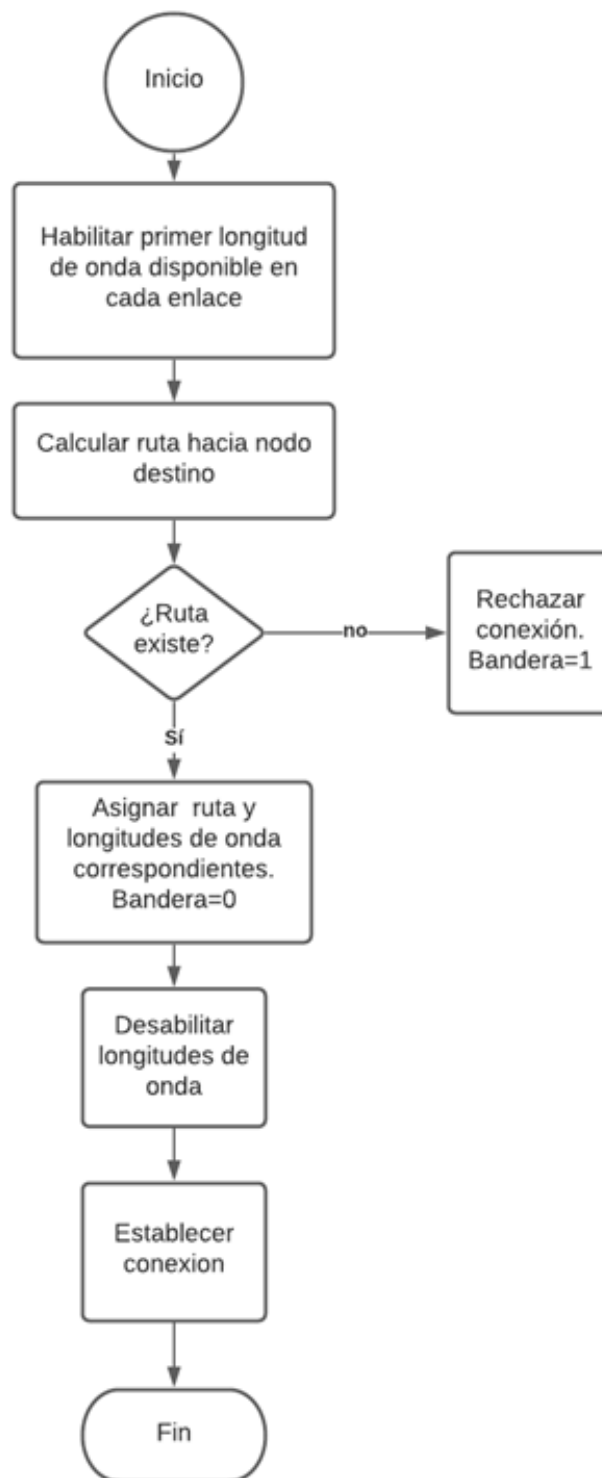


Figura 2.15: Diagrama de flujo algoritmo SP, elaboración propia

```

for(i=0;i<num_enlaces/num_lng_ond;i++){
for(w=0;w<num_lng_ond;w++){
    if(modulo_estado->checkChannelAvailability(i,w)==true){
        enlaces_en_topologia[num_lng_ond*i+w]->enable();
        break;
    }
}
}
modulo_auxiliar = red_fisica->getSubmodule("nodo", origen )assert( modulo_auxiliar != NULL );
origennod = netTopology.getNodeFor( modulo_auxiliar );
assert( origennod != NULL );

modulo_auxiliar = red_fisica->getSubmodule("nodo", destino );
assert( modulo_auxiliar != NULL );
destinonod = netTopology.getNodeFor( modulo_auxiliar );
assert( destinonod != NULL );

netTopology.calculateUnweightedSingleShortestPathsTo( destinonod );

longitud = origennod->getDistanceToTarget();

if(origennod->getDistanceToTarget()!=HUGE_VAL ){
path.resize( num_nodos );
j=0;
while(origennod!=destinonod){
path[j] = origennod->getPath(0);
j++;
    origennod = origennod->getPath(0)-> getRemoteNode();
}
}
else{
    bandera=1;
}
}

```

Figura 2.16: Codificación del algoritmo Shortest Path en OMNeT

Algoritmo de enrutamiento Genético

El diseño del GA se realizó al adaptar el algoritmo desarrollado en [34], el cual se implementó sobre una red OBS/WDM distribuida. En la figura 2.17 se observa que al igual que el algoritmo SP, el GA debe esperar a recibir la información del estado de los enlaces por parte del algoritmo FF, para iniciar su funcionamiento. El primer paso en la aplicación del GA es generar la población inicial de individuos de forma aleatoria, en los cuales no se repiten nodos y tienen en común un nodo de origen y destino.

El cálculo de ruta se realiza con la información de los enlaces almacenada en el módulo estado, en donde el algoritmo consulta los nodos adyacentes al origen y se escoge uno de forma aleatoria. Luego sobre el nodo elegido, se consultan de nuevo los nodos adyacentes, se escoge uno aleatoriamente sin tomar en cuenta el nodo anterior, así sucesivamente hasta llegar al nodo de destino, y en caso de no ser encontrado se inicia de nuevo el proceso. El número de rutas iniciales a calcular se establece en esta etapa y en caso de no obtenerse una población inicial luego de un número determinado de interacciones que por diseño es aproximadamente 8000, se rechaza la solicitud de conexión, indicando al nodo de control por medio de un mensaje con un atributo bandera establecido en "1" que la ruta no pudo ser establecida.[52].

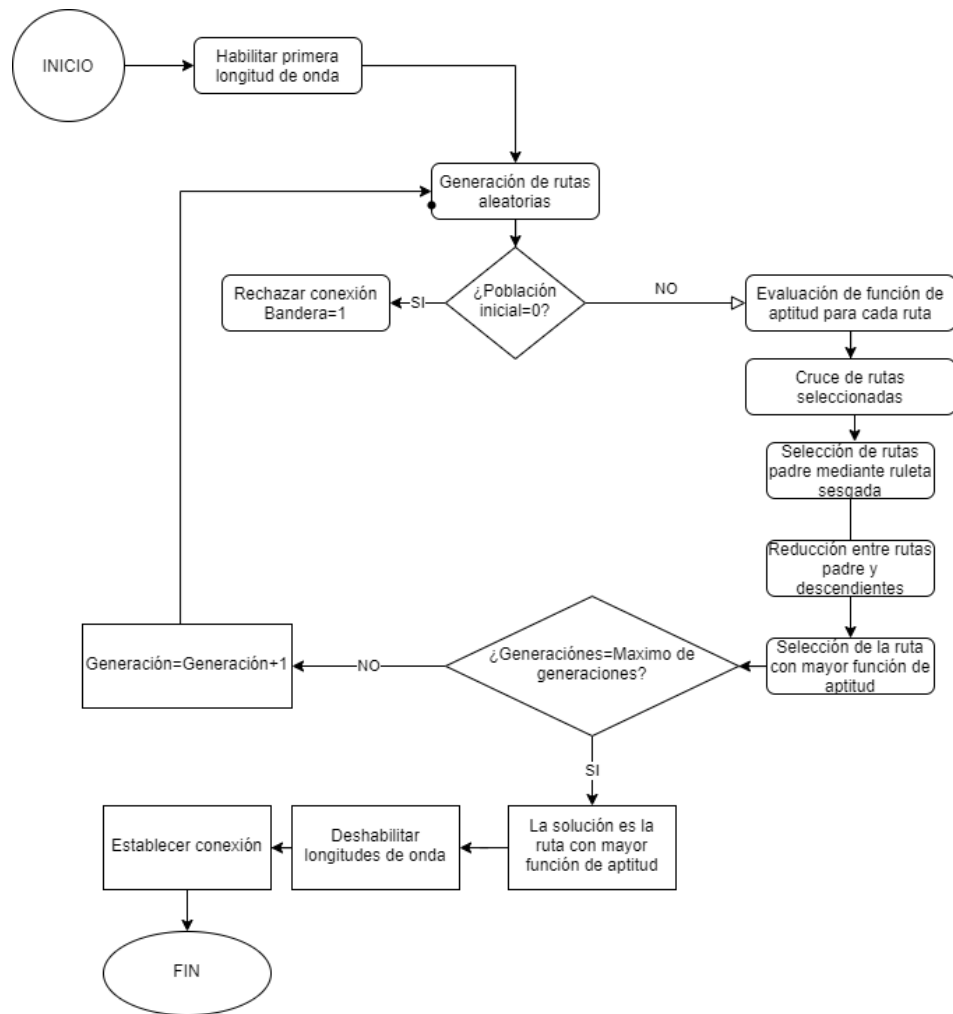


Figura 2.17: Diagrama de flujo algoritmo GA, elaboración propia

La evaluación de la bondad de las rutas iniciales le corresponde a la función de aptitud, quien les otorga un puntaje numérico con el fin de medir sus características. Se introducen los términos “costos de ruta”, que permiten examinar el estado de la red dependiendo diversos factores[15].

Se implementa el C1 que permite calcular el costo en función de la cantidad de saltos y por el número de longitudes de onda disponibles en cada nodo y se obtiene con la fórmula 2.5[53].

$$C1 = S + a \sum_{i=1}^s (L_i - L_f * i) \quad (2.5)$$

Donde S representa el número de saltos, L_i el número de longitudes de onda totales en la ruta, L_f el número de longitudes libres y a representa un parámetro de diseño entre 0 y 1 que varía dependiendo del número de longitudes de onda en el enlace.

El costo C2 simboliza la congestión en un nodo a partir de los enlaces conectados a él, si tiene más conexiones con otros nodos, su costo aumenta y se obtiene con la fórmula 2.6

$$C2 = \sum_{i=1}^n E_i \quad (2.6)$$

La sumatoria de los enlaces de cada nodo que compone la ruta es el valor del costo C2, donde E_i es el número de conexiones de cada nodo.

Las funciones de aptitud son inversamente proporcionales al valor de los costos, por lo cual se describen en la ecuación 2.7

$$FA1 = \frac{1}{S + a \sum_{i=1}^s (L_i - L_f * i)} \quad (2.7)$$

$$FA2 = \frac{1}{\sum_{i=1}^n E_i} \quad (2.8)$$

El valor máximo que podrá tener cada función de aptitud es 1, por tanto, será $FA1+FA2$ igual a 2 el valor máximo que podrá tener una ruta, y sucede cuando esta tiene un solo salto, todas las longitudes de onda disponibles y presenta solo una conexión hacia otros nodos. El número de longitudes de onda libre en cada enlace y su congestión se extraen del módulo estado[54]. En la figura 2.18 se muestra la codificación correspondiente a la función de aptitud que otorga el valor a las rutas. Seguidamente de la aplicación de la función de aptitud, se seleccionan las rutas padres por medio del método de ruleta sesgada[41]. Según se observa en la figura 2.19, a cada individuo se le asigna un porcentaje de la ruleta normalizado a uno, que le otorgará una mayor porción al individuo con mayor función de aptitud y en consecuencia aumenta su probabilidad de ser escogido como padre. Esto hará que la nueva generación tenga diversidad y puedan encontrarse soluciones aproximadas a óptimos globales y no solo a locales. Para asignar las porciones de la ruleta se divide el valor de aptitud de cada ruta, entre la suma del total de aptitudes de la población inicial[55].

Luego de seleccionar las rutas candidatas a padres, se descartan aquellas que solo tengan dos nodos, ya que no tendría sentido aplicarles el operador genético de cruce. Las rutas que se seleccionen finalmente para crear la nueva generación se analizan en búsqueda de un elemento en común que no sea la primera ni la última posición y será en éste donde se aplicará el cruce, que dará lugar a dos rutas hijas como se muestra en la figura 1.5 del capítulo 1. Tanto las rutas padres como las hijas se organizan en un mapa que posteriormente es reducido y se escoge la ruta de mayor aptitud. Después se verifica que se haya cumplido con el número máximo de generaciones, y en caso de no ser así, se realiza de nuevo el proceso de generación de rutas aleatorias. Luego de aplicar los pasos explicados anteriormente se obtiene la ruta con mayor valor de aptitud que será guardada en un vector junto con los mejores individuos en cada generación.

```

std::map<std::vector<int>,double>Genetico::fAptitud(std::map<std::vector<int>,
int>rutas)
{
    std::map<std::vector<int>, double> rutasAptitud;
    cModule *c_modulo_estado = getParentModule()->getParentModule()->getParentModule()
->getSubmodule("estado");
    Estado *modulo_estado = check_and_cast<Estado *>(c_modulo_estado);
    for(std::map<std::vector<int>, int>::iterator it1= rutas.begin(); it1!=rutas.end();
++it1){
        int saltos=(it1->first).size()-1;
        int cont_onda_libre=0;
        int pos=0;
        int id_enlace;
        int totalw= num_lng_ond*saltos;
        double alpha= 1/(double)(totalw-1);
        double costo1;
        double aptitud1;

        while(pos<saltos){
            for(int p=0; p<num_enlaces; p++){
                if(modulo_estado->tabla_enlaces[p][1]-2==(it1->first)[pos] &&
modulo_estado->tabla_enlaces[p][2]-2==(it1->first)[pos+1] ){
                    id_enlace=modulo_estado->tabla_enlaces[p][0];
                    pos++;
                }
            }
            for(int l=0; l<num_lng_ond;l++){
                if(modulo_estado->estado[int(id_enlace/num_lng_ond)][l]==0){
                    cont_onda_libre++;
                }
            }
        }
        costo1= saltos+ alpha*(totalw-cont_onda_libre);
        aptitud1=1/costo1;
        rutasAptitud[(it1->first)]=aptitud1;
    }
    return rutasAptitud;
}

```

Figura 2.18: Codificación para la función de aptitud

Al finalizar el proceso de obtención de las mejores rutas de cada generación se escoge la solución con la mayor función de aptitud entre todas y se asigna la ruta al igual que las longitudes de onda necesarias. A continuación, se envía un mensaje de regreso al módulo “Control-RWA-Dinámico” para que se comunique con el nodo de origen, pueda iniciarse la conexión y se deshabiliten las longitudes de onda seleccionadas, durante el tiempo que dure la transmisión.

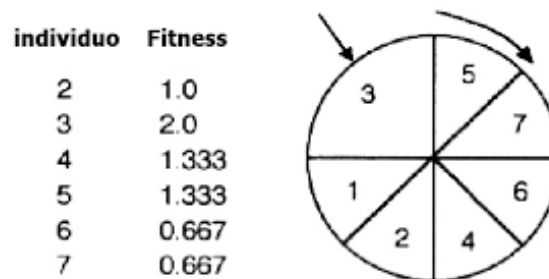


Figura 2.19: Metodo ruleta sesgada.

2.2.5. Descripción de casos de uso

Con el fin de obtener una variedad de resultados que evalúen de forma correcta el objetivo del trabajo, se definen los casos de uso en dos grupos principales.

- **Escenario 1:** Este caso de estudio hace referencia al comportamiento de la red usando el método de referencia SP donde se encuentran dos subcasos en los que varía el número de longitudes de onda totales por enlace (5 y 10), e internamente cada uno de ellos se varia la velocidad de transmisión de 1.25Gbps a 2.5Gbps.
- **Escenario 2:** En el segundo caso hace uso exclusivamente del GA y está dividido en dos subcasos al igual que en el “caso 1” las longitudes de onda y la velocidad de transmisión. Para cada valor de velocidad de transmisión se varía el número de generaciones que emplea el GA antes de obtener una ruta óptima, para este caso de 1 a 4, con el propósito de analizar la variación en el tiempo de computación y calidad de la ruta obtenida.

Para obtener la probabilidad de bloqueo de peticiones se registran las solicitudes de conexión totales y las rechazadas, el retardo extremo a extremo se obtiene al promediar el retardo de cada conexión establecida, y se captura cuando llega la información al destino.

A continuación se ilustran los casos de estudio en la tabla 2.2. donde se observa que para cada subcaso de estudio se simulan en total 5 cargas, que corresponden al tráfico erlang usado en telefonía para calcular el grado o niveles de servicio.

Tabla 2.2: Ilustración casos de estudio

Escenarios de Simulación							
Topología de red NSFNet-SDON							
Escenario 1				Escenario 2			
Implementando el algoritmo SP				Implementando el GA			
Casos de simulación							
Caso 1		Caso 2		Caso 1		Caso 2	
5 longitudes de onda		10 longitudes de onda		5 longitudes de onda		10 longitudes de onda	
Subcaso a	Subcaso b	Subcaso a	Subcaso b	Subcaso a	Subcaso b	Subcaso a	Subcaso b
Velocidad de 1.25Gbps	Velocidad de 2.5Gbps	Velocidad de 1.25Gbps	Velocidad de 2.5Gbps	Velocidad de 1.25Gbps	Velocidad de 2.5Gbps	Velocidad de 1.25Gbps	Velocidad de 2.5Gbps
				1 y 4 Generaciones			
Cada caso de estudio a y b se simularán para cada una de las cargas de tráfico (0.2, 0.4, 0.6, 0.8, 1.0)							

float

Capítulo 3

IMPLEMENTACIÓN Y RESULTADOS

Continuando el proceso metodológico y de acuerdo con los casos de estudio definidos, se realizó la implementación de la SDON para el entorno de simulación OMNeT, con el objetivo de obtener los resultados y evaluar la red con y sin el algoritmo de asignación dinámica de ruta, midiendo el desempeño en función del retardo extremo a extremo, la probabilidad de bloqueo de peticiones y el throughput.

Debido a que el OMNeT es un simulador de eventos discretos, el tiempo de simulación es distinto al tiempo real, el cual es directamente proporcional a la complejidad de la red, es decir, a la cantidad de módulos que posea, y a los procesos de cálculo subyacentes en su mayoría relacionados con el tipo de algoritmo de cálculo de ruta que se utilice[50][49].

Las simulaciones realizadas permiten validar el modelo de red implementado, por lo cual fue necesario el uso de técnicas estadísticas que permitieran analizar el comportamiento de la red cuando alcanzaba el estado estacionario[56]. Según la figura 3.1, cuando comienza el proceso de simulación la red se encuentra en estado transitorio, ya que las muestras recogidas de los parámetros de desempeño varían en el tiempo, es decir, presentan una gran varianza respecto a la media, y en consecuencia se obtienen resultados con un mayor margen de error. Luego se presenta un tiempo estacionario donde se recomienda tomar las muestras ya que idealmente son invariantes en el tiempo. Finalmente se presenta un tiempo transitorio, el cual nos indica que no es recomendable someter la red a grandes tiempos de simulación, ya que resultaría en datos inestables, un gasto innecesario de tiempo y recursos computacionales [57].

La rapidez con la que se alcanza el tiempo estacionario y transcurre el tiempo real depende en mayor medida de la complejidad de la red, del tráfico que la atraviesa y de la limitación de los recursos, por lo cual se tomaron muestras en distintos tiempos de simulación y finalmente para 2s, 3s y 4s, en ambos escenarios se observó que los resultados se encontraban dentro del régimen estacionario. Los resultados de dichas muestras se promediaron para obtener los parámetros de desempeño finales.

En consecuencia se obtuvieron 60 simulaciones con duración en tiempo real de una hora y otras con duración de hasta 8 horas, estas últimas corresponden al algoritmo genético. El total de simulaciones, que tomaron alrededor de 105 horas se llevaron a cabo utilizando 3 equipos con altas especificaciones, detallados en la tabla 3.1.

Tabla 3.1: Especificaciones técnicas de los equipos empleados para las simulaciones

Equipo	Procesador	Memoria RAM	Disco Duro	Sistema Operativo
Toshiba Satélite L45	Intel core i5	8 GB	1TB	W10 x64bits
Asus Vivo-Book	Ryzen 5	8 GB	500 GB	W10 x64bits
Asus X555LN	Intel Core i7	8GB	500GB	W10 x64bits

Parámetros de simulación: En la tabla 3.2 se presentan los valores de los parámetros necesarios para llevar a cabo cada una de las simulaciones, donde comúnmente la elección de la cantidad longitudes de onda se basa en la canalización $2n$ usada en los sistemas de comunicaciones ópticos comerciales [58], pero en este caso se usaron 5 y 10 longitudes para visualizar de una mejor forma la variación de los parámetros de desempeño con el GA. Las velocidades de transmisión permitirán observar el comportamiento de la red, especialmente en parámetros de desempeño como el retardo extremo a extremo. La duración media del lightpath corresponde a un valor promedio que se tomó al ejecutar un par de simulaciones de prueba a ambas simulaciones en cada algoritmo y la variación de la carga representa un tráfico Erlang normalizado que varía desde 0.2 a 1 en intervalos de 0.2.

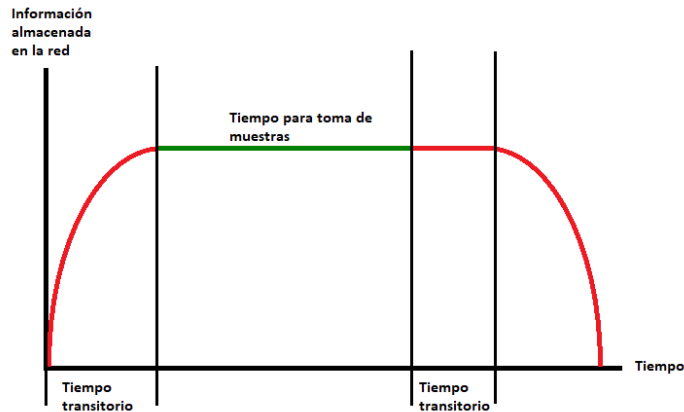


Figura 3.1: Estado de tiempos de la red.

Tabla 3.2: Parámetros fijos y variables de las simulaciones

Parámetro	Variable	Valores
Longitudes de onda	num_lng_ond	5 y 10
Velocidad de transmisión	wavelength_bit_rate	1,25 Gbps y 2,5 Gbps
Generaciones	generaciones	1 y 4
Duración media de un lightpath	duracion_media_lightpath_dinamico	0,012 Segundos
Carga	dynamic_load	0,2 a 1.0 Erlang
Datos	packetLength	1472Bytes

3.1. Implementación de la SDON sin el algoritmo adaptado

A continuación, se presentan los detalles de cada caso de estudio con el algoritmo SP, así como los resultados de la probabilidad de bloqueo, retardo extremo a extremo y throughput.

3.1.1. Caso 1: 5 longitudes de onda

Se configura el parámetro de las longitudes de onda a 5 trayectos en el archivo om-net.ini, lo cual significa que por cada enlace se tendrán 5 longitudes de onda.

Velocidad de transmisión de 1.25 Gbps: Se configura el parámetro de velocidad de transmisión a 1.25Gbps y se ejecuta una simulación para cada carga de tráfico 0.2, 0.4, 0.6, 0.8, 1.0 respectivamente. Esto quiere decir que, en cada escenario, por enlace se tiene una velocidad de transmisión de 1.25Gbps, se obtienen valores de probabilidad de bloqueo además del retardo extremo a extremo. La tabla 3.3 muestra los resultados obtenidos.

Tabla 3.3: Resultados caso 1, velocidad 1.25Gbps.

Carga (E)	Probabilidad de bloqueo	Retardo extremo a extremo (seg)	Throughput (Mbps)
1	0,614323	0,009082	1,30
0,8	0,554165	0,009454	1,25
0,6	0,462324	0,009881	1,19
0,4	0,282682	0,010180	1,16
0,2	0,004360	0,009129	1,29
Promedio	0,383571	0,009545	1,24

Según la ecuación 2.4 el Throughput (Mbps) para la carga 1.0, puede calcularse de

la siguiente manera:

$$\text{Throughput} = \frac{1472 * 8\text{bits}}{0,009082\text{seg}} = 1296630\text{bps} \approx 1,30\text{Mbps} \quad (3.1)$$

Donde el valor $1472 * 8 \text{ bits}$ corresponde al tamaño de la información por conexión y $0,009082 \text{ seg}$ corresponde al retardo extremo a extremo.

Velocidad de transmisión de 2.5 Gbps: En este escenario se aumenta la velocidad de 1.25Gbps a 2.5Gbps, y se ejecuta una simulación para cada carga de tráfico 0.2, 0.4, 0.6, 0.8, 1.0 respectivamente. Lo significa que, en cada escenario, por enlace se tiene una velocidad de transmisión de 2.5Gbps, y se obtienen valores de probabilidad de bloqueo además del retardo extremo a extremo. La tabla 3.4 muestra los resultados arrojados por la simulación.

Tabla 3.4: Resultados caso 1, velocidad 2.5Gbps.

Carga (E)	Probabilidad de bloqueo	Retardo extremo a extremo (seg)	Throughput (Mbps)
1	0,613135	0,009081	1,30
0,8	0,551211	0,009419	1,25
0,6	0,460273	0,009819	1,20
0,4	0,281478	0,010167	1,16
0,2	0,004355	0,009118	1,29
Promedio	0,382090	0,009521	1,24

3.1.2. Caso 2: 10 longitudes de onda

Velocidad de transmisión de 1.25 Gbps: Se configura el parámetro de velocidad de transmisión a 1.25Gbps y se ejecuta este escenario 5 veces con las cargas de tráfico 0.2, 0.4, 0.6, 0.8, 1.0 respectivamente. Esto quiere decir que, en cada escenario, por enlace se tiene una velocidad de transmisión de 1.25Gbps, se obtienen valores de probabilidad de bloqueo además del retardo extremo a extremo. La tabla 3.5 muestra los resultados obtenidos.

Tabla 3.5: Resultados caso 2, velocidad 1.25Gbps.

Carga (E)	Probabilidad de bloqueo	Retardo extremo a extremo (seg)	Throughput (Mbps)
1	0,328672	0,009839	1,20
0,8	0,211592	0,010010	1,18
0,6	0,030640	0,009380	1,26
0,4	0,000085	0,008894	1,32
0,2	0	0,008876	1,33
Promedio	0,114198	0,009400	1,26

Velocidad de transmisión de 2.5 Gbps: En este escenario se aumenta la velocidad de 1.25Gbps a 2.5Gbps, y se ejecuta una simulación para cada carga de tráfico

0.2, 0.4, 0.6, 0.8, 1.0 respectivamente. Por lo cual, se tiene una velocidad de transmisión de 2.5Gbps en los enlaces de cada escenario, y se obtienen valores de probabilidad de bloqueo además del retardo extremo a extremo. La tabla 3.6 muestra los resultados.

Tabla 3.6: Resultados caso 2, velocidad 2.5Gbps.

Carga (E)	Probabilidad de bloqueo	Retardo extremo a extremo (seg)	Throughput (Mbps)
1	0,326016	0,009857	1,19
0,8	0,209453	0,009995	1,18
0,6	0,031489	0,009375	1,26
0,4	0,000065	0,008889	1,32
0,2	0	0,008860	1,33
Promedio	0,11340	0,009400	1,26

3.2. Implementación de la SDON con algoritmo GA

En este caso se implementa el método GA y los diferentes subcasos de simulación.

3.2.1. Caso 1: 5 longitudes de onda

En el archivo omnetpp.ini se configura el parámetro longitudes de onda a 5 trayectos, lo cual indica que se tienen 5 longitudes de onda por enlace.

Velocidad de transmisión de 1.25 Gbps: Este caso incluye la variación del número de generaciones de 1 a 4, lo que implica que el algoritmo genético se iterará con el objetivo de incrementar la calidad de la ruta que se obtenga de origen a destino. Se configura el parámetro de velocidad de transmisión a 1.25Gbps y se ejecuta este escenario 5 veces con las cargas de tráfico 0.2, 0.4, 0.6, 0.8, 1.0 respectivamente. Esto quiere decir que, en cada escenario, por enlace se tiene una velocidad de transmisión de 1.25Gbps, y para las generaciones se tienen en total 10 ejecuciones. En las tablas 3.7 y 3.8 se muestran los resultados.

Tabla 3.7: Resultados caso 1, velocidad 1.25Gbps, 1 Generación

Carga (E)	Probabilidad de bloqueo	Retardo extremo a extremo (seg)	Throughput (Mbps)
1	0,707344	0,010349	1,14
0,8	0,669167	0,011080	1,06
0,6	0,624635	0,011974	0,98
0,4	0,542656	0,013368	0,88
0,2	0,374023	0,016087	0,73
Promedio	0,583565	0,012572	0,96

Tabla 3.8: Resultados caso 1, velocidad 1.25Gbps, 4 Generaciones

Carga (E)	Probabilidad de bloqueo	Retardo extremo a extremo (seg)	Throughput (Mbps)
1	0,653620	0,009592	1,23
0,8	0,615052	0,010071	1,17
0,6	0,551719	0,010845	1,09
0,4	0,430156	0,011740	1,00
0,2	0,213437	0,013654	0,86
Promedio	0,492797	0,011180	1,07

Velocidad de transmisión de 2.5 Gbps: Este caso incluye la variación del número de generaciones de 1 a 4, lo que implica que el algoritmo genético se iterara n generaciones con el objetivo de incrementar la calidad de la ruta que se obtenga de origen a destino. Se configura el parámetro de velocidad de transmisión a 2.5Gbps y se ejecuta este escenario 5 veces con las cargas de tráfico 0.2, 0.4, 0.6, 0.8, 1.0 respectivamente. Esto quiere decir que, en cada escenario, por enlace se tiene una velocidad de transmisión de 2.5Gbps, y para las generaciones se tienen en total 10 ejecuciones. En las tablas 3.9 y 3.10 se muestran los resultados.

Tabla 3.9: Resultados caso 1, velocidad 2.5Gbps, 1 Generación

Carga (E)	Probabilidad de bloqueo	Retardo extremo a extremo (seg)	Throughput (Mbps)
1	0,700195	0,010339	1,14
0,8	0,669010	0,011105	1,06
0,6	0,626302	0,011839	0,99
0,4	0,539360	0,013271	0,89
0,2	0,357734	0,015911	0,74
Promedio	0,578520	0,012493	0,96

Tabla 3.10: Resultados caso 1, velocidad 2.5Gbps, 4 Generaciones

Carga (E)	Probabilidad de bloqueo	Retardo extremo a extremo (seg)	Throughput (Mbps)
1	0,650664	0,009597	1,23
0,8	0,606380	0,010057	1,17
0,6	0,535104	0,010748	1,10
0,4	0,426094	0,011703	1,01
0,2	0,167656	0,012950	0,91
Promedio	0,477180	0,011011	1,07

3.2.2. Caso 2: 10 longitudes de onda

Se varía el parámetro longitudes de onda de 5 a 10 trayetos. Esto significa que por cada enlace se tendrán 10 longitudes de onda.

Velocidad de transmisión de 1.25 Gbps: Este caso incluye la variación del número

de generaciones de 1 a 4, lo que implica que el algoritmo genético se iterara con el objetivo de incrementar la calidad de la ruta que se obtenga de origen a destino. Se configura el parámetro de velocidad de transmisión a 1.25Gbps y se ejecuta este escenario 5 veces con las cargas de tráfico 0.2, 0.4, 0.6, 0.8, 1.0 respectivamente. Por lo cual se tiene una velocidad de transmisión de 1.25 Gbps por enlace en cada escenario con 10 ejecuciones por generación como se muestra en las tablas 3.11 y 3.12 se muestran los resultados.

Tabla 3.11: Resultados caso 2, velocidad 1.25Gbps, 1 Generación

Carga (E)	Probabilidad de bloqueo	Retardo extremo a extremo (seg)	Throughput (Mbps)
1	0,564531	0,012490	0,94
0,8	0,516602	0,013171	0,89
0,6	0,454062	0,014445	0,82
0,4	0,343984	0,016294	0,72
0,2	0,115469	0,019639	0,60
Promedio	0,398930	0,015208	0,80

Tabla 3.12: Resultados caso 2, velocidad 1.25Gbps, 4 Generaciones

Carga (E)	Probabilidad de bloqueo	Retardo extremo a extremo (seg)	Throughput (Mbps)
1	0,468359	0,011097	1,06
0,8	0,409063	0,011823	1,00
0,6	0,316250	0,012653	0,93
0,4	0,156224	0,013716	0,86
0,2	0,001068	0,014199	0,83
Promedio	0,270193	0,012698	0,94

Velocidad de transmisión de 2.5 Gbps: Este caso incluye la variación del número de generaciones de 1 a 4, lo que implica que el algoritmo genético se iterara con el objetivo de incrementar la calidad de la ruta que se obtenga de origen a destino. Se configura el parámetro de velocidad de transmisión a 2.5Gbps y se ejecuta este escenario 5 veces con las cargas de tráfico 0.2, 0.4, 0.6, 0.8, 1.0 respectivamente. Esto quiere decir que, en cada escenario, por enlace se tiene una velocidad de transmisión de 2.5Gbps, y para las generaciones se tienen en total 10 ejecuciones. En las tablas 3.13 y 3.14 se muestran los resultados.

Tabla 3.13: Resultados caso 2, velocidad 2.5Gbps, 1 Generación

Carga (E)	Probabilidad de bloqueo	Retardo extremo a extremo (seg)	Throughput (Mbps)
1	0,561432	0,012470	0,94
0,8	0,507930	0,013246	0,89
0,6	0,437318	0,014419	0,82
0,4	0,317474	0,016011	0,74
0,2	0,090835	0,019133	0,62
Promedio	0,382998	0,015056	0,80

Tabla 3.14: Resultados caso 2, velocidad 2.5Gbps, 4 Generaciones

Carga (E)	Probabilidad de bloqueo	Retardo extremo a extremo (seg)	Throughput (Mbps)
1	0,449844	0,011019	1,07
0,8	0,384375	0,011621	1,01
0,6	0,281250	0,012265	0,96
0,4	0,106693	0,012981	0,91
0,2	0,000195	0,012736	0,92
Promedio	0,244471	0,012124	0,97

Capítulo 4

ANÁLISIS DE RESULTADOS

En este capítulo se presenta el análisis de los resultados obtenidos. A partir de estos resultados se realiza un análisis descriptivo para cada método, se comparan los resultados y se establece una relación entre la probabilidad de bloqueo, el retardo extremo a extremo y el throughput.

4.1. Análisis de resultados escenario 1: SDON sin el método.

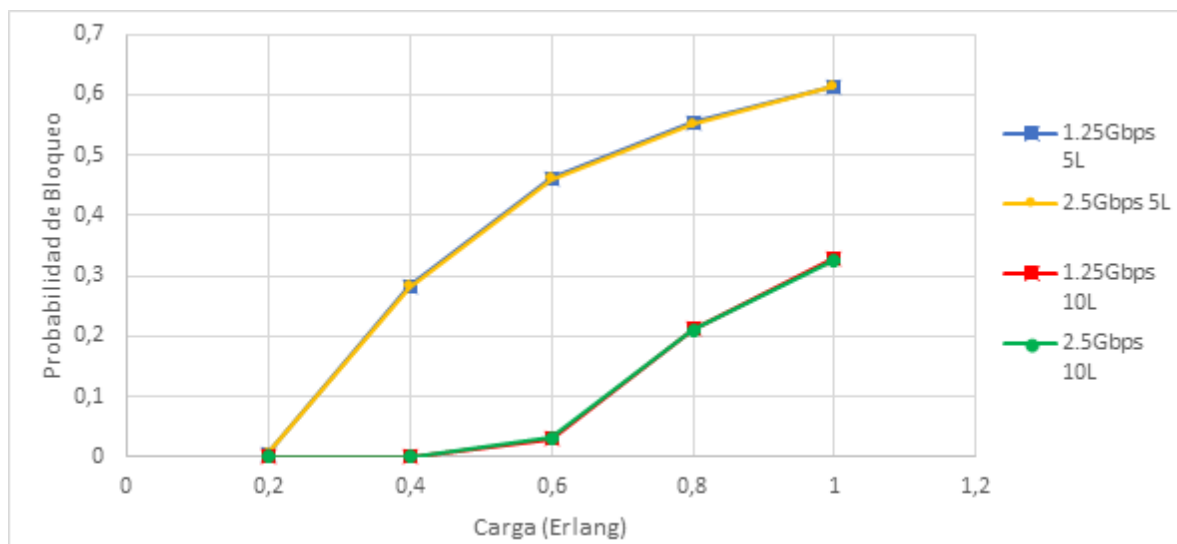


Figura 4.1: Probabilidad de bloqueo método SP.

En la figura 4.1 se observa que la probabilidad de bloqueo tiene un comportamiento similar a las velocidades de 1.25Gbps y 2.5Gbps en 5 longitudes de onda al igual que el caso de 10 longitudes de onda, debido a que no se varía el tamaño de los datos que circulan por la red, es decir, no se aplica el fenómeno flujo elefante - ratón descrito en [59], el cual podría cambiar el comportamiento del sistema al variar las velocidades. La aplicación de dicho fenómeno sale del alcance de este trabajo ya que implica escenarios de prueba enfocados específicamente en él.

La probabilidad de bloqueo de peticiones para cargas de tráfico menores a 0.6, en 10 longitudes de onda se mantiene por debajo de 10 %, en comparación a 5 longitudes de onda donde se mantiene menor a 50 %, lo cual indica que cuando el algoritmo SP

dispone de mayor cantidad de recursos y bajo tráfico, puede disminuir la pendiente de crecimiento de dicho parámetro.

Los casos que implementan 10L presentan una probabilidad de bloqueo promedio de 11 %, menor en comparación a 5L con 38 %, ya que el algoritmo al disponer de un mayor número de longitudes de onda puede atender un mayor número de peticiones y por ende mantener a la red fuera de saturación. Así mismo la probabilidad de bloqueo presenta una relación directamente proporcional a la carga, es decir, a mayor tráfico, mayor probabilidad de bloqueo, lo cual es un comportamiento esperado, pues el algoritmo SP, conforme se incrementa el tráfico, tiene menos opciones de rutas y por ende rechaza un mayor número de solicitudes. En la figura 4.2 se observa que el retardo extremo a

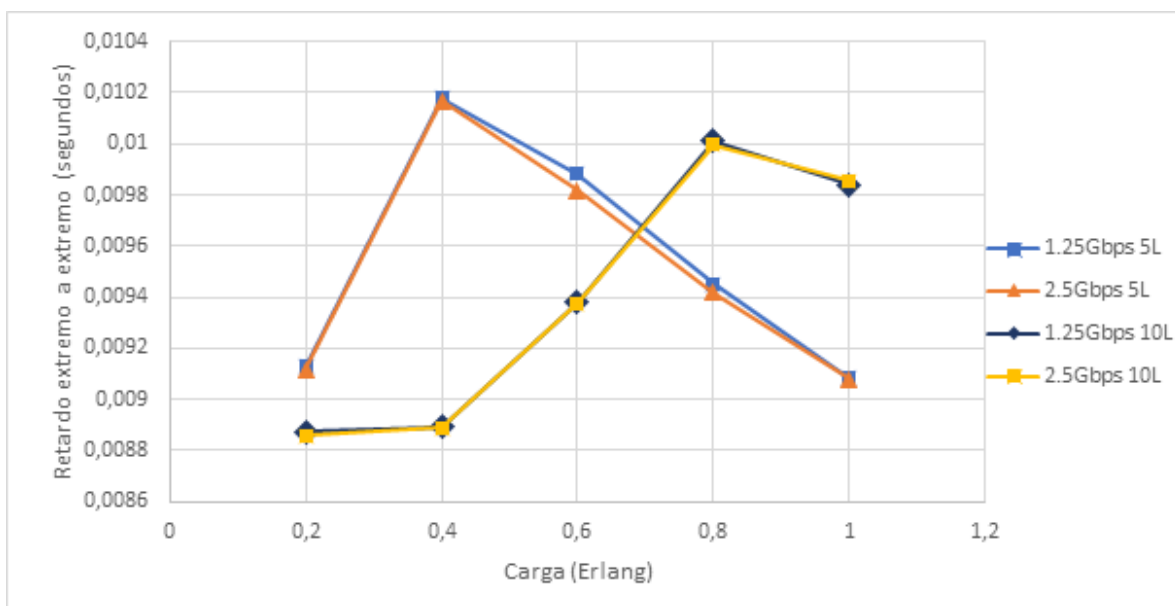


Figura 4.2: Retardo extremo a extremo método SP

extremo tiene un comportamiento similar a las velocidades de 1.25 y 2.5Gbps en 10 longitudes de onda, y como se explicó anteriormente, se debe a que no se varía el tamaño de los datos que circulan por la red, es decir, no se aplica el fenómeno de flujo elefante - ratón, que cambiaría el comportamiento del retardo extremo a extremo al tener trayectos de menor duración con flujo ratón frente a los elefantes que incrementan la duración.

El retardo extremo a extremo promedio para 10L a 1.25Gbps y 2.5Gbps es 9.4ms (ver tablas 3.5 y 3.6), menor en comparación para 5L a 1.25Gbps y 2.5Gbps con 9.54ms y 9.52ms (ver tablas 3.3 y 3.4) respectivamente, debido a que se cuenta con una mayor disponibilidad de longitudes de onda, haciendo que el algoritmo pueda establecer rutas más cortas con mayor probabilidad que 5L. Los cambios de pendiente para 5 y 10 longitudes de onda, en las cargas de tráfico 0.4 y 0.8 respectivamente se deben a la sobrecarga de los enlaces, porque se dispone de una menor cantidad de recursos, por ende se rechazan más solicitudes. En este sentido el retardo extremo a extremo de

todos los trayectos disminuye, indicando que se están estableciendo rutas con un menor número de saltos. La probabilidad de establecer rutas que impliquen un mayor uso de los recursos se hace menor en comparación a aquellas que presentan un menor uso.

En la figura 4.3 se observa que el throughput tiene un comportamiento similar a

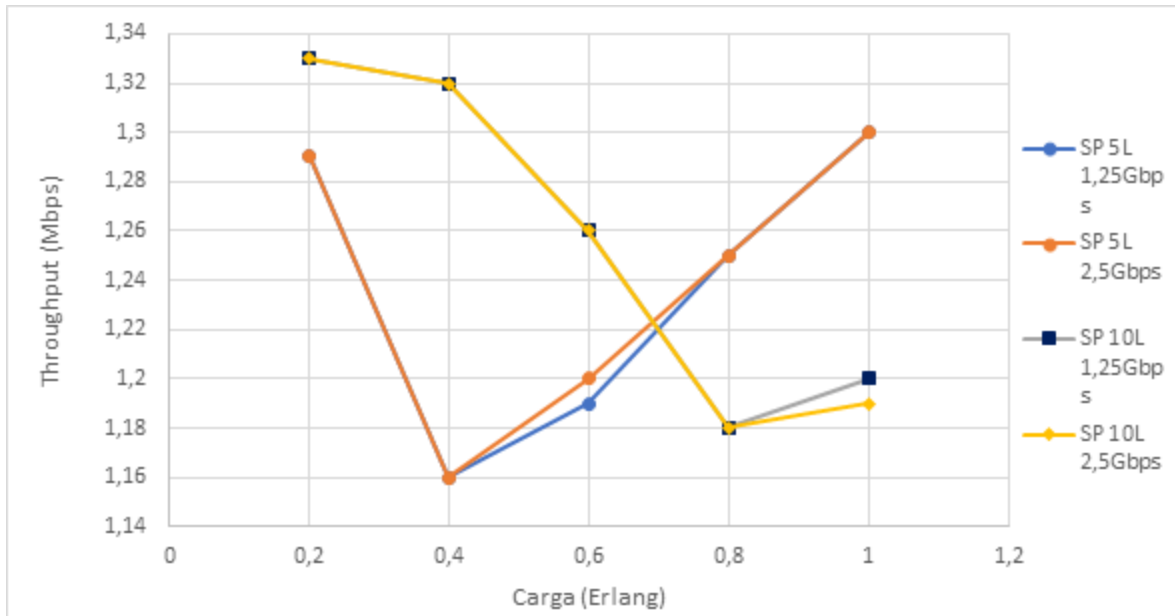


Figura 4.3: Throughput método SP

las velocidades de 1.25 y 2.5Gbps en 5L, al igual que el caso de 10L y, solo difieren en la carga 0.6 y 1.0 respectivamente con 1.2Mbps a 1.25Gbps y 1.19Mbps a 2.5Gbps.

Conforme aumenta la carga, en los casos con 10L el throughput disminuye hasta la carga 0.8 y después aumenta, lo cual es similar en los casos con 5L, donde disminuye en el rango 0.2 - 0.4 y aumenta desde la carga 0.4. Esto se debe a que en los casos de 5L desde la carga 0.4 se inicia el proceso de saturación de la red a causa de la limitación de los recursos, se rechazan un mayor número de solicitudes, y la probabilidad de establecer conexiones que impliquen un mayor uso de enlaces y longitudes de onda disminuye. Los casos de 10L tiene un mejor throughput promedio de 1.26 Mbps en comparación a los 1.24Mbps de 5L, debido a que el retardo extremo a extremo es menor para 10L y según la ecuación 3.3 entre menor sea el retardo, mayor es el throughput. El cambio de comportamiento de esta gráfica en la carga 0.4 a 5 longitudes de onda y en 0.8 para 10 longitudes de onda se debe a que el throughput es inversamente proporcional al retardo extremo a extremo, que también presenta inestabilidad producida por la limitación de recursos en la red.

En las figuras 4.1 y 4.2 se observa que la red para los casos con 5L y 10L entra en saturación desde la carga 0.4 y 0.8 respectivamente, debido a la disminución del retardo (a causa de la limitación de recursos) y se infiere que ésta inicia cuando se tiene una

probabilidad de bloqueo mayor o igual a 30 %. Se observa un aumento en el throughput de 1.16 a 1.3Mbps en el rango de cargas de 0.4 a 1.0 para los casos con 5L y de 1.18 a 1.2Mbps en el rango de cargas 0.8-1.0 para los casos con 10L, ya que el retardo extremo a extremo disminuye, debido a la inestabilidad del sistema limitado en recursos y por ende se necesita menos tiempo para transmitir una mayor cantidad de megabites, pero se rechaza un mayor número de solicitudes de conexión.

4.2. Análisis de resultados Escenario 2: SDON con GA

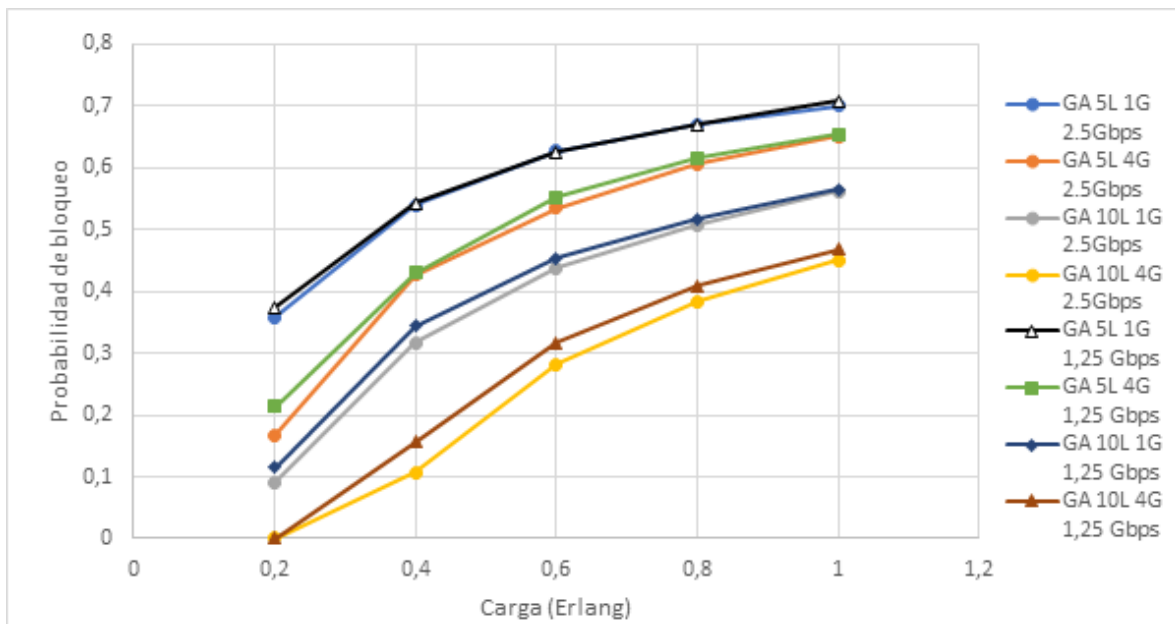


Figura 4.4: Probabilidad de bloqueo para todas las variaciones en el GA

En la figura 4.4 se observa que la probabilidad de bloqueo tiene un comportamiento similar a las velocidades de 1.25Gbps y 2.5Gbps, para 5L en 1 generación con un valor promedio de 58 %, debido a que es el caso con mayor limitación de longitudes de onda y una sola generación. A pesar de que la probabilidad de bloqueo a 1.25 y 2.5Gbps para 5L en 4 generaciones presenta un comportamiento similar, éstas corresponden a 49 % y 47 % respectivamente, lo cual sucede análogamente en 10L y 1 generación con probabilidad de bloqueo promedio de 40 % y 38 % respectivamente. Esta tendencia también se debe a que no se varía el tamaño del paquete, la cual afectaría principalmente la capacidad de acceso a la red y por ende a los parámetros de desempeño, al duplicar la velocidad de transmisión.

La red a 1.25Gbps y 2.5Gbps para 10L y 4 generaciones presenta un mejor desempeño con probabilidad de bloqueo promedio de 27 y 24 % respectivamente, en comparación

a los demás casos con promedios superiores a 38 %, lo cual es un comportamiento esperado para una red con mayor número de longitudes de onda y optimización de las rutas.

Según se observa en la figura 4.5, el retardo extremo a extremo promedio es mayor en los casos 10L para 1 y 4 generaciones con 15ms y 12ms respectivamente frente a los demás casos con promedios inferiores o iguales a 11ms, ya que, la red se encuentra en saturación para todos los casos, y se cuenta con una mayor disponibilidad de longitudes de onda en comparación a los casos de 5L, por ende tiene más probabilidad de establecer conexiones que impliquen atravesar un mayor número de enlaces y a mayor cantidad de enlaces en una ruta, mayor es el retardo que presenta. Esto permite inferir que, el GA necesita una mayor cantidad de recursos para que el nodo de control establezca un mayor número de conexiones, pero en consecuencia el retardo incrementa mientras la red se encuentra en saturación.

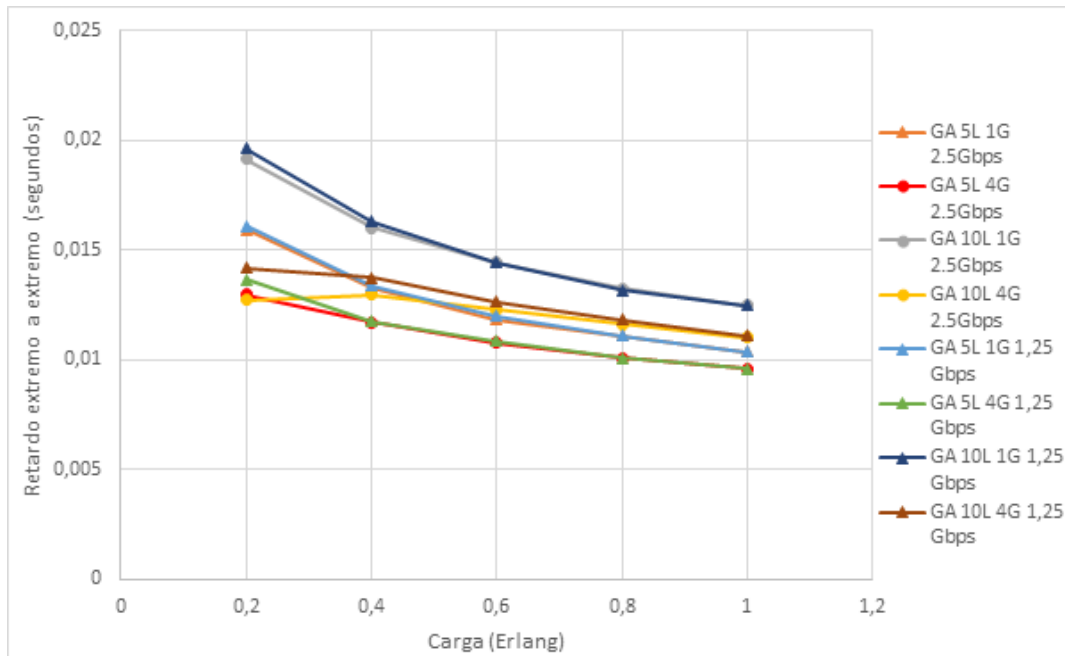


Figura 4.5: Retardo extremo a extremo del GA

En la figura 4.6 se observa que el throughput a 2.5Gbps en 10L y 4 generaciones muestra la transición de la red hacia saturación en la carga 0.4, debido a que para cargas inferiores mantiene su tendencia de disminución, pero, en cargas mayores a 0.4 esta tendencia, cambia a aumento. Para los demás casos se observa que la red ya se encuentra saturada desde la carga medida 0.2. Así mismo, tiene un comportamiento similar a las velocidades de 1.25 y 2.5Gbps en 5 longitudes de onda, al igual que los casos de 10 longitudes de onda y 1 generación, debido a que no se está variando el tamaño de los datos transmitidos y por ende tampoco la capacidad de acceso al medio.

En las figuras 4.4 y 4.5 se observa que la red en los casos con 10L para 4 generaciones

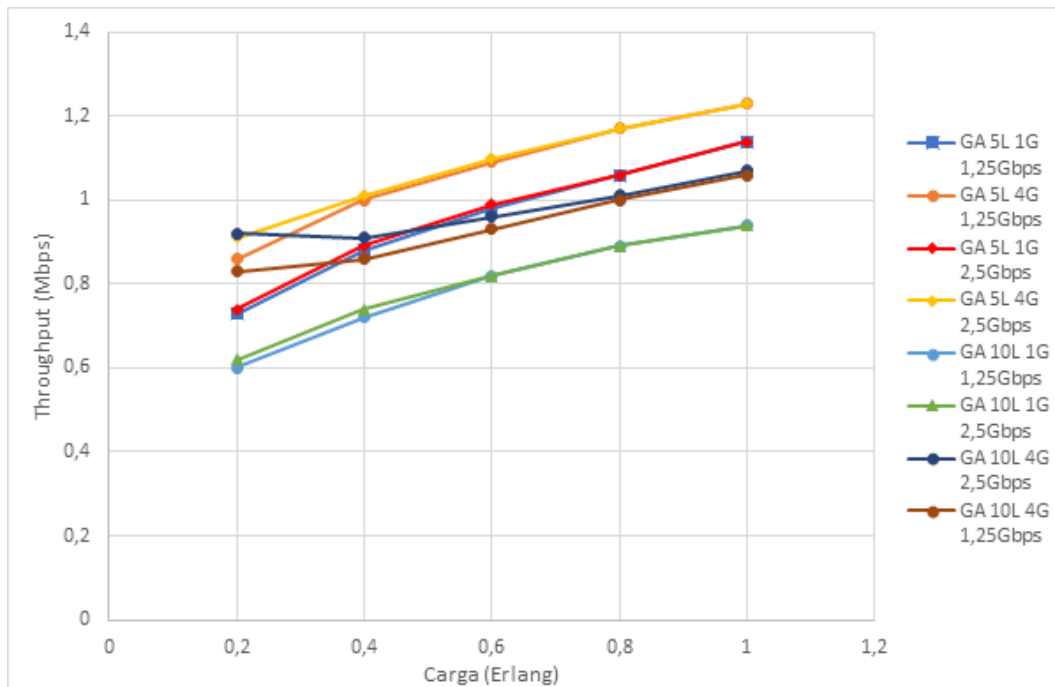


Figura 4.6: Throughput del GA

entra en saturación desde la carga 0.4 y 0.6 respectivamente, debido a la disminución del retardo (a causa de la limitación de recursos) y se infiere que ésta inicia cuando se tiene una probabilidad de bloqueo mayor o igual a 30 %, al igual que en el SP. Así mismo, el aumento del throughput para todos los casos del GA se debe a la inestabilidad del sistema, ya que el retardo extremo a extremo disminuye e incluso se rechazan un mayor número de solicitudes de conexión.

4.3. Comparación de desempeño SDON con y sin el método.

En esta sección se comparan los resultados obtenidos al simular los subcasos de la SDON con el método SP frente al GA. Adicionalmente, el GA se simula para 1 y 4 generaciones, lo que permite establecer un análisis en torno a la calidad de las rutas calculadas por el algoritmo genético.

4.3.1. 5 longitudes de onda, Velocidad 1.25Gbps

Se observa en la figura 4.7 que el algoritmo SP presenta un mejor desempeño con una probabilidad de bloqueo promedio de 38 %, en comparación al GA que para 1 generación presenta un promedio de 58 % y para 4 generaciones 49 %. Esto se debe a que el SP tiene como objetivo encontrar las rutas más cortas, y por ende utiliza menor cantidad de recursos para asignar una conexión. El GA implementa una población inicial

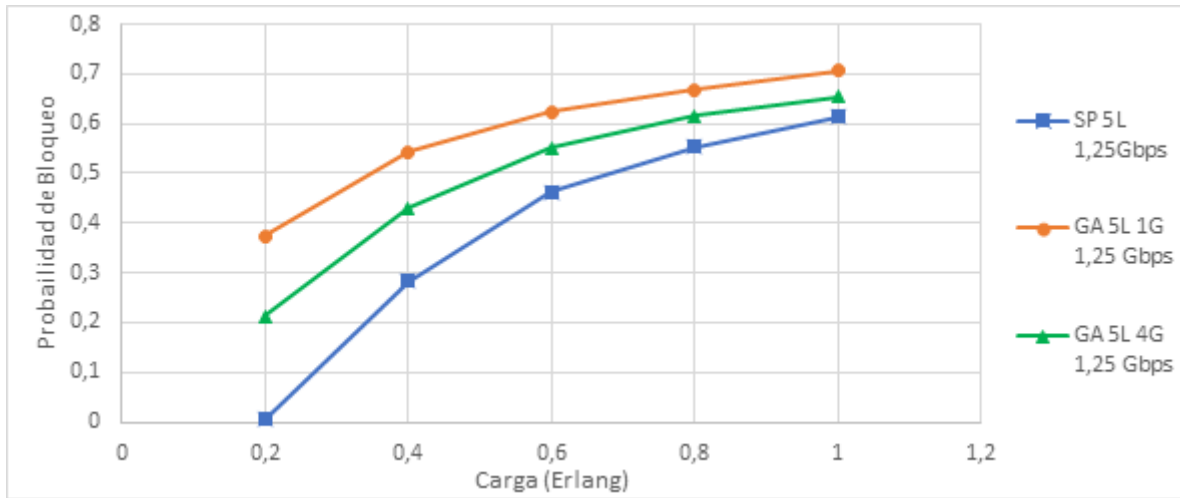


Figura 4.7: Comparación de probabilidad de bloqueo con y sin el método, 5L a 1.25Gbps

aleatoria y tiene una mayor probabilidad de que las rutas que arroje se encuentren en óptimos locales y no globales, por ende, la ruta obtenida representa una buena alternativa, pero no es la mejor.

La cantidad de iteraciones necesarias para establecer las rutas de la población inicial y el número de generaciones en conjunto con la aplicación de los operadores genéticos, son los factores que incrementan el costo computacional, lo cual se infiere a partir del tiempo que toman las simulaciones del GA en comparación a las del SP. Esta probabilidad de bloqueo de conexiones tiene una relación directamente proporcional con la carga, es decir, a mayor tráfico, mayor probabilidad de bloqueo, ya que los enlaces se encontrarán ocupados con más frecuencia, por ende, se rechaza un mayor número de solicitudes de conexión. También es importante recalcar que el método GA con 4 generaciones presenta un menor bloqueo que con 1 generación, ya que el algoritmo puede explorar un mayor espacio de soluciones, a las cuales se le aplican los operadores genéticos y entrega rutas con un menor número de enlaces. La desventaja de aplicar un mayor número de generaciones se ve reflejada en el elevado costo computacional y es un problema en conexiones que requieran una mayor rapidez en la entrega de la ruta.

El retardo extremo a extremo promedio del algoritmo SP presenta mejores resultados que el GA para 1 y 4 generaciones con 9ms, 13ms y 11ms respectivamente, donde este último tiene una tendencia a disminuir conforme aumenta la carga, lo cual se debe a la limitación de los recursos de red y que, para los valores de carga mostrados en la gráfica, la red ya se encuentra saturada. Así mismo, el GA en 4 generaciones presenta valores de retardo extremo a extremo menores que para 1 generación, pues como ya se mencionó, aumentar las generaciones puede mejorar la calidad de las rutas como se observa en la figura 4.8. El algoritmo SP presenta un mejor throughput con promedio de 1.24Mbps en comparación al GA para 1 y 4 generaciones con 0,96 y 1,07 Mbps respectivamente, lo cual se debe a que requiere menos tiempo (9ms) para transmitir sus

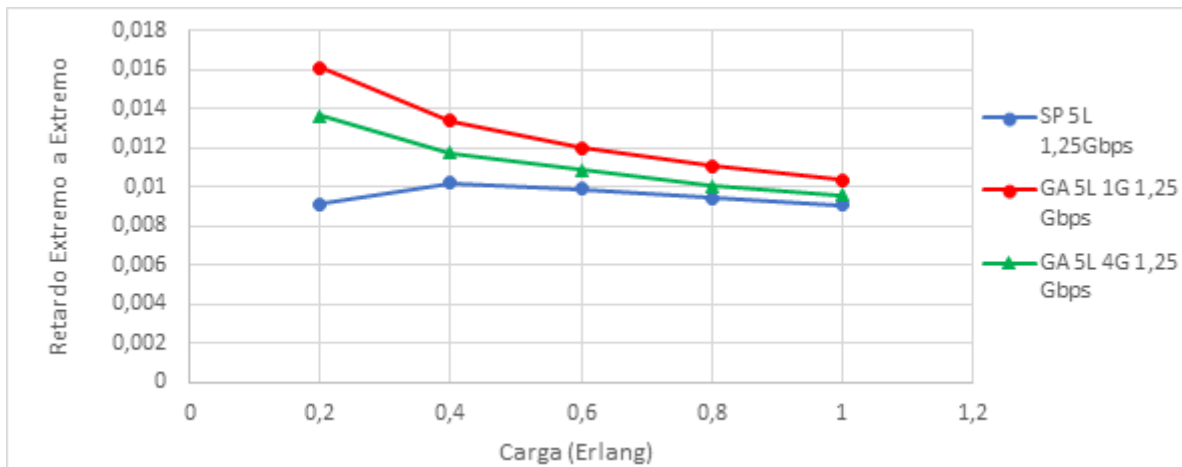


Figura 4.8: Comparación de retardo extremo a extremo con y sin el método, 5L a 1.25Gbps

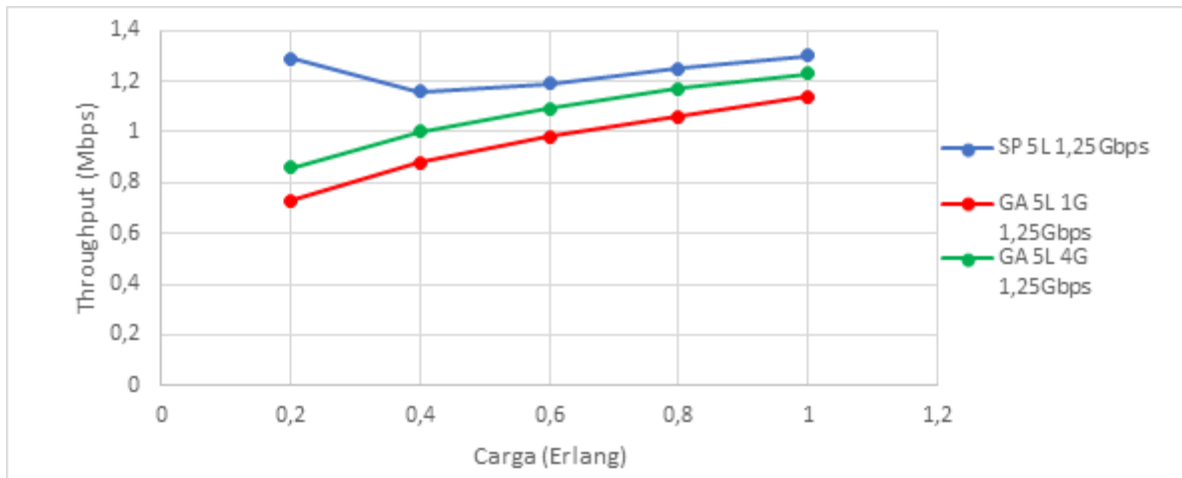


Figura 4.9: Comparación de retardo extremo a extremo con y sin el método, 5L a 1.25Gbps

datos. El throughput en el SP disminuye en el rango de cargas 0.2-0.4 debido a que la red aún no se encuentra en saturación.

La correlación del retardo y el throughput en el algoritmo SP permite observar en la figura 4.9 que a partir de la carga 0.4 la red entra en saturación, lo cual ya sucedía para cargas inferiores a 0.4 en el GA. Se puede inferir también que la probabilidad de bloqueo cuando se inicia el proceso de saturación es del 30 %.

4.3.2. 5 longitudes de onda, Velocidad 2.5Gbps

En las figuras 4.10, 4.11 y 4.12 se observa que el comportamiento de la probabilidad de bloqueo, el retardo extremo a extremo y el throughput conforme aumenta la carga

es similar a los casos con 1.25Gbps. Como ya se explicó anteriormente, la velocidad de transmisión produce un retardo de transmisión el cual está variando en el orden de los nanosegundos, en comparación al retardo total que se da en milisegundos.

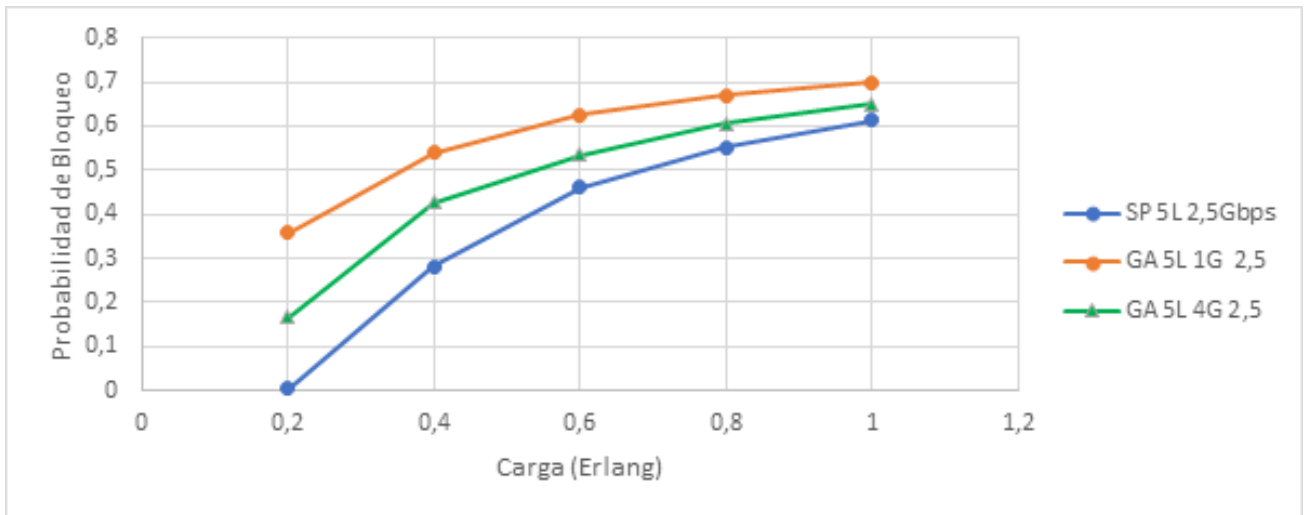


Figura 4.10: Comparación de probabilidad de bloqueo con y sin el método, 5L a 2.5Gbps

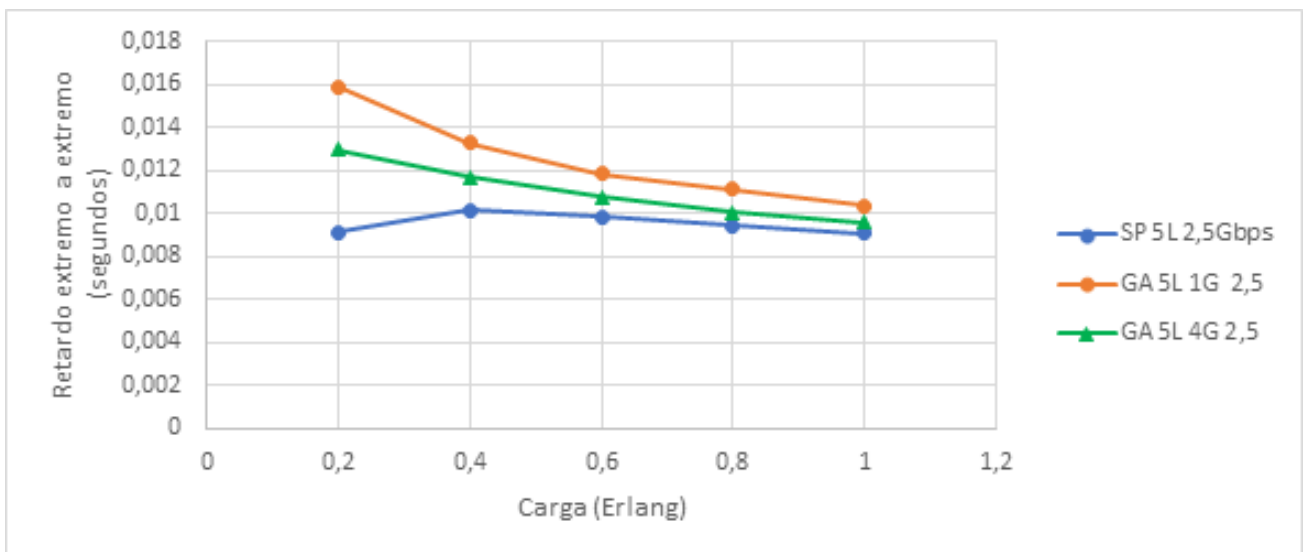


Figura 4.11: Comparación de retardo con y sin el método, 5L a 2.5Gbps

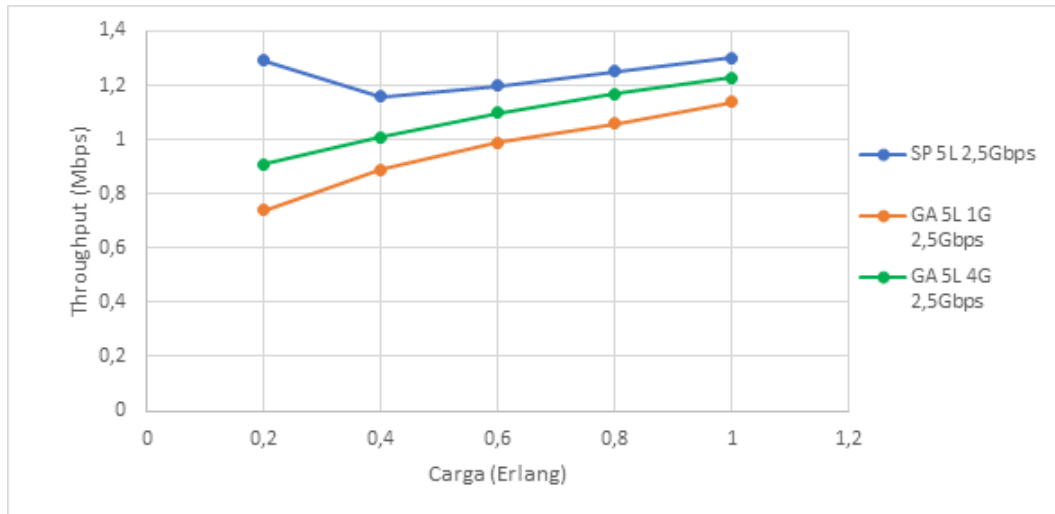


Figura 4.12: Comparación Throughput con y sin el método, 5L a 2.5Gbps

De nuevo, el SP presenta un mejor desempeño en general con probabilidad de bloqueo promedio de 38 %, retardo extremo a extremo de 9.52ms y throughput de 1.24Mbps, en comparación al GA con valores superiores a 47 %, 11ms, e inferiores a 1.07Mbps respectivamente.

Se infiere que la variación del número de generaciones de 1 a 4 para el GA redujo la probabilidad de bloqueo y el retardo en 10 % y 1ms respectivamente, e incrementó el throughput 0.11Mbps.

4.3.3. 10 longitudes de onda, Velocidad 1.25Gbps

En la figura 4.13 se observa que la probabilidad de bloqueo promedio para el SP presenta un mejor desempeño con 11 % en comparación al GA en 1 y 4 generaciones con 40 y 27 % respectivamente, pues el número de longitudes de onda es 10, por lo cual tanto el algoritmo SP y GA rechazan un menor número de solicitudes de conexión.

La figura 4.14 muestra que el SP presenta un menor retardo extremo a extremo promedio (9.4ms) con una tendencia de crecimiento conforme aumenta el tráfico, pues la red al encontrarse fuera de saturación aún tiene enlaces disponibles que permiten al algoritmo establecer rutas alternativas, aunque con mayores retardos. De forma similar el GA para 4 generaciones presenta un menor retardo extremo a extremo que en 1 generación, con promedios de 12 y 15 ms respectivamente.

En la figura 4.15 se observa un incremento el throughput del GA para 1 y 4 generaciones, lo cual se debe a que la red se encuentra en saturación y con el aumento del tráfico el retardo disminuye provocando el incremento del throughput. Así mismo el SP tiene un mejor promedio con 1,26Mbps frente al GA para 1 y 4 generaciones con 0.8Mbps y 0.94Mbps respectivamente. Al relacionar la tendencia del retardo extremo a

extremo, la probabilidad de bloqueo y el throughput es posible inferir que para valores de tráfico menores a 0.9 aproximadamente, la red funcionando con el SP no se encuentra en saturación y por ende responde a un mayor número de solicitudes de conexión. Lo mismo sucede para el GA en 4 generaciones, donde valores de carga menores a 0.6 arrojan una probabilidad de bloqueo menor al 30% en la SDON y hacen que se encuentre fuera de saturación.

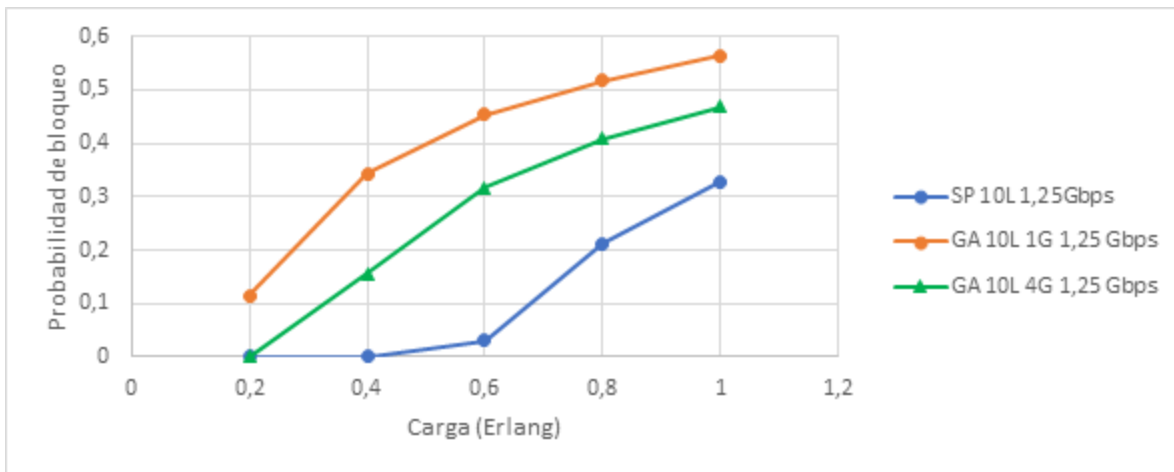


Figura 4.13: Comparación de probabilidad de bloqueo con y sin el método, 10L a 1.25Gbps

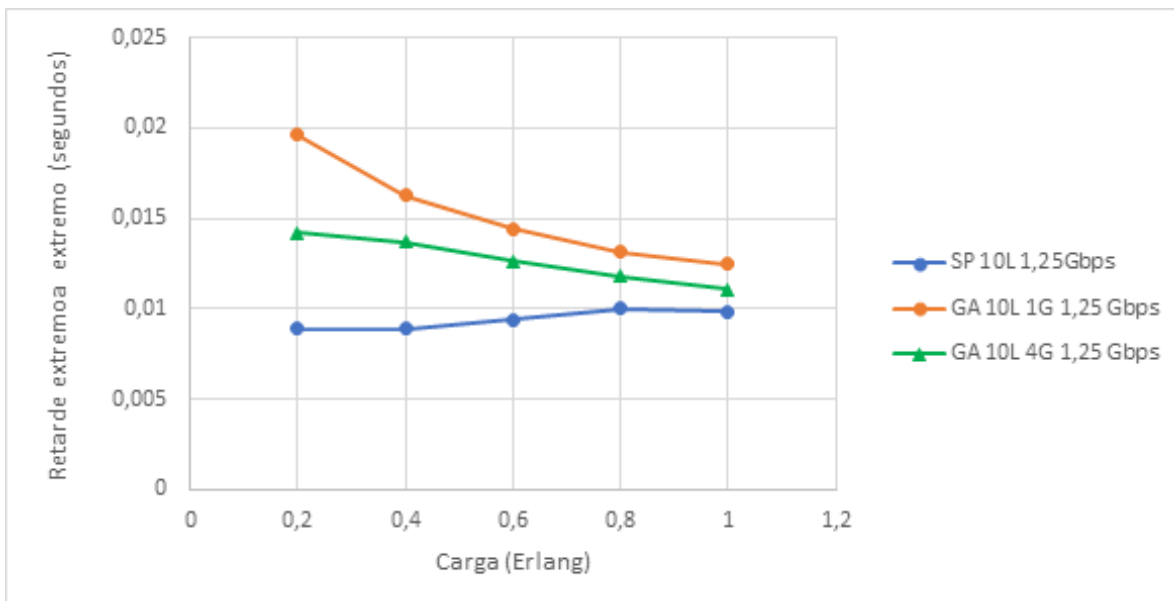


Figura 4.14: Comparación de retardo extremo a extremo con y sin el método, 10L a 1.25Gbps

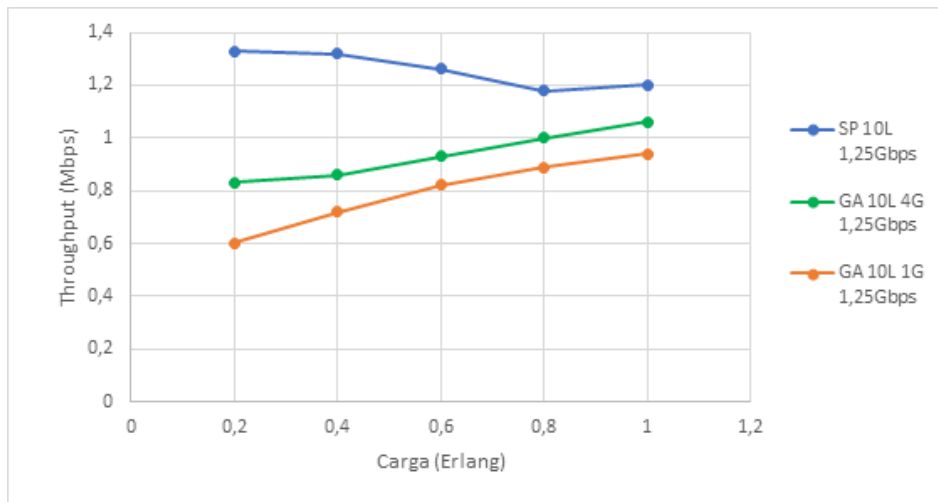


Figura 4.15: Comparación del Throughput con y sin el método, 10L a 1.25Gbps

4.3.4. 10 longitudes de onda, Velocidad 2.5Gbps

En las figuras 4.16 y 4.17 y 4.18 se observa que el comportamiento de la probabilidad de bloqueo, el retardo extremo a extremo y el throughput conforme aumenta la carga es similar a los casos con 1.25Gbps con 5 longitudes de onda. Como ya se explicó anteriormente, la velocidad de transmisión produce un retardo de transmisión el cual está variando en el orden de los nanosegundos, en comparación al retardo total que se obtiene en el orden de los milisegundos. La probabilidad de bloqueo observada en

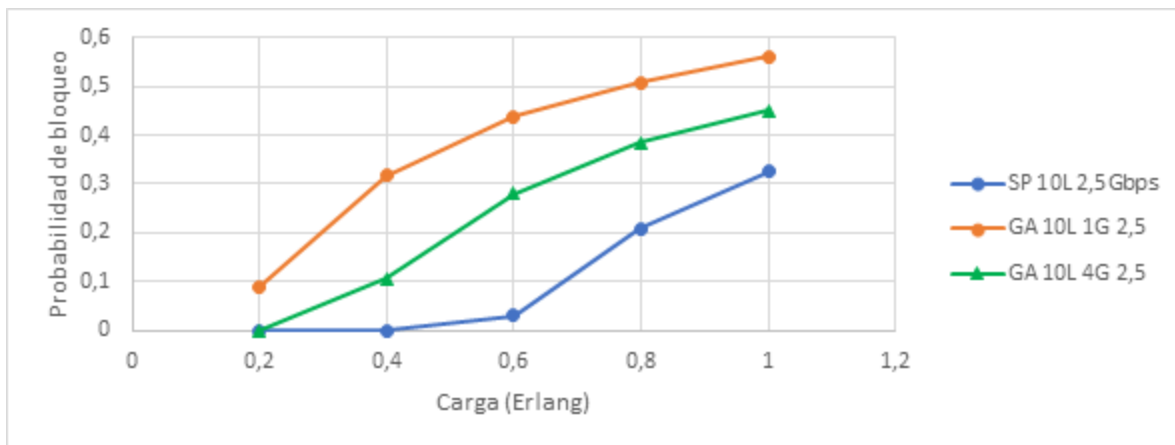


Figura 4.16: Comparación de probabilidad de bloqueo con y sin el método, 10L a 2.5Gbps

el algoritmo SP mantiene un valor bajo incluso hasta valores de carga mediana como 0.6, lo cual deduce que mantener niveles bajos de probabilidad de bloque depende en gran medida de la disponibilidad de recursos de la red, como las longitudes de onda. Analizando el comportamiento general del GA, al ajustar parámetros como el número

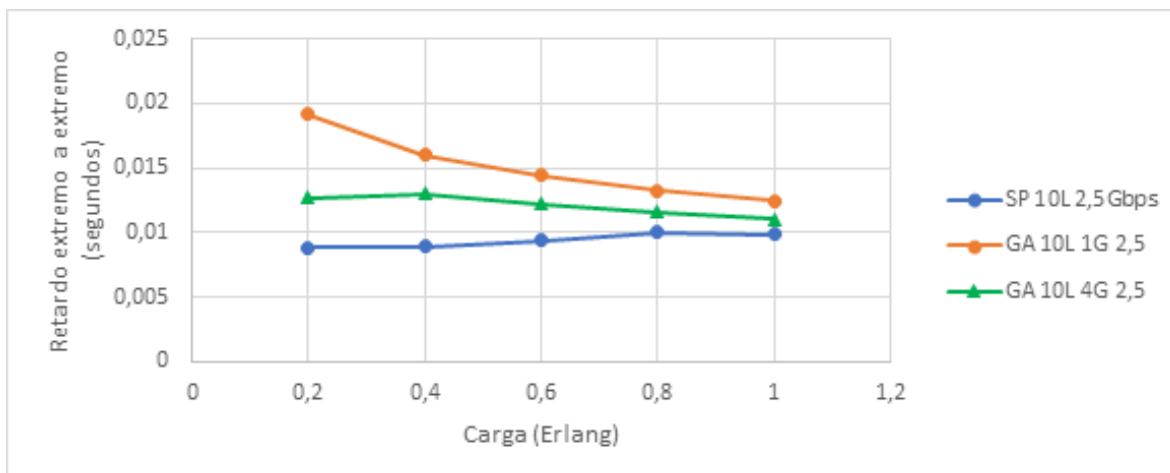


Figura 4.17: Comparación de retardo extremo a extremo con y sin el método, 10L a 2.5Gbps

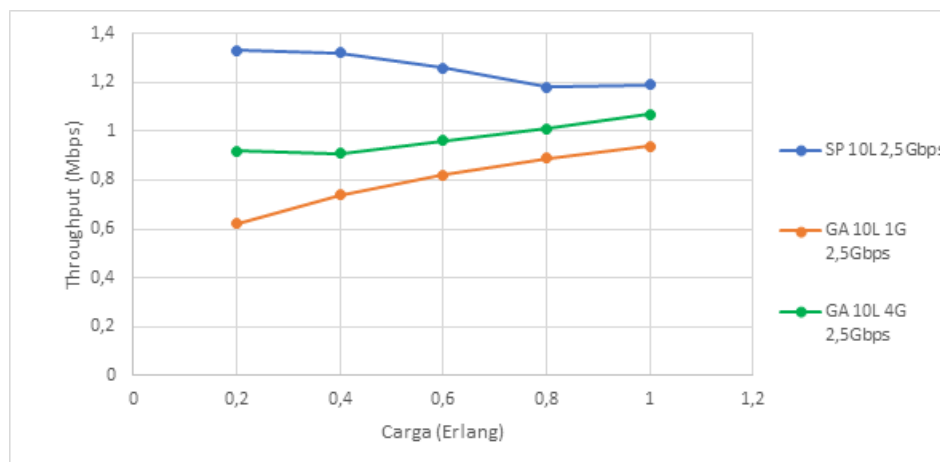


Figura 4.18: Comparación del Throughput con y sin el método, 10L a 2.5Gbps

de generaciones (1 a 4), se puede lograr una reducción de la probabilidad de bloqueo promedio de 38% a 24%, del retardo promedio de 15ms a 12ms y del throughput de 0.8Mbps a 0.97Mbps. Sin embargo, se debe tener especial cuidado porque incrementar las generaciones del GA aumenta el costo computacional, es decir se debe lograr un Pareto óptimo, el cual consiste en optimizar una característica sin afectar otra.

4.4. Comparación 5L vs 10L

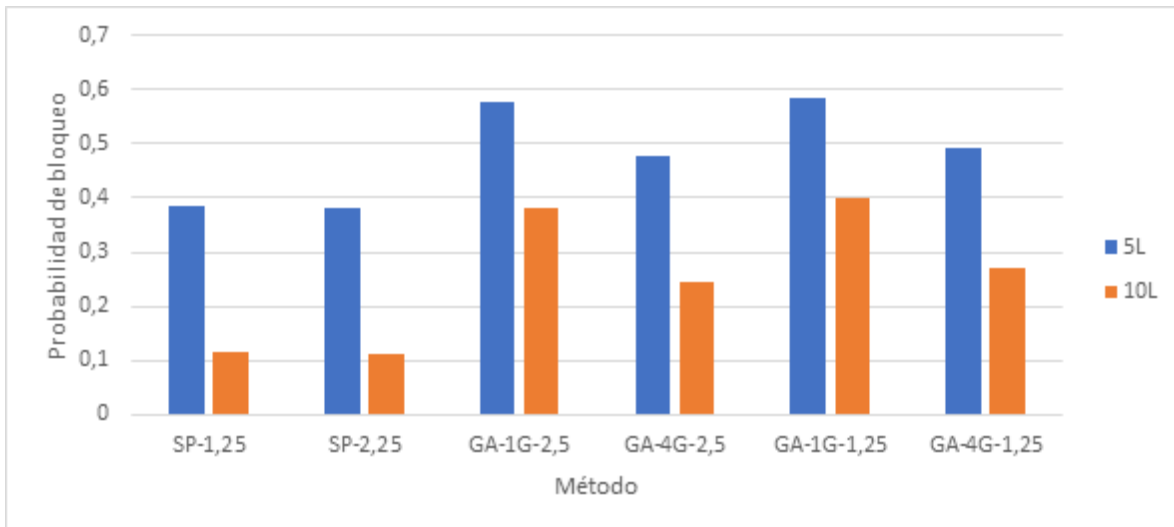


Figura 4.19: Comparación de la probabilidad de bloqueo 5L vs 10Ls

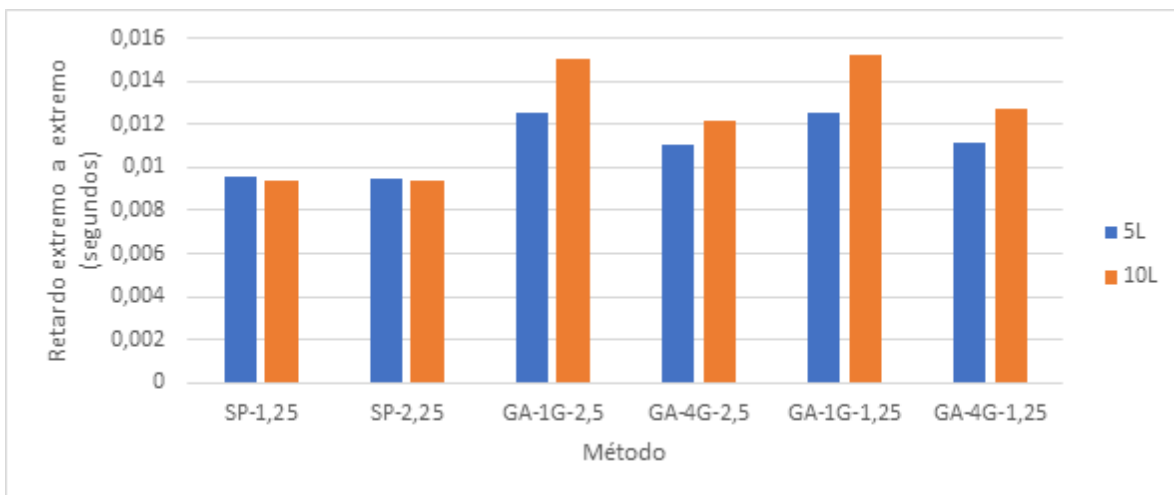


Figura 4.20: Comparación de retardo extremo a extremo 5L vs 10L

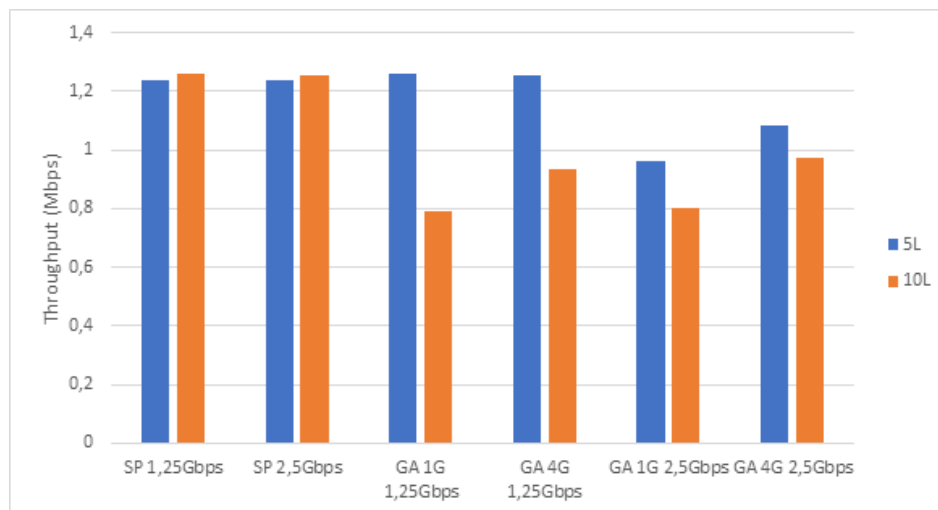


Figura 4.21: Comparación del throughput 5L vs 10L

Para evaluar el impacto de la limitación de recursos en la SDON con y sin el método, se optó por comparar los resultados para 5 y 10 longitudes de onda, lo cual se observa en las figuras 4.19, 4.20 y 4.21.

De forma general se deduce que la red tiene dos comportamientos, el primero es cuando se encuentra fuera de saturación y el segundo es cuando ya se encuentra en este estado. Para el algoritmo SP variar las longitudes de onda de 5 a 10 reduce 3.45 veces la probabilidad de bloqueo, pero para el GA esta disminución no es mayor a 1.55 veces en cualquiera de sus casos.

El retardo extremo a extremo para el SP disminuye al aumentar el número de longitudes de de 5 a 10, sin embargo, en el GA se incrementa el retardo debido a que la red se encuentra en saturación. En este sentido, la red con el GA al disponer de una mayor cantidad de longitudes de onda puede establecer trayectos que impliquen un mayor uso de los recursos, viéndose reflejado en el incremento del retardo extremo a extremo en comparación a los casos con 5 longitudes de onda, donde la limitación de los recursos reduce la probabilidad de establecimiento de rutas con un mayor número de saltos. Así mismo el throughput disminuye, debido a que es inversamente proporcional al retardo, lo que resulta en algo no beneficioso para el rendimiento en general, pues se busca reducir los tiempos de transmisión y aumentar la tasa de transferencia de datos efectiva.

4.5. Comparativa promedio parámetros de desempeño.

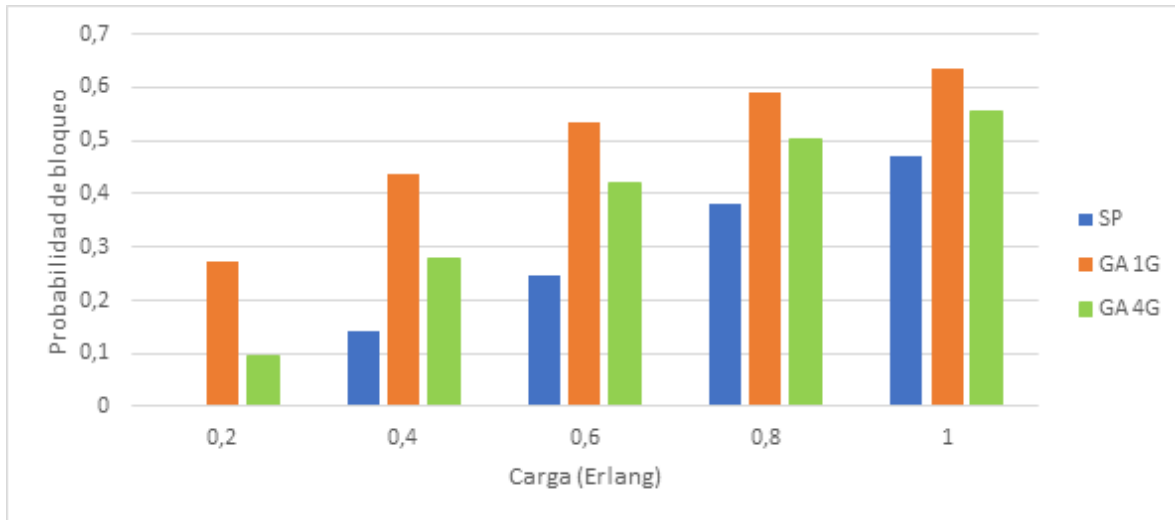


Figura 4.22: Comparativa promedio de la probabilidad de bloqueo

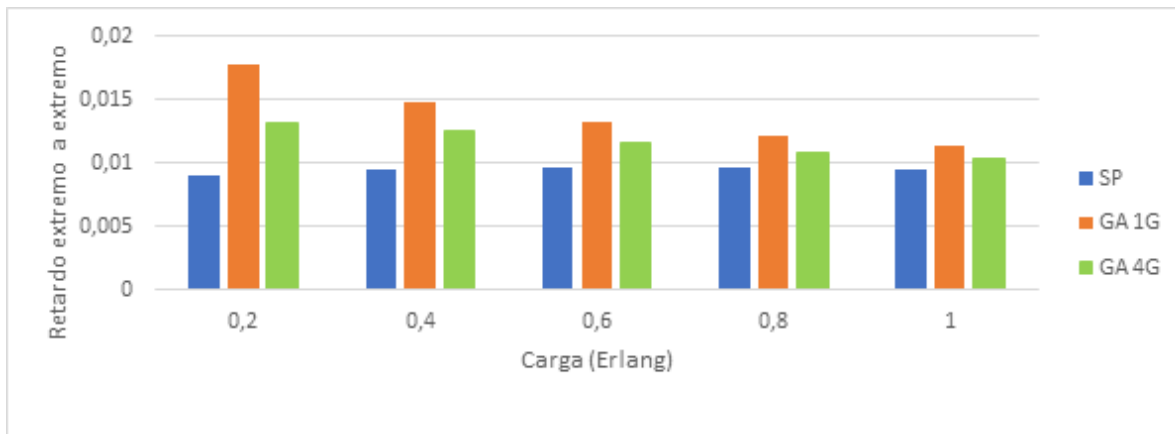


Figura 4.23: Comparación promedio del retardo extremo a extremo

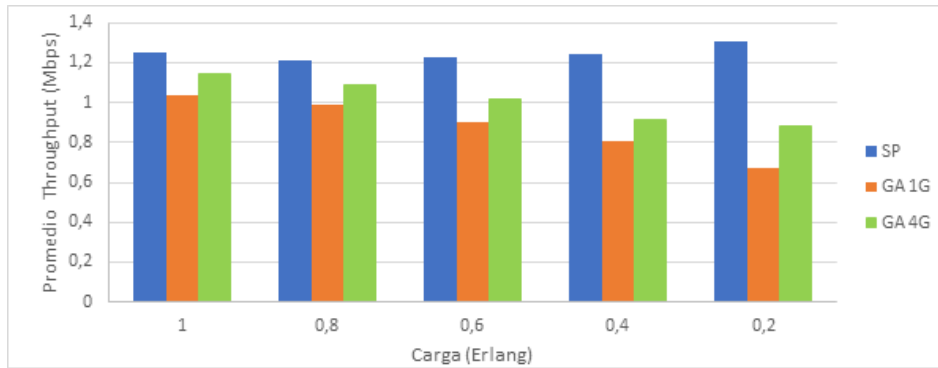


Figura 4.24: Comparación promedio del throughput

Como se puede ver en las figuras 4.22, 4.23 y 4.24, la red implementando el SP presenta un mejor desempeño general tanto en la probabilidad de bloqueo, retardo extremo a extremo y throughput a comparación del GA. En la gráfica 4.22, para la carga 0.2, la probabilidad de bloqueo tiende a cero, por tal motivo no se observa. Estos resultados permiten realizar las siguientes inferencias:

- El costo computacional del algoritmo genético es mucho más alto que el del Shortest Path, al momento de calcular las rutas.
- Incrementar el número de generaciones del algoritmo genético mejora la calidad de la ruta final, pero aumenta el costo computacional.
- La SDON no tiene un buen desempeño con el método GA para medianas 0.4 - 0.7 ni grandes cantidades de tráfico 0.7 - 1.0 en comparación al SP.
- La limitación de recursos y el aumento del tráfico provocan sobrecarga en la red, produciendo la inestabilidad que se observa en los cambios de pendiente en el retardo extremo a extremo y throughput. Esta inestabilidad puede ser el objeto de estudio central en otras investigaciones para determinar los factores a detalle que la producen.

Capítulo 5

CONCLUSIONES Y TRABAJOS FUTUROS

5.1. CONCLUSIONES

- La adaptación del método GA para la asignación dinámica de rutas en la SDON fue posible en bloques con funciones específicas, lo que permitió identificar que etapas requerían mayor robustez. Así mismo definir de manera detallada todos los parámetros que afectan el estado de la red dentro de la función de aptitud permitió incrementar la calidad de la ruta escogida en comparación a la población inicial.
- Se implementó el entorno de simulación mediante la herramienta OMNeT++ para la red óptica basada en SDN, en particular para la solución al problema de RWA, donde los comportamientos deseados de la SDON fueron abstraídos del marco conceptual, e implementados en la simulación.
- El algoritmo genético, que presenta un elevado costo computacional conforme aumenta la escalabilidad, tráfico de la red, y los resultados obtenidos en los parámetros de desempeño la llevan a saturación con mayor rapidez mientras que el SP mantiene a la red fuera de saturación, con un bajo costo computacional, probabilidad de bloqueo, retardo extremo a extremo y mayor throughput siendo una mejor solución al problema de enrutamiento.
- La limitación de los recursos en la red permitió identificar un umbral de tráfico, el cual, al ser sobrepasado, hizo que la red entrara en saturación, es decir, el tráfico fue tan elevado que se rechazaron un mayor número de solicitudes y el retardo extremo a extremo presentó inestabilidad, sin importar cuál de los dos métodos fuera implementado.
- El elevado costo computacional que supone implementar el algoritmo genético, evidenciado en la duración del tiempo real para cada subcaso, hacen que no sea óptimo en redes que requieran una respuesta rápida al cálculo de ruta, redes con un mayor número de nodos, puesto que al establecer la población inicial encontrar los nodos de destino será más complicado, y en redes con limitación de recursos, puesto que conforme se ocupen los recursos las posibilidades de encontrar una población inicial aleatoria se reducen.
- El aumento del número de longitudes de onda por enlace supone una estrecha relación con los parámetros de desempeño y por ende una gran mejora para am-

bos métodos, especialmente en la reducción de la probabilidad de bloqueo de peticiones.

- El incremento de las generaciones en el algoritmo genético demuestra una mejora en los parámetros de desempeño, pero hay que tener cuidado con no establecer un valor muy grande que pueda evidenciar otros problemas como el aumento del costo computacional.
- La herramienta OMNeT++ ofrece grandes posibilidades para la construcción detallada de los escenarios de simulación y personalización de su comportamiento, pero presenta una curva de aprendizaje alta.

5.1.1. TRABAJOS FUTUROS

El desarrollo de este trabajo de grado permite generar nuevas propuestas de investigación, las cuales se describen a continuación:

- Analizar el desempeño de una SDON con conmutación y asignación de espectro OBS/RSA al implementar algoritmos genéticos con elitismo para enrutamiento
- Desarrollo de un mecanismo de cálculo de rutas híbrido sobre una SDON que permita la alternancia de algoritmos de enrutamiento según las necesidades y prioridades de los paquetes.
- Comparar enrutamiento con algoritmos genéticos con y sin control cognitivo.
- Analizar la incidencia de los operadores genéticos y la función de aptitud de los algoritmos genéticos en el enrutamiento de redes de comunicaciones, con el fin de reducir el número de generaciones necesarias para un resultado óptimo y reducir el tiempo de computación.

Bibliografía

- [1] R. K. Jha and B. N. M. Llah, “Software defined optical networks (sdon): proposed architecture and comparative analysis,” *Journal of the European Optical Society-Rapid Publications*, vol. 15, no. 1, pp. 1–15, 2019.
- [2] G. Puerto-Leguizamón, B. Ortega, J. Capmany, K. Cardona-Urrego, and C. Suárez-Fajardo, “Evolución de las redes de datos: Hacia una plataforma de comunicaciones completamente óptica,” *Revista Facultad de Ingeniería Universidad de Antioquia*, no. 45, pp. 148–156, 2008.
- [3] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.
- [4] I. J. Roa Rodríguez, E. F. Arias Gerenas, *et al.*, “El problema de enrutamiento y asignación de espectro en redes ópticas flexibles: una solución desde la perspectiva de la optimización,”
- [5] B. C. Chatterjee, N. Sarma, P. P. Sahu, and E. Oki, *Routing and wavelength assignment for wdm-based optical networks: quality-of-service and fault resilience*, vol. 410. Springer, 2016.
- [6] A. M. Minguez, *Contribución al estudio y optimización de dispositivos basados en holografía dinámica para su uso en redes ópticas pasivas multiplexadas en longitud de onda Wdm-Pon*. PhD thesis, Universidad Politécnica de Madrid, 2007.
- [7] “Fibra ptica cwdm y dwdm blog davantel.” <https://blog.davantel.com/fibra-optica-cwdm-dwdm>, 04 2019.
- [8] D. A. B. Peñarredonda, I. D. T. Silva, E. A. Mateus, *et al.*, “Redes ópticas dwdm: diseño e implementación,” *Visión electrónica*, vol. 4, no. 1, pp. 70–80, 2010.
- [9] “Equipos dwdm pruebas monitorizacin y solucin de problemas de redes dwdm.” <https://www.viavisolutions.com/es-es/soluciones/proveedores-de-servicios/cableado/dwdm>, april 2021.
- [10] M. Nord, S. Bjørnstad, and C. Gauger, “Ops or obs in the core network,” in *Proc. of ONDM conference*, Citeseer, 2003.
- [11] Ó. G. de Dios, I. de Miguel, V. L. Álvarez, R. J. Durán, N. Merayo, and J. F. L. Poyo, “Estudio y simulación de tcp en redes de conmutación óptica de ráfagas (obs),” *XV Jornadas Telecom I+ D, Madrid*, 2005.
- [12] A. L. Lucia, *Diseño y análisis de una red de comunicaciones con conmutación óptica de paquetes*. PhD thesis, Universidad de Zaragoza, 2009.

- [13] W. Puche, G. Montoya, J. E. Sierra, Y. Donoso, *et al.*, “Tecnologías de transporte óptico: hacia optical burst switching (obs),” 2008.
- [14] M. Martín, “Evaluación de arquitecturas de red híbridas obs/ocs,” 05 2009.
- [15] Y. Liu, Y. Pan, M. Yang, W. Wang, C. Fang, and R. Jiang, “The multi-path routing problem in the software defined network,” in *2015 11th International Conference on Natural Computation (ICNC)*, pp. 250–254, IEEE, 2015.
- [16] M. Kantor, E. Biernacka, P. Boryło, J. Domżał, P. Jurkiewicz, M. Stypiński, and R. Wójcik, “A survey on multi-layer ip and optical software-defined networks,” *Computer Networks*, vol. 162, p. 106844, 2019.
- [17] G. Pereira and E. Gamess, “Lineamientos para el despliegue de redes sdn/open-flow,” *Selección*, vol. 4, no. 2, pp. 21–33, 2017.
- [18] M. Jarschel, T. Zinner, T. Hofffeld, P. Tran-Gia, and W. Kellerer, “Interfaces, attributes, and use cases: A compass for sdn,” *IEEE Communications Magazine*, vol. 52, no. 6, pp. 210–217, 2014.
- [19] Y. Zhao, Y. Wang, W. Wang, and X. Yu, “Software-defined optical networking (sdon): Principles and applications,” in *Optical Fiber and Wireless Communications*, InTech, 2017.
- [20] I. S. Pujol, *Contribución al diseño de redes ópticas de nueva generación: protección, calidad de servicio y eficiencia energética*. PhD thesis, Universidad Carlos III de Madrid, 2012.
- [21] A. S. Thyagaturu, A. Mercian, M. P. McGarry, M. Reisslein, and W. Kellerer, “Software defined optical networks (sdons): A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2738–2786, 2016.
- [22] D. T. Hai, “Multi-objective genetic algorithm for solving routing and spectrum assignment problem,” in *2017 Seventh international conference on information science and technology (ICIST)*, pp. 177–180, IEEE, 2017.
- [23] M.-Y. Wu and W. Shu, “Rpp: A distributed routing mechanism for strategic wireless ad hoc networks,” in *IEEE Global Telecommunications Conference, 2004. GLOBECOM'04.*, vol. 5, pp. 2885–2889, IEEE, 2004.
- [24] A. Erickson, I. A. Stewart, J. A. Pascual, and J. Navaridas, “Improved routing algorithms in the dual-port datacenter networks hcn and bcn,” *Future Generation Computer Systems*, vol. 75, pp. 58–71, 2017.
- [25] M. Forghani-elahabad and L. H. Bonani, “An improved algorithm for rwa problem on sparse multifiber wavelength routed optical networks,” *Optical switching and networking*, vol. 25, pp. 63–70, 2017.

- [26] S. Jain *et al.*, *Performance Evaluation of Multi-hop Routing in Software Defined Network*. PhD thesis, 2018.
- [27] Y. D. Rosita, E. E. Rosyida, and M. A. Rudiyanto, “Implementation of dijkstra algorithm and multi-criteria decision-making for optimal route distribution,” *Procedia Computer Science*, vol. 161, pp. 378–385, 2019.
- [28] C. T. P. Thanh, H. H. Nam, and T. C. Hung, “A heuristic algorithm for bandwidth delay constrained routing,” in *2014 International Conference on Advanced Technologies for Communications (ATC 2014)*, pp. 99–104, IEEE, 2014.
- [29] S. Jing, W. Muqing, B. Yong, and Z. Min, “An improved gac routing algorithm based on sdn,” in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, pp. 173–176, IEEE, 2017.
- [30] R. Kulkarni and K. Sharma, “Qos-based routing algorithm for software-defined network using ant colony optimization,” in *International Conference on Advances in Computational Intelligence and Informatics*, pp. 37–45, Springer, 2019.
- [31] R. Elshaer and H. Awad, “A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants,” *Computers & Industrial Engineering*, vol. 140, p. 106242, 2020.
- [32] C. Chen, C.-h. Yang, Z.-h. Wang, Q. Wu, Z. Wang, and X.-m. Wang, “Risk balancing routing assignment mechanism based software defined optical network service importance,” in *2017 16th International Conference on Optical Communications and Networks (ICOON)*, pp. 1–3, IEEE, 2017.
- [33] P. T. A. Quang, J.-M. Sanner, C. Morin, and Y. Hadjadj-Aoul, “Multi-objective multi-constrained qos routing in large-scale networks: A genetic algorithm approach,” in *2018 International conference on smart communications in network technologies (SaCoNeT)*, pp. 55–60, IEEE, 2018.
- [34] T. P. Valencia, A. M. H. Moncayo, and J. G. L. Perafan, “Cognitive control based on genetic algorithm for routing and wavelength assignment in optical obs/wdm networks,” in *Proceedings of the 2nd International Conference on Communications, Computation, Networks and Technologies INNOV*, pp. 11–17, 2013.
- [35] M. Fu and F. Wu, “Investigation of multipath routing algorithms in software defined networking,” in *2017 International Conference on Green Informatics (ICGI)*, pp. 269–273, IEEE, 2017.
- [36] C. L. Soria, D. R. Pandolfi, and S. M. Villagra, “Algoritmos genéticos celulares con operadores de recombinación aplicados a problemas de optimización discretos,” *Informes Científicos Técnicos-UNPA*, vol. 6, no. 3, pp. 1–21, 2014.

- [37] Y.-S. Yu and C.-H. Ke, “Genetic algorithm-based routing method for enhanced video delivery over software defined networks,” *International Journal of Communication Systems*, vol. 31, no. 1, p. e3391, 2018.
- [38] T. de Amorim Monteiro, E. Q. de Albuquerque, and A. M. Balieiro, “Traffic engineering in sdn with cultural algorithms,” in *2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pp. 1–5, IEEE, 2018.
- [39] C. Insfrán, D. Pinto, and B. Barán, “Diseño de topologías virtuales en redes ópticas. un enfoque basado en colonia de hormigas,” in *XXXII Latin-American Conference on Informatics*, pp. 173–195, 2006.
- [40] Y.-C. Chang, W.-X. Cai, and J.-W. Jhuang, “Bacteria-inspired communication mechanism based on software-defined network,” in *2018 27th Wireless and Optical Communication Conference (WOCC)*, pp. 1–3, IEEE, 2018.
- [41] A. Moujahid, I. Inza, and P. Larrañaga, “Tema 2. algoritmos genéticos,” *Departamento de Ciencia de la Computación e Inteligencia Artificial Universidad del País Vasco-Euskal Herriko Unibertsitatea*, pp. 1–33, 2008.
- [42] M. Pizzutti and A. E. Schaeffer-Filho, “Adaptive multipath routing based on hybrid data and control plane operation,” in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 730–738, IEEE, 2019.
- [43] A. B. Rodriguez, L. J. Ramirez, and F. R. Basile, “Solución metaheurística para el problema de enrutamiento (rwa) en redes de ópticas de multiplexación por división de longitud de onda (wdm),” *Información tecnológica*, vol. 28, no. 6, pp. 133–146, 2017.
- [44] A. Pérez, “Pros y contras de la metodología en cascada — obs business school,” 2016.
- [45] J. López, *Diseño de métodos Cross Layer cognitivos para redes de Comunicación Óptica de Ráfagas (OBS)*. PhD thesis, Tesis Doctoral), Universidad del Cauca, Popayán, Colombia, 2014., 2014.
- [46] M. Y. Arafat, F. Ahmed, and M. Sobhan, “On the migration of a large scale network from ipv4 to ipv6 environment,” *International journal of Computer Networks Communications*, vol. 6, pp. 111–126, 03 2014.
- [47] A. u. R. Khan, S. Bilal, and M. Othman, “A performance comparison of open source network simulators for wireless networks,” 11 2012.
- [48] A. Varga and R. Hornig, “An overview of the omnet++ simulation environment,” p. 60, 01 2008.
- [49] R. J. Durán Barroso, I. d. Miguel Jiménez, P. Fernández Reguero, N. Merayo Álvarez, J. C. Aguado Manzano, R. M. Lorenzo Toledo, E. J. Abril Domingo, *et al.*, “Simulador de redes wron,” 2015.

- [50] omnet.org, “<https://github.com/omnetpp/omnetpp>,” 12 de Noviembre de 2020.
- [51] M. F. C. Higuera and G. G. Acuña, “Estimación del valor teórico para el throughput en redes lan basadas en tecnología power line communications bajo el estándar homeplug 1.0,” *Publicaciones e Investigación*, vol. 7, pp. 89–97, 2013.
- [52] E. Chemeritskiy and R. Smelansky, “On qos management in sdn by multipath routing,” in *2014 International Science and Technology Conference (Modern Networking Technologies)(MoNeTeC)*, pp. 1–6, IEEE, 2014.
- [53] I. Tomkos, M. Angelou, R. J. D. Barroso, I. de Miguel, R. M. L. Toledo, D. Siracusa, E. Salvadori, A. Tymecki, Y. Ye, and I. T. Monroy, “Next generation flexible and cognitive heterogeneous optical networks,” in *The Future Internet Assembly*, pp. 225–236, Springer, Berlin, Heidelberg, 2012.
- [54] P. Lechowicz and K. Walkowiak, “Genetic algorithm for routing and spectrum allocation in elastic optical networks,” in *2016 Third European Network Intelligence Conference (ENIC)*, pp. 273–280, IEEE, 2016.
- [55] N. Muruganantham and H. El-Ocla, “Genetic algorithm-based routing performance enhancement in wireless sensor networks,” in *2018 IEEE 3rd international conference on communication and information systems (ICCIS)*, pp. 79–82, IEEE, 2018.
- [56] R. S. Tessinari, B. Puype, D. Colle, and A. S. Garcia, “Elastico++: An elastic optical network simulation framework for omnet++,” *Optical Switching and Networking*, vol. 22, pp. 95–104, 2016.
- [57] I. d. Gracia Moreno, “Integración en omnet++ de un estimador eficiente de medias como mecanismo de control del fin de la simulación,” B.S. thesis, 2009.
- [58] “G6961 aplicaciones de la multiplexacin por divisin en longitud de onda densa en el intradominio longitudinalmente compatibles.” <https://www.itu.int/rec/T-REC-G.696.1/es>, 10 2014. (undefined 8/11/2021 12:20).
- [59] J. L. R. hurtado, *Método de control congnitivo aplicado al enrutamiento de una red bajo arquitectura SDN/NFV*. PhD thesis, Universidad Del Cauca, 2018.

Anexos

Anexo 1)

```
std::vector<int> Genetico::findAdjacentNodes(int actNode){

    cModule *modulo_auxiliar= getParentModule()->getParentModule()-
>getParentModule()->getParentModule()->getSubmodule("nodo", actNode);
    assert( modulo_auxiliar != NULL);
    cTopology::Node *actualnod= netTopology.getNodeFor(modulo_auxiliar);
    assert( actualnod != NULL );

    int num_enlace_salida_nodo;
    int dirAdjacentNode;
    std::vector<int>adj;
    num_enlace_salida_nodo= actualnod->getNumOutLinks();
    for(int a=0; a<num_enlace_salida_nodo;a++){
        if(actualnod->getLinkOut(a)->isEnabled()){
            dirAdjacentNode= actualnod->getLinkOut(a)->getRemoteNode()-
                >getModuleId()-2;
            adj.push_back(dirAdjacentNode);
        }
    }
    return adj;
}
```

Figura 5.1: Algoritmo para encontrar nodos adyacentes

Anexo 2)

```

std::map<std::vector<int>,int>Genetico::rutasAleatorias(int origen, int destino,
cTopology::Node *origennod , cTopology::Node *destinonod){
    std::map<std::vector<int>,int> rutas;
    int numRutas=6;
    int limite=0;
    while(contRutas<numRutas){
        if(limite<1000){
            limite++;
            bool i=false;
            int adjacent;
            std::vector<int>v;
            int actNode=origen;
            std::vector<int>vAdj;
            std::map<int,int> ant;
            v.push_back(origen);
            do{
                vAdj=findAdjacentNodes(actNode);
                if(vAdj.size(>0){
                    int nodo;
                    int bandera=0;
                    for (std::size_t i = 0; i < vAdj.size(); i++){
                        nodo=vAdj[i];
                        if(ant.find(nodo)!=ant.end()){
                            bandera++;
                        } }
                    if(bandera==vAdj.size()){
                        i=true;
                        ant.clear();
                        vAdj.clear();
                    }
                }
                else{
                    do{ adjacent= vAdj[intuniform(0,vAdj.size()-1)];
                    }while(ant.find(adjacent)!=ant.end());
                    v.push_back(adjacent);
                    if(adjacent==destino){if(rutas.find(v)==rutas.end()){
                        rutas[v]=contRutas;
                        contRutas++;
                        ant.clear();
                        i=true; }
                    else{ant[actNode]=1;
                        actNode=adjacent;
                        vAdj.clear();}}}
                    else{ i=true; }} while(i==false);}
            else{contRutas=numRutas;} } }
    return rutas;}

```

Figura 5.2: Algoritmo para cálculo de rutas aleatorias

Anexo 3)

```

std::map<std::vector<int>, double>Genetico::seleccionRutas(std::map<std::vector<int>,
double>rutasAptitud){
    int padres=2;
    std::vector<int>vectorDeRutas;
    int_matrix_t vectorRutasOrdenadas;
    double funcionAptitud;
    double sumaAptitudes;
    double rangos;
    std::vector<double>vectorFAptitud;
    std::vector<int>rutaPadre;
    std::map<std::vector<int>, double>resultadoSeleccion;
    for(std::map<std::vector<int>, double>:: iterator it= rutasAptitud.begin();
it!=rutasAptitud.end();
++it){
        vectorDeRutas=it->first;
        vectorRutasOrdenadas.push_back(vectorDeRutas);
        funcionAptitud=it->second;
        sumaAptitudes=sumaAptitudes+funcionAptitud;
        vectorFAptitud.push_back(funcionAptitud);    }
    size_t pos=0;
    for(int i=0; i<padres;i++){
        double numeroRandom= uniform(0,1);
        std::vector<double>:: iterator itedos;
        pos=0;
        double sumaRangos =0;
        for( itedos=vectorFAptitud.begin(); itedos!= vectorFAptitud.end(); itedos++){
            rangos = *itedos/sumaAptitudes;
            sumaRangos = sumaRangos + rangos;
            if(sumaRangos>=numeroRandom)break;
            pos++;        }
        rutaPadre= vectorRutasOrdenadas.at(pos);
        if(resultadoSeleccion.size()>0){
            if(resultadoSeleccion.find(rutaPadre)==resultadoSeleccion.end()){
                resultadoSeleccion.insert(std::pair<vector<int>,double>(rutaPadre,*itedos));
            }
            else{
                i--;        }        }
        else{
            resultadoSeleccion.insert(std::pair<vector<int>,double>(rutaPadre,*itedos));    }
    }
    return resultadoSeleccion;
}

```

Figura 5.3: Algoritmo para selección de rutas

Anexo 4)

```

std::map<std::vector<int>,int>Genetico::producirHijo1(std::vector<int>vectorResultante1,
std::vector<int>vectorResultante2){
    size_t posCruce1;
    size_t posCruce2;
    std::vector<int>vectorResultantet;
    int posNI1;
    int posNI2;
    std::map<std::vector<int>,int>hijoUno;
    bool flag=false;
    for(int p=1; p<(vectorResultante1.size()-1);p++){
        for(int q=1; q<(vectorResultante2.size()-1);q++){
            if(vectorResultante1.at(p)==vectorResultante2.at(q)){
                posCruce1=p;
                posCruce2=q;
                p=vectorResultante1.size()-1;
                break;
            }
            else{
                flag=true;
            }
        }
    }
    if(flag==true){
        return hijoUno;
    }
    else{
        for(int p1=0;p1<posCruce1; p1++){
            vectorResultantet.push_back(vectorResultante1.at(p1));
        }
        for(int p1=posCruce2;p1<vectorResultante2.size(); p1++){
            vectorResultantet.push_back(vectorResultante2.at(p1));
        }
        for(int yu=0;yu<vectorResultantet.size();yu++){
            for(int wu=0; wu<vectorResultantet.size();wu++){
                if(vectorResultantet.at(yu)==vectorResultantet.at(wu) && yu!=wu){
                    posNI1= yu;
                    posNI2=wu;
                    yu=vectorResultantet.size();
                    break;
                }
            }
        }
        std::vector<int>vectorResultantet_recortado;
        for(int yu=0;yu<posNI1;yu++){
            vectorResultantet_recortado.push_back(vectorResultantet.at(yu));
        }
        for(int wu=posNI2;wu<vectorResultantet.size();wu++){
            vectorResultantet_recortado.push_back(vectorResultantet.at(wu));
        }
        if(vectorResultantet_recortado.empty()){
            hijoUno.insert(std::pair<vector<int>, int>(vectorResultantet,0));
        }
        else{
            hijoUno.insert(std::pair<vector<int>, int>(vectorResultantet_recortado,0));
        }
        return hijoUno;
    }
}

```

Figura 5.4: Algoritmo para producir rutas previas al cruce