



Propuesta de un algoritmo para la asignación dinámica de espectro en redes ópticas elásticas (EON)

Proposal of an algorithm for the dynamic spectrum allocation on elastic optical networks (EON)

Sebastián David Ossa Hernández*
José Alejandro Vargas Gutiérrez**
Gustavo Gómez***

(Recibido el 26-11-2019. Aprobado el 17-01-2020)

Estilo de citación de artículo:

S. D. Ossa Hernández, J. A. Vargas Gutiérrez, y G. Gómez, "Propuesta de un algoritmo para la asignación dinámica de espectro en redes ópticas elásticas (EON)", *Lámpsakos*, (23), pp. 15-24. (enero-junio, 2020). DOI: 10.21501/21454086.3145

Resumen

Este artículo trata sobre el desarrollo e implementación de una nueva propuesta para realizar la asignación dinámica de espectro en una red OPS FlexGrid, mediante un algoritmo voraz basado en el movimiento pendular. Este algoritmo fue implementado en una red con rejillas de ocho y dieciséis slots y también fue comparado con el algoritmo LAUC – VF adaptado a redes RSA, arrojando resultados satisfactorios relacionados con los parámetros de medición, tales como, la probabilidad de bloqueo de la red y el retardo de extremo a extremo. El desarrollo de esta propuesta proporciona una nueva alternativa para la gestión de las redes ópticas elásticas (EON) con asignación de espectro fija en el tiempo.

Palabras clave: Algoritmo; cognitividad; EON; FlexGrid; metaheurísticas; NSFNeT; OPS; redes elásticas; RSA; WDM.

Abstract

This article deals with the development and implementation of a new proposal to perform dynamic spectrum allocation in an OPS Flex Grid network, using a voracious algorithm based on pendulum motion. This algorithm was implemented in a network with grids of eight and sixteen slots and was also compared with the LAUC-VF algorithm adapted to RSA networks, yielding satisfactory results related to the measurement parameters, such as the probability of blocking the network and the end-to-end delay. The development of this proposal provides a new alternative for the management of Elastic Optical Networks (EON) with fixed spectrum allocation over time.

Keywords: Algorithm; cognitive; elastic networks; EON; FlexGrid; metaheuristics; NSFNeT; OPS; RSA; WDM.

* Ingeniero en Electrónica y Telecomunicaciones, Universidad del Cauca. Integrante del grupo de investigación Nuevas Tecnologías en Telecomunicaciones – GNTT, Popayán-Colombia. Contacto: sebastianossa@unicauca.edu.co

** Ingeniero en Electrónica y Telecomunicaciones, Universidad del Cauca. Integrante del grupo de investigación: Nuevas Tecnologías en Telecomunicaciones – GNTT, Popayán-Colombia. Contacto: vgjose@unicauca.edu.co

*** MsC. Electrónicas y telecomunicaciones, Universidad del Cauca. Integrante del grupo de investigación Nuevas Tecnologías en Telecomunicaciones – GNTT, Popayán-Colombia. Contacto: gtgomez@unicauca.edu.co

1. INTRODUCCIÓN

La asignación de espectro es un proceso clave en cualquier OTN (Optical Transport Network) para garantizar la calidad del servicio o QoS (Quality of Service), pues de esta depende que los servicios ofrecidos en la red dispongan de los recursos necesarios para viajar a través de ella [1]. Tanto en redes convencionales como en las nuevas redes flexibles bajo RSA (Routing and Spectrum Allocation) es necesario disponer de algoritmos capaces de asignar las longitudes de onda sobre las que viajará la información, siendo este un campo de investigación bastante importante y prometedor [2]. En este documento presentamos el diseño y aplicación de un algoritmo para la asignación de espectro de forma dinámica en una red RSA con ocho o dieciséis slots de ancho de banda y con tasas de datos de bits variables.

Se ha partido con la implementación del modelo de red NSFNet con tecnología OPS (Optical Packet Switching) en la herramienta de simulación de redes OMNeT++ [3]. Las metaheurísticas para la solución de problemas de optimización han sido aplicadas para el diseño del algoritmo propuesto, el cual ha sido llamado “algoritmo de péndulo” y su funcionamiento ha sido puesto a prueba sobre la red mencionada y comparado con el desempeño del algoritmo de asignación de espectro LAUC – VF (Latest Available Unscheduled Channel – Void Filling) adaptado a redes RSA, de manera que se evalúa el comportamiento a nivel de probabilidad de bloqueo y retardo de extremo a extremo, como se podrá observar a lo largo del artículo.

Diseño de la red y el algoritmo

Diseño de la red

La red NSFNet (National Science Foundation Network) es una red WAN (Wide Area Network) utilizada ampliamente para el estudio e investigación en el campo de las telecomunicaciones, esta se distribuye a lo largo de catorce ciudades de Estados Unidos mediante una topología de red irregular [4].

El IDE OMNeT++ fue utilizado para el modelado y caracterización de la red, permitiendo la implementación de los algoritmos y su posterior análisis. El desarrollo de los nodos de la red fue realizado mediante la implementación de un sistema de módulos funcionales que contienen las instrucciones necesarias para la operación de la red y la descripción virtual de los respectivos elementos que la conforman. En la Fig. 1 puede observarse la red NSFNet implementada en OMNeT++.

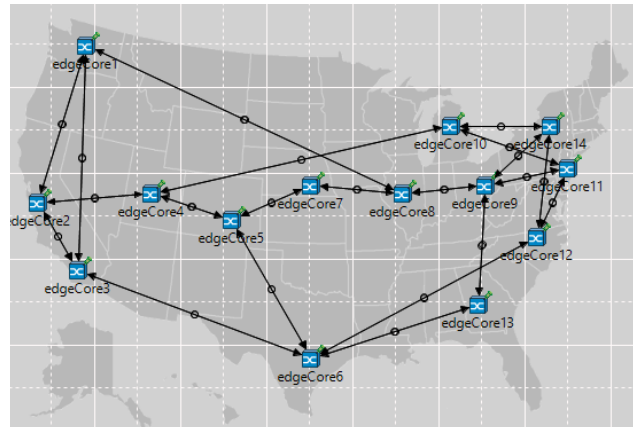


Fig. 1. Red NSFNet implementada en OMNeT++.

Redes elásticas

Las redes de fibra óptica convencionales, caracterizadas por usar típicamente tecnología CWDM (Coarse Wavelength Division Multiplexing) o DWDM (Dense Wavelength Division Multiplexing) son “rígidas” en cuanto a su funcionamiento, pues su operación es estática en el tiempo, manteniendo siempre su configuración inicial, incapaces de adaptarse a los nuevos requerimientos de la red [5], [6], por lo que las redes elásticas o flexibles (FlexGrid) se perfilan como una alternativa interesante y prometedora para solucionar problemas como el uso eficiente del espectro del que dispone la red [7], [8]. Por ejemplo, en una red WDM (Wavelength Division Multiplexing), los canales disponen de un ancho de banda fijo invariante en el tiempo, el cual es aprovechado en su totalidad por los servicios

DOI: <https://doi.org/10.21501/21454086.3145>

entrantes, aun cuando estos ocupen solo una porción de todo el ancho de banda disponible, lo que supone un desperdicio de recursos que bien podrían emplearse para enviar más información en el mismo instante de tiempo. Las redes ópticas elásticas buscan evitar el desperdicio de recursos adaptando dinámicamente las características de la red al estado de uso de esta en cada instante, por lo que si, por ejemplo, un servicio solicita recursos para ser transmitido, le sea asignada la cantidad de ancho de banda real que demanda y no todo el espectro del que dispone el canal, lo que supone un ahorro real del espectro óptico.

RSA busca solucionar el problema de enrutamiento de paquetes a lo largo de la red y, al mismo tiempo, pretende realizar la asignación de espectro para el paquete en cuestión. Esto puede hacerse de dos formas, la primera y más compleja es RSA de una etapa, en la que los dos problemas se solucionan de manera simultánea, lo que implica mayor gasto computacional; la segunda es RSA de dos etapas, en la que el problema es reducido a dos problemas: R+SA, de esta forma, el enrutamiento y la asignación de espectro se calculan de forma independiente el uno del otro, lo que si bien aumenta el tiempo de cálculo, disminuye la complejidad computacional. Este segundo método ha sido el elegido para el desarrollo de este trabajo de investigación.

RSA opera sobre una red que dispone de una rejilla o grilla espectral que es una concatenación de slots o porciones de espectro sobre los cuales viaja la información [9]. Típicamente, las redes cuentan con rejillas de cuatro, ocho o dieciséis slots de ancho de banda. Cada uno de estos slots tiene dos características principales que lo identifican: la frecuencia central y su ancho de banda. El ancho de banda de cada slot es de 12,5 GHz y está dividido en dos slices de 6,25 GHz y ubicado en las frecuencias que indica [10]. En la Fig. 2 se observa la grilla espectral.

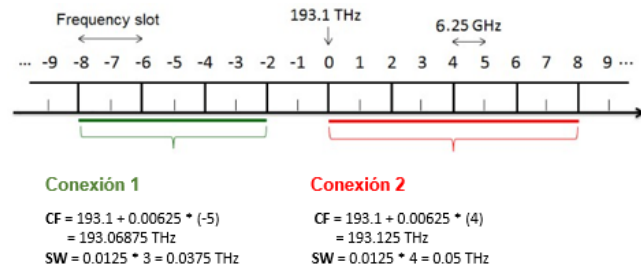


Fig. 2. Ejemplo de grilla espectral RSA.

En este trabajo de investigación el enrutamiento de los paquetes se realiza mediante la aplicación del algoritmo del camino más corto [11], también conocido como algoritmo Dijkstra, en el que mediante un grafo de red se calcula la ruta más corta entre el origen A y el destino B, proporcionando así el camino para el envío de la información.

Una vez se dispone de la ruta que seguirá el paquete, se hace uso de un algoritmo de asignación de espectro para la solución del problema SA, en este caso y como ya se mencionó, los algoritmos LAUC – VF adaptado a RSA y el algoritmo del péndulo se encargarán de esta labor.

Algoritmo LAUC-VF

El algoritmo LAUC implementado inicialmente para redes WDM ha sido adaptado para realizar la asignación de espectro en redes FlexGrid. Este algoritmo se basa en la implementación del vector Horizon, el cual contiene los tiempos en los que cada slot de la red estará disponible. Así pues, cuando un servicio solicita recursos para ser transmitido, el algoritmo consulta el vector Horizon y asigna aquellos slots que cumplan el requerimiento espectral y se encuentren más próximos a ser desocupados, evitando de esta manera la fragmentación del espectro.

El algoritmo LAUC es ampliamente utilizado gracias a su relativa baja complejidad y buena eficiencia, por lo que es un buen punto de comparación para evaluar

el comportamiento del algoritmo péndulo. A continuación, en Fig 3, se muestra el pseudocódigo del algoritmo LAUC.

```

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

Fig. 3. Ejemplo de algoritmo LAUC

Algoritmo péndulo

El algoritmo péndulo busca realizar una asignación de espectro al hacer un barrido en forma pendular sobre la grilla espectral, de modo que parte de un extremo de la grilla, para posteriormente pasar al otro extremo. Después de cada iteración, el algoritmo reduce su movimiento, y logra examinar los slots más centrales de la grilla, de ahí que recibe su nombre.

Diferentes metaheurísticas fueron revisadas para lograr el desarrollo del algoritmo, para finalmente dar con aquella que mejor se adecuaba al problema en cuestión, por lo que el algoritmo puede clasificarse dentro de la metaheurística de algoritmos voraces [12], [13], pues recibe un conjunto finito de elementos y retorna un conjunto de aquellos que pueden considerarse como una solución al problema. De esta manera son retornados aquellos slots que cumplen las condiciones de ancho de banda solicitadas por el servicio a transmitir.

Una característica de este algoritmo es que cumple con tres de las tareas del ciclo cognitivo propuesto en [14] y [15]. Este ciclo puede observarse en la Fig. 4, y se compone de seis tareas o fases: observación, orienta-

ción, decisión, planeación, acción y aprendizaje. Según [4] y [16], un algoritmo puede denominarse cognitivo si cumple con al menos las etapas de observación y acción del ciclo cognitivo. Dado que el algoritmo del péndulo realiza un recorrido de la rejilla espectral en busca de slots disponibles, puede decirse que cumple con la tarea de observación. Posteriormente, el algoritmo analiza si los slots encontrados cumplen con los requisitos de ancho de banda demandados por el servicio o información a transmitir, lo que corresponde a la fase de análisis del ciclo. Finalmente, una vez se han determinado cuáles de los slots resultan adecuados para el envío de la información, se realiza la asignación de estos al servicio a servir, cumpliendo así con la fase de acción del ciclo. Por lo que el algoritmo del péndulo encaja en la categoría de algoritmos cognitivos.

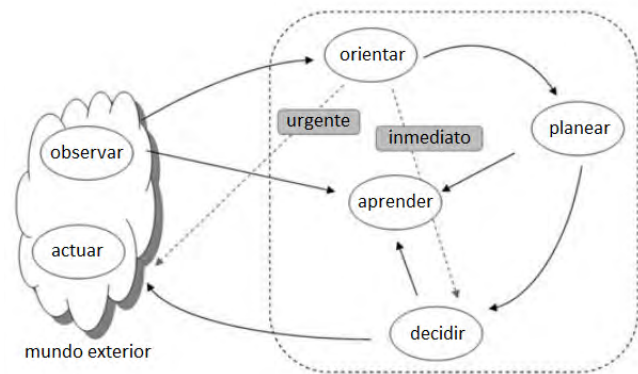


Fig. 4. Ciclo cognitivo [14].

El diseño inspirado en la naturaleza del movimiento pendular permite al algoritmo recorrer la rejilla espectral de forma tal que realiza una inspección minuciosa del espectro disponible en la red, buscando slots a lo largo de toda la rejilla y evaluando cada conjunto posible de tamaño S , donde S es la cantidad de ancho de banda ocupada por el servicio o información a transmitir.

El algoritmo evalúa en cada iteración los posibles conjuntos de tamaño S contenidos dentro de la rejilla, exceptuando cada vez aquellos conjuntos ya revisados y desplazando la revisión hacia al centro de la rejilla un slot a la vez, lo que garantiza la exploración de toda la grilla, la cual finaliza una vez se encuentra los recursos necesarios para la transmisión de la información.

DOI: <https://doi.org/10.21501/21454086.3145>

La Fig. 5 muestra un ejemplo de cómo opera el algoritmo péndulo para atender un servicio que ocupa dos slots de ancho de banda, sobre una rejilla de ocho slots en total.

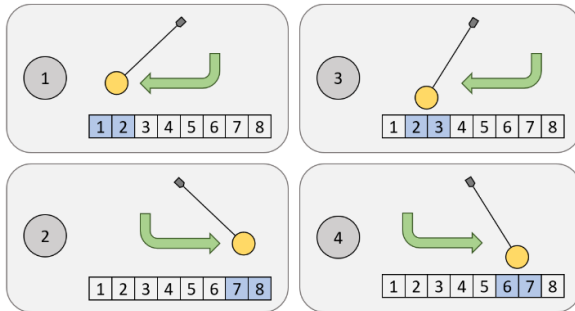


Fig. 5. Ejemplo de funcionamiento del algoritmo péndulo.

A continuación, en Fig. 6, se muestra el pseudocódigo correspondiente al algoritmo del péndulo.

```

PÉNDULO (packetSlots, slotsDisponibles[numSlots])
wl = 0
n = numSlots
s = packetSlots
si no hay slots disponibles hacer
    descartar paquete
de otra forma, hacer
    si s = 1 hacer
        para i = 0; i < n/2; i++ hacer
            si slotsDisponibles[i] = 0 hacer
                slotsDisponibles[i] = 1
                wl = i
                retornar wl
            de otra forma, si slotsDisponibles[n-1-i] = 0 hacer
                slotsDisponibles[n-1-i] = 1
                wl = n-1-i
                retornar wl
        si s = 2 hacer
            para i = 0; i < n/2; i++ hacer
                si slotsDisponibles[i] = 0 y slotsDisponibles[i+1] = 0 hacer
                    slotsDisponibles[i] = 1
                    slotsDisponibles[i+1] = 1
                    wl = i+1
                    retornar wl
                de otra forma, si slotsDisponibles[n-1-i] = 0 y slotsDisponibles[n-2-i] hacer
                    slotsDisponibles[n-1-i] = 1
                    slotsDisponibles[n-2-i] = 1
                    wl = n-2-i
                    retornar wl
        si s = 3 hacer
            para i = 0; i < n/2; i++ hacer
                si slotsDisponibles[i] = 0 y slotsDisponibles[i+1] = 0 y slotsDisponibles[i+2] = 0 hacer
                    slotsDisponibles[i] = 1
                    slotsDisponibles[i+1] = 1
                    slotsDisponibles[i+2] = 1
                    wl = i+2
                    retornar wl
                de otra forma, si slotsDisponibles[n-1-i] = 0 y slotsDisponibles[n-2-i] y slotsDisponibles[n-3-i] hacer
                    slotsDisponibles[n-1-i] = 1
                    slotsDisponibles[n-2-i] = 1
                    slotsDisponibles[n-3-i] = 1
                    wl = n-3-i
                    retornar wl
    
```

Fig. 6. Ejemplo algoritmo péndulo

La complejidad computacional del algoritmo es mayor que la del algoritmo LAUC, pero es gracias a esto que es capaz de realizar una mejor asignación del espectro.

Escenario de simulación

Se han implementado diferentes configuraciones en la red NSFNet con el fin de evaluar el comportamiento del algoritmo péndulo al compararlo con el algoritmo LAUC bajo diferentes condiciones de red. La red ha sido diseñada de forma tal que haga uso de la tecnología OPS para el envío de información.

Las fuentes generadoras de datos dispuestas en cada uno de los nodos de la red son capaces de generar paquetes de diferente tamaño y de forma aleatoria, siguiendo una distribución exponencial, los cuales pueden ser vistos como abstracciones de diferentes servicios, tales como datos, audio y video, pasando por paquetes que ocupan uno, dos y tres slots de ancho de banda, de forma tal que la red deberá proporcionar recursos para atender paquetes de tamaño variable y generados aleatoriamente en el tiempo.

La frecuencia de generación de los paquetes puede ser modificada, lo que permite modelar diferentes densidades de tráfico, y poder así observar el comportamiento del algoritmo para cargas de tráfico muy bajas, bajas, medias, altas y muy altas.

La red ha sido dispuesta de forma tal que la velocidad de datos puede ser de 1,25 o 2,5 Gbps y el tamaño de la rejilla espectral fue de ocho y dieciséis slots, lo que permite evaluar el desempeño del algoritmo en redes con media o alta disponibilidad de espectro. Los casos de simulación pueden verse condensados en la Tabla I.

Tabla I. Casos de simulación

Escenarios de simulación							
Topología de red NSFNet – Red Distribuida							
Casos de simulación							
Caso 1				Caso 2			
Red NSFNet RSA distribuida, implementando el algoritmo LAUC				Red NSFNet RSA distribuida, implementando el algoritmo péndulo			
Sub casos de simulación							
Sub caso 1		Sub caso 2		Sub caso 1		Sub caso 2	
8 slots disponibles		16 slots disponibles		8 slots disponibles		16 slots disponibles	
Casos de tercer nivel		Casos de tercer nivel		Casos de tercer nivel		Casos de tercer nivel	
Caso a	Caso b	Caso a	Caso b	Caso a	Caso b	Caso a	Caso b
1,25	2,5	1,25	2,5	1,25	2,5	1,25	2,5
Gbps	Gbps	Gbps	Gbps	Gbps	Gbps	Gbps	Gbps

2. EJECUCIÓN DE LAS SIMULACIONES

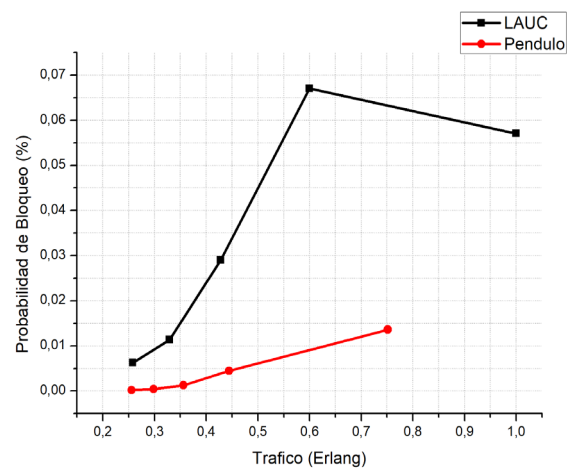
Las simulaciones realizadas en el IDE OMNeT++ arrojaron los siguientes resultados de probabilidad de bloqueo. Un resumen de los parámetros de simulación es mostrado en la Tabla II.

Tabla II. Resumen de los parámetros de simulación

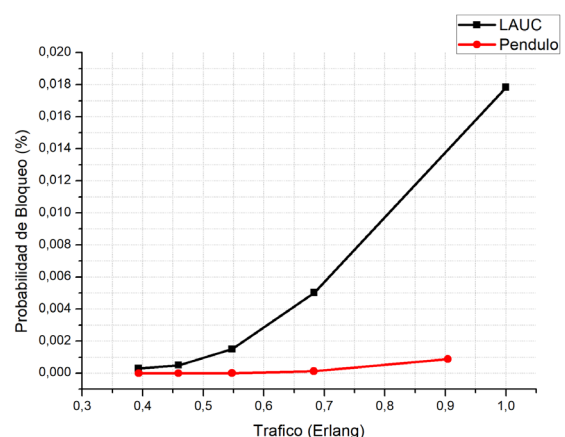
Parámetro	NSFNet con LAUC	NSFNet con péndulo
Algoritmo de enrutamiento	Dijkstra	Dijkstra
Numero de slots	8–16	8–16
Tiempo de simulación	80 ms	80 ms
Tasa de bits	1,25 Gbps – 2.5 Gbps	1,25 Gbps – 2.5 Gbps
Algoritmo de asignación de espectro	LAUC adaptado	Péndulo

Análisis de la probabilidad de bloqueo

La probabilidad de bloqueo (P_b) es la relación entre la cantidad de paquetes generados por las fuentes de datos y la cantidad de paquetes que se perdieron sin importar el motivo y que nunca alcanzaron su destino. A continuación, se muestran los resultados obtenidos para los casos de simulación expuestos en la Tabla I.

Fig. 7. P_b , red NSFNet con 8 slots, 1.25 Gbps.

En la Fig. 7 se observa que para cantidades intermedias de tráfico (0.4 – 0.7 erlangs) el comportamiento del algoritmo péndulo es mejor que el LAUC, comportándose de manera más estable y sin presentar pendientes elevadas que lo conduzcan a mayores probabilidades de bloqueo, como si lo hace el LAUC. Mientras que la probabilidad de bloqueo máxima del péndulo es de alrededor de 0.015%, la P_b del LAUC llega hasta cerca de 0.067%.

Fig. 8. P_b , red NSFNet con 8 slots, 2.5 Gbps.

DOI: <https://doi.org/10.21501/21454086.3145>

En la Fig. 8 se observa que el algoritmo péndulo es capaz de mantener una tasa de Pb de cero por ciento para valores inferiores de 0.5 erlangs y aumenta solo hasta cerca del 0.005% para tráfico muy alto, mientras que el algoritmo LAUC presenta valores mucho mayores, alcanzando pérdidas de hasta 0.018%.

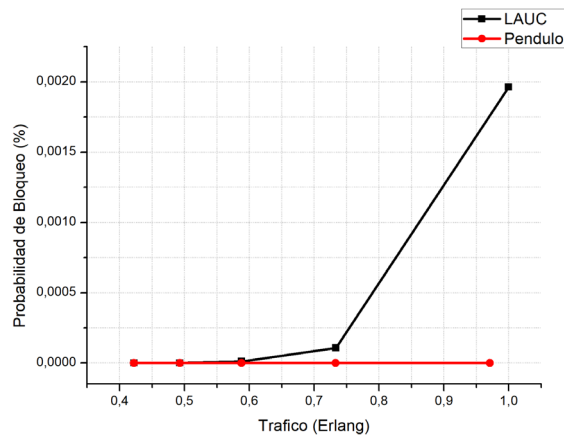


Fig. 9. Pb, red NSFNet con 16 slots, 1.25 Gbps.

En la Fig. 9 se observa que el algoritmo péndulo presenta un comportamiento ideal en el que para cualquier carga de tráfico mantiene una probabilidad de bloqueo de 0.00% mientras que el algoritmo LAUC, después de 0.6 erlangs empieza a presentar una Pb que está cerca del 0.002%.

Se observa en la Fig. 10 que el aumento de la cantidad de slots disponibles y la tasa de velocidad de 2.5 Gbps permite que el comportamiento de ambos algoritmos sea ideal, puesto que disponen de un gran ancho de banda y de una rápida liberación de los recursos, permitiendo que ningún paquete sea desechado. Para este caso, el uso de uno u otro algoritmo es indiferente, dado que sin importar el volumen de tráfico en la red, ambos algoritmos van a garantizar que todos los paquetes de información dispongan de recursos suficientes para ser transmitidos.

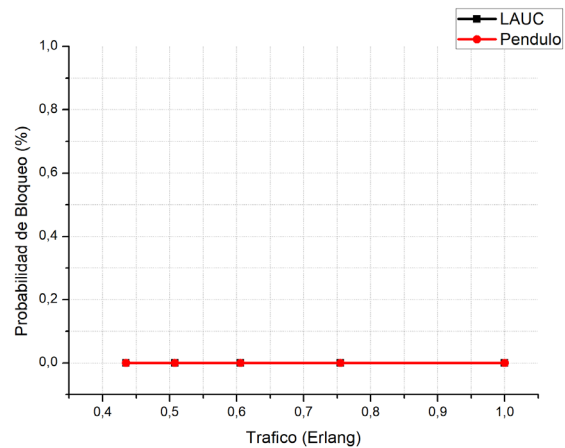


Fig. 10. Pb, red NSFNet con 16 slots, 2.5 Gbps.

Análisis del retardo de extremo a extremo

El retardo de extremo a extremo (EED) es un valor que mide el tiempo desde que un paquete es generado en la fuente de datos hasta que es entregado a su destinatario. A continuación, se muestran los resultados obtenidos para los casos de simulación expuestos en la Tabla 1.

En la Fig. 11 se observa que ambas gráficas parten de un valor mínimo de retardo que está dado por las características inherentes a toda red de datos, como por ejemplo, el tiempo que le toma a un paquete recorrer los enlaces entre nodos y los tiempos de procesamiento y establecimiento de los dispositivos de red. Puede verse que el retardo de los paquetes es menor cuando se usa el algoritmo péndulo, manteniéndose en el valor mínimo posible hasta cerca de los 4.5 erlangs de tráfico y alcanzando los 5.5 ms de retardo para una cantidad elevada de tráfico.

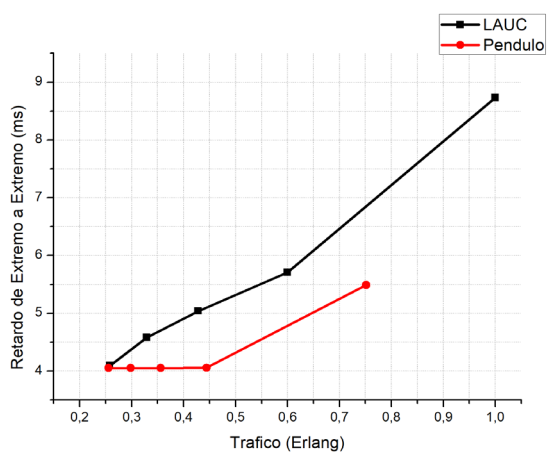


Fig. 11. EED, red NSFNet con 8 slots, 1.25 Gbps.

Se observa en la Fig. 12 que para una red con 8 slots de ancho de banda y una velocidad de 2.5 Gbps el retardo generado por el algoritmo péndulo es el mínimo posible, mientras que el algoritmo LAUC llega a presentar un retardo de cerca de 4.55 ms, aunque, para valores medios de tráfico, el comportamiento de ambos algoritmos es prácticamente el mismo.

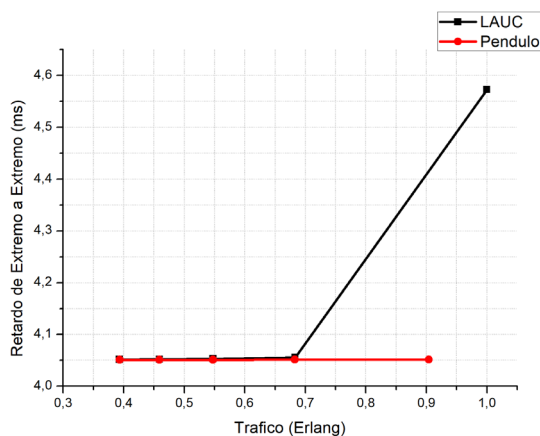


Fig. 12. EED, red NSFNet con 8 slots, 2.5 Gbps.

En la Fig. 13 se observa que el algoritmo péndulo no agrega ningún retardo adicional, manteniendo el retardo de extremo a extremo en un valor cercano al mínimo posible; esto es debido a que la grilla cuenta con la cantidad de slots suficientes para garantizar la atención a los paquetes entrantes. El algoritmo LAUC se comporta bastante bien para cantidades intermedias de tráfico, pero para valores elevados, y a pesar de que cuenta con suficientes slots, no es capaz de realizar una gestión eficiente de los recursos, por lo que algunos paquetes son puestos en cola, llevando a retardos de cerca de 4.2 ms, mientras que el retardo del algoritmo péndulo se mantiene cerca de 4.05 ms.

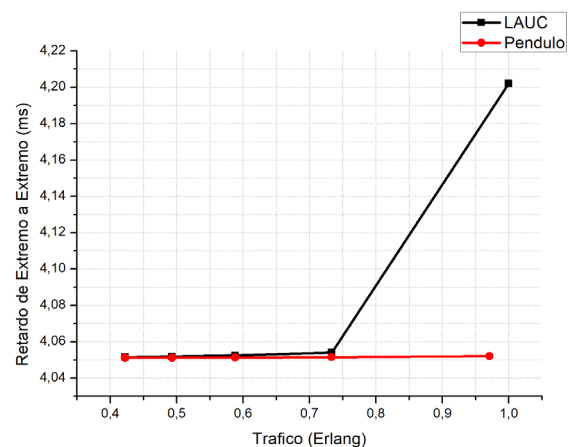


Fig. 13. EED, red NSFNet con 16 slots, 1.25 Gbps.

En la Fig. 14 se observa que para una velocidad de 2.5 Gbps, la red se comporta aun mejor que en el caso anterior, gracias a la rápida liberación de los recursos, y el retardo se mantiene en valores muy bajos para ambos algoritmos, pero con un mejor desempeño para el algoritmo péndulo que alcanza un retardo máximo cercano a 4.0512 ms, mientras que el LAUC llega hasta cerca de 4.0515 ms.

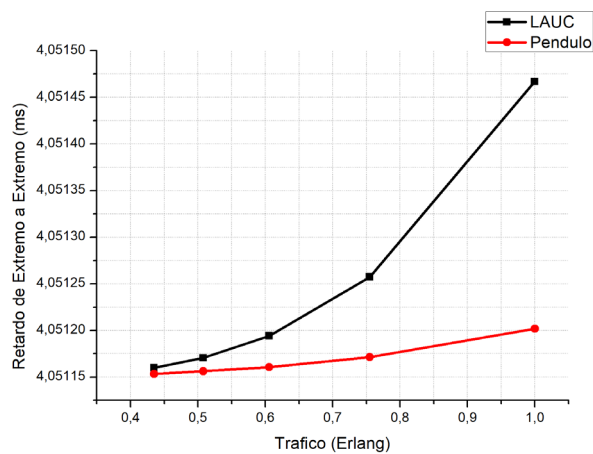


Fig. 14. EED, red NSFNet con 16 slots, 2.5 Gbps.

3. TRABAJOS FUTUROS

Como continuación de esta línea de investigación, son propuestos los siguientes posibles trabajos futuros:

- Adaptación y análisis del algoritmo péndulo para su operación en una red óptica convencional con tecnología RWA.
- Análisis del impacto de velocidades de datos superiores sobre los algoritmos estudiados.
- Diseño e implementación de un módulo cognitivo capaz de elegir entre dos o más algoritmos para la gestión de recursos de red.
- Adaptar el algoritmo de péndulo para la búsqueda y asignación de rutas en una red óptica.

4. CONCLUSIONES

- Para redes con poca disponibilidad espectral o baja tasa de bits, el algoritmo péndulo es una muy buena alternativa, pero a medida que los recursos de red aumentan, el algoritmo LAUC también se ubica en los parámetros aceptables.

- Para las redes ópticas actuales (distribuidas) el algoritmo péndulo es una gran alternativa para asignar recursos dado que realiza una gestión muy eficiente del espectro disponible, mejorando parámetros como la probabilidad de bloqueo y el retardo extremo a extremo.
- Para redes con alta disponibilidad espectral y tasas de transmisión alta, el uso de uno u otro algoritmo es prácticamente indiferente.
- OMNeT++ se considera como una elección acertada en cuanto a las características que brinda para la construcción de modelos de simulación. Sin embargo, posee una curva de aprendizaje alta, ya que exige un conocimiento en programación orientada a objetos en C++ y requiere de un proceso de familiarización fuerte.

5. CONFLICTO DE INTERESES

Los autores declaran la inexistencia de conflicto de interés con institución o asociación comercial de cualquier índole.

6. REFERENCIAS

- [1] J. P. Elbers, and A. Autenrieth, "From static to software-defined optical networks," in 16th International Conference on Optical Network Design and Modelling (ONDM), Colchester, UK. doi: 10.1109/ONDM.2012.6210207
- [2] A. Castro, L. Velasco, M. Ruiz, M. Klinkowski, J. P. Fernández-Palacios, and D. Careglio, "Dynamic routing and spectrum (re)allocation in future flex-grid optical networks," *Computer Networks*, vol. 56, no. 12, pp. 2869–2883, 2012. doi: 10.1016/j.comnet.2012.05.001

- [3] C. A. Castaño Salazar, y F. Amaya-Fernández, "Commutación de paquetes en el dominio óptico," *Revista en Telecomunicaciones e Informática*, vol. 2, no. 4, pp. 25–42, 2012. Recuperado de <https://revistas.upb.edu.co/index.php/telecomunicaciones/article/download/3303/2904>
- [4] J. G. López Perafán, "Diseño de métodos cross layer cognitivos para redes de comunicación óptica de ráfagas (OBS)," Tesis Doctoral, Departamento de Telecomunicaciones, Universidad del Cauca, 2014. Disponible en: http://www.unicauca.edu.co/doctoradoce/publicaciones/Monografia_Lopez.pdf
- [5] S. V. Kartalopoulos, *Introduction To Dwdm Technology*, First edit., no. 6387. Piscataway, Nj, USA: IEEE Press, 2000.
- [6] L. N. Binh, *Advanced Digital Optical Communications*, First edit. Munich, Germany: CRC Press, Taylor & Francis Group, 2015.
- [7] P. M. Pereira, "Redes Ópticas Elásticas," Tesis de pregrado, Escola de Engenharia de São Carlos, Universidade de São Paulo, 2013.
- [8] A. A. Garcia, "Elastic Spectrum Allocation In Flex-grid Optical Networks," Tesis de pregrado, Optical Communications Group, Universitat Politècnica de Catalunya, 2012. Disponible en: <https://upcommons.upc.edu/bitstream/handle/2099.1/16481/MEMORIA.pdf?sequence=4&isAllowed=y>
- [9] A. F. Sevilla Majin and E. C. Zúñiga Quisoboní, "Análisis comparativo del desempeño de algoritmos RSA y RWA sobre una red óptica basada en la topología NSFNET," Tesis de pregrado, Departamento de Telecomunicaciones, Universidad del Cauca, 2017. Disponible en: <http://repositorio.unicauca.edu.co:8080/xmlui/handle/123456789/1573>
- [10] ITU-T, "G.694.1 (02/2012), Spectral grids for WDM applications: DWDM frequency grid," Ser. G.694.1, pp. 1–16, 2012.
- [11] A. Salas, "Acerca del algoritmo de dijkstra," arXiv preprint arXiv:0810.0075, October 2008.
- [12] J. F. Chicano, "Metaheurísticas e Ingeniería del Software," Tesis Doctoral, Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, 2007. Disponible en: <http://neo.lcc.uma.es/tesis/PhD-Chicano07.pdf>
- [13] J. Brito Santana et al., "Metaheurísticas: Una revisión actualizada", Grupo de Computación Inteligente, Universidad de La Laguna, 2004. Disponible en: <http://www-eio.upc.es/personal/homepages/elena/web/mhreport.pdf>
- [14] Qusay. Mahmoud, *Cognitive Networks: Towards Self-Aware Networks*, Qusay H. M., vol. 91. Guelph, Canada: Wiley, 2017.
- [15] C. Fortuna, and M. Mohorcic, "Trends in the development of communication networks: Cognitive networks," *Computer Networks*, vol. 53, no. 9, pp. 1354–1376, 2009. doi: 10.1016/j.comnet.2009.01.002
- [16] H. Arslan, *Cognitive Radio, Software Defined Radio, and Adaptative Wireless Systems*, First. Tampa, FL, U.S.A: Springer, 200.