

Algoritmos de Routing en Redes Totalmente Ópticas

Francesc Guitart,

Resumen—Los usuarios y las nuevas aplicaciones telemáticas requieren, cada día más, un mayor ancho de banda. Esto empuja a las operadoras a mover ficha y mejorar sus redes *backbone* incrementando su capacidad. La evolución de las redes de telecomunicaciones ofrece nuevas tecnologías para cumplir estos nuevos propósitos. En esta dirección las redes ópticas se han erguido como la clave para el futuro. Tanto es así, que estas tienden a deshacerse de elementos no ópticos intermedios. Ahí es donde nacen las redes totalmente ópticas y tratan de ofrecer un servicio adaptado a las exigencias actuales.

Este documento engloba los problemas con los que nos enfrentamos a la hora de enrutar información dentro de este tipo de redes, también propone algunas soluciones aplicables en los diferentes tipos de infraestructuras que podemos encontrar. El documento se organiza como sigue: una primera parte donde se introducen los conceptos básicos de las redes totalmente ópticas, así como los elementos que las forman, seguidamente se divide el problema del routing en dos problemas básicos y finalmente se explican algunos algoritmos para el encaminamiento, propuestos por otros autores.

Index Terms—Redes de Banda Ancha, Algoritmos de Routing, Redes Totalmente Ópticas, RWA, WDM.

I. INTRODUCCIÓN

LA tecnología para poder ofrecer altas prestaciones en las redes ópticas es la WDM (*Wavelength Division Multiplexing*). La clave de esta tecnología radica en ofrecer diferentes *lambdas* (longitudes de onda) para incrementar la capacidad en un canal de fibra convencional usando luz procedente de un láser o un LED. Esto permite mejorar la infraestructura sin necesidad de grandes cambios. La escalabilidad de esta tecnología llega a permitir el transporte de cualquier tipo de señal al usuario (ATM, SONET/SDH, IP, etc.).

Para ser más fieles a la realidad, y basándonos en la rentabilidad que los ISP pretenden alcanzar, parece estar claro que la tecnología a usar en el futuro sea DWDM (*Dense Wavelength Division Multiplexing*). A diferencia del sistema WDM convencional, en este caso todas las portadoras ópticas viajan por la fibra con separaciones inferiores a 1 nm. Algunos autores se han puesto de acuerdo al denominar al *IP over DWDM* la puerta hacia las redes Terabit.

El uso de las tecnologías WDM para redes totalmente ópticas, elimina las conversiones óptico-eléctricas favoreciendo a la rapidez en la conmutación y evitando los cuellos de botella. Nos encontramos ante una simplificación enorme de la conmutación mediante WDM ya que reconocemos la identificación del canal sin necesidad de cabezas. Es evidente que según la arquitectura descrita, la información viajará por canales definidos por una *lambda*.

F. Guitart; Ingeniero Técnico en Informática de Sistemas por la Universitat de Lleida y estudiante del Máster de Ingeniería Telemática en Universitat Politècnica de Catalunya, Barcelona, España. Phone: +34 646465373, e-mail: guitart.francesc@gmail.com

A estos caminos los llamaremos *lightpaths*. Un *lightpath* se define como el camino entre dos nodos (no necesariamente adyacentes) creado por la asignación de una longitud de onda. Para poder crear un *lightpath* entre dos nodos, debemos tener al menos una longitud de onda común libre en todos los enlaces que conforman el camino. Un *lightpath* no requiere conversiones O/E, procesamiento de datos ni buffering en nodos intermedios, pero sí la dotación de inteligencia para poder asignar los recursos de una manera óptima o al menos tan buena como nos sea posible.

A. Componentes en un sistema DWDM

Optical Amplifiers: Elemento activo que se usa para amplificar la señal que llega débil y distorsionada con el objetivo de generar buena señal. No convierte la señal en pulsos eléctricos. La propiedad de amplificación se realiza dopando un pequeño filamento de fibra con un metal como el Erblio. Un *Erbium Doped Fiber Amplifier* (EDFA) es un amplificador comúnmente utilizado en la mayoría de redes.

Wavelength Converters: Se utilizan para convertir información en una longitud de onda de entrada en otra posible *lambda* de salida. Anteriormente, se ha dicho que un *lightpath* no requiere de conversiones, esto no quiere decir que no sean posibles. Más adelante se verá que en algunos casos es beneficioso el uso de estos elementos de red. Esta conversión puede ser óptico-eléctrica o totalmente óptica. En el caso que nos concierne, cuando hablemos de este tipo de conversores, carecerán de la conversión eléctrica.

Wavelength Multiplexer/Demultiplexer: Los MUX/DeMUX son elementos activos que se usan para quitar o reinsertar selectivamente canales individuales sin tener que regenerar todos los canales WDM. Los canales que tienen que ser añadidos o quitados pueden ser asignados o preconfigurados automáticamente basándose en el tipo de implementación. La primera se llama *Fixed WADM* y la última *Re-configurable WADM*.

Dispersion Compensating Fiber: La dispersión cromática es uno de los efectos más temidos en el uso de redes ópticas. Los *Dispersion Compensating Fiber* son unos elementos pasivos, que intentan compensar la dispersión acumulada. Dada la ineficiencia de estos elementos en según que condiciones otros elementos pueden ser añadidos (DCG y FBG).

La topología física de una red DWDM puede ser en forma de anillo o en forma de red *mesh* dependiendo del tipo de conexión entre los diferentes nodos. Existe el concepto de topología virtual que indica como se conectan los

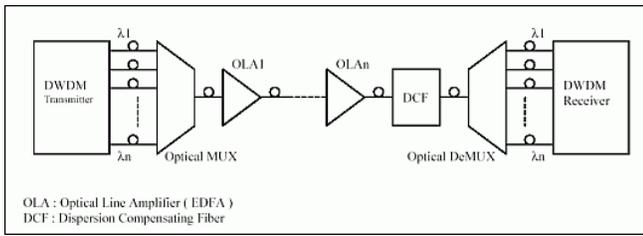


Figura 1. Sistema DWDM

nodos entre sí, mediante el uso de *lightpaths*. El concepto de topología virtual surge del punto de vista lógico de la conexión y no de la estructura física. Un algoritmo eficiente debe ser diseñado para calcular una topología virtual óptima basándose en el patrón adecuado de tráfico.

II. EL PROBLEMA DEL *Routing* Y *Wave-Length Assignment*

Abandonamos las antiguas arquitecturas ópticas, donde el ancho de banda se estrecha en las conversiones de los datos en elementos de almacenamiento eléctricos, los cuales reducen notablemente el rendimiento de todo el sistema. Como se puede sospechar, esto no será ni mucho menos gratuito, ya que como se ha anticipado anteriormente, deberemos dotar los nodos intermedios de mayor inteligencia para poder establecer los caminos de luz entre nodos, y dibujar así una red sin ningún elemento eléctrico.

Construir un camino de un nodo A a un nodo B, requiere asignar una lambda en cada uno de los nodos intermedios para este *lightpath* dado. Esto quiere decir que tendremos que asegurarnos de que existe una misma lambda libre en todos los nodos intermedios para poder hacer viable la comunicación. A esta restricción se le llama *wavelength continuity constraint*. Esta restricción puede llevarnos a una utilización ineficiente de las fibras y una alta probabilidad de que no quede ninguna longitud de onda disponible para construir nuestro camino. En el momento que todas las lambdas de un *link* o enlace entre dos nodos estén totalmente asignadas, este *link* queda bloqueado para la interconexión de nodos. Vemos que la utilización eficiente de las lambdas se convierte en un aspecto crucial.

Alternativamente podemos pensar en la conversión de longitudes de onda. El uso de los *wavelength converters* para cambiar la lambda de salida para un *lightpath* entrante. Este tipo de conversión suprime la *wavelength continuity constraint* y de hecho disminuye la probabilidad de bloqueo significativamente. Actualmente este tipo de dispositivos conversores tienen un precio muy alto, y dotar toda una red de nodos conversores eleva mucho los presupuestos. Mucha de la investigación llevada a cabo recientemente, tiene en cuenta este aspecto y se centra en redes con una capacidad de conversión limitada. Estas redes tienen solo un subgrupo de nodos capaces de cambiar las lambdas de los *lightpaths* a lo largo del camino, y se las acostumbra a llamar redes con *sparse wavelength conversion* o conversión limitada. Subramaniam *et al.* han mostrado que con el uso de conversión limitada, también se consiguen muy buenos

resultados [6]. Podemos ver que se añade otro problema en la construcción de una topología de red totalmente óptica y un tipo de tráfico entrante, ya que ¿cómo puedo colocar los conversores para minimizar la probabilidad de bloqueo total? Existen algunos algoritmos óptimos para redes con topología de bus y de anillo. Se considera que una solución óptima para una topología arbitraria constituye un problema difícil de solucionar.

Llegados a este punto vemos que al problema de asignar un camino a través de la red a un flujo de datos (lo que comúnmente llamamos encaminamiento o *routing*) según diferentes tipos de métricas, se nos añade el problema de la asignación de una lambda a cada camino. Estos dos problemas se agrupan en uno más general, llamado *RWA Routing and Wavelength Assignment*. La aportación de una solución para el problema general ofrece una mayor eficiencia a todo el sistema, pero es muy habitual encontrar el planteamiento de los dos subproblemas por separado que cambiándolos al final, dan una solución general.

En la bibliografía podemos encontrar diferentes estudios sobre la solución al problema del RWA. Podemos afrontar el problema asumiendo dos tipos de tráfico:

Estático: El tráfico se conoce de antemano y se asume fijo, por lo tanto podemos determinar los caminos previamente. El objetivo a conseguir en este caso, será poder asignar *lightpaths* a todos los tráfico y minimizar el uso de lambdas.

Dinámico: El tráfico llega uno a uno aleatoriamente y termina en tiempos aleatorios. En este caso el objetivo es minimizar la probabilidad de bloqueo.

Y dependiendo del tráfico asumido podemos tratar los dos problemas:

- Heurística combinada: se considera el conjunto de todas las lambdas y se tratan ambos problemas al mismo tiempo.
- Heurística de selección de ruta previa: primero se escoge la ruta, y posteriormente se intenta asignar la longitud de onda más apropiada, de entre el subgrupo de las disponibles en el camino seleccionado.

Incluso en el esquema estático más simple, se transforma en un problema difícil de solucionar. En particular, la solución óptima sin *wavelength converters* se trata de un problema NP-completo, como se prueba en [3]. En la demostración se prueba que el problema equivale al coloreado de un grafo.

En teoría de complejidad computacional se dividen los problemas según el coste en cuanto a recursos para el computo de un algoritmo, en otras palabras, los recursos requeridos para obtener una solución. Así pues, si la relación del tamaño del algoritmo y el tiempo es polinómica, decimos que se trata de un problema P. Hoy en día un problema P puede ser tratable por una máquina determinista (las computadoras actuales). Los problemas que no pueden ser resueltos en un tiempo razonable por una máquina determinista (generalmente problemas con costes factoriales o combinatorios) se denominan NP. Sin entrar en más detalle, podemos concluir diciendo que los problemas NP-completos se trata de un subconjunto con ciertas

propiedades de los NP.

Cuando hablamos de heurística, nos referimos a utilizar algoritmos desarrollados utilizando métodos exploratorios en la búsqueda de un resultado final. Estos algoritmos nos darán resultados subóptimos, los cuales pueden funcionar en un tiempo razonable para no todo el conjunto de resultados posibles.

III. ALGORITMOS PARA *Wave-Length Assignment*

Muchos algoritmos para la asignación de lambdas han sido propuestos, y también muchos están siendo utilizados. Su clasificación puede hacerse en dos grupos[2]:

1. Algoritmos Combinatorios: Los algoritmos combinatorios o selección combinatoria, consideran la interdependencia entre la selección de diferentes *lightpaths*. Podemos redividirlos de nuevo, recalando en conceptos introducidos con anterioridad:

- a) Solución óptima: se intentan usar todas las posibles combinaciones de interdependencia. Es extremadamente difícil en términos de complejidad computacional (de hecho son problemas NP-Complejos como ya se ha explicado) y muy difíciles de aplicar en redes muy grandes. Por ejemplo, una búsqueda exhaustiva en profundidad hasta encontrar la mejor opción.
- b) Solución heurística: en este planteamiento, el algoritmo se va ejecutando, y a la vez se reducen las posibles lambdas a escoger. El proceso se va ejecutando hasta que no se puede reducir más el número de longitudes de onda asignables. Un ejemplo claro son los algoritmos genéticos. Estos algoritmos se desarrollan mimetizando alguna de las leyes de la evolución y selección natural. La idea se basa en encontrar primero de todo una solución que se pueda considerar legal para el problema. Esta se representa mediante un conjunto genes llamados cromosomas. Luego, se construye una población inicial de cromosomas legales. A continuación se prueba cuan correcto es cada cromosoma, y los mejores cromosomas son utilizados más tarde para producir una nueva generación de cromosomas que se adaptarán mejor que la población inicial.

También cabe la posibilidad de usar meta-algoritmos probabilísticos para la optimización del problema. Los algoritmos de Temple Simulado se inspiran en el proceso físico de enfriamiento controlado. A partir de un estado inicial, se eligen estados siguientes de manera aleatoria, si el resultado es mejor se avanza y si no, se permanece en el actual. La idea es que el problema vaya enfriándose, ya que como menor es la temperatura, la probabilidad de pasar a un estado erróneo es menor.

2. Algoritmos Secuenciales: estos son los más simples que encontramos. La selección de cada lambda se hace de manera secuencial. Esta técnica, no cambia el resultado obtenido con anterioridad, pero tiene en cuenta el resultado anterior. Están pensados para arquitecturas monofibra o multifibra. Vamos a ver una breve

selección [7] de los más usados, centrándonos en el FF (*First-Fit*) dada su sencillez de implementación, convirtiéndose así en uno (sinó el que más) de los más famosos:

A. *Random*

La lambda a escoger para el uso del *lightpath* es completamente aleatoria. Se basa en asignar de manera aleatoria una longitud de onda para un tráfico estático, de una lista donde existen las posibles lambdas. Si se puede satisfacer la condición de que esta lambda esté libre en todos los nodos de la red que pretende crearse, se asigna tal lambda; si no puede crearse se prueba otra. El algoritmo finalizará devolviendo una lambda o bien no devolviendo ninguna, siendo este el caso donde la comunicación queda bloqueada.

B. *First-Fit*

De entre todos los algoritmos para la asignación de longitudes de onda, el First-Fit es el más usado y el más fácil de implementar. Este algoritmo, trata de seleccionar una lambda de entre una lista ordenada ascendente de entre las posibles longitudes de onda a lo largo del camino. Para minimizar la probabilidad de bloqueo al establecer un nuevo *lightpath* trataremos de maximizar el reuso de lambdas en toda la red, podemos ver esto desde dos puntos de vista:

1. Si hay una lambda que no esté asignada a ningún *lightpath* en ningún nodo, podremos satisfacer la llegada de una demanda de conexión sea cual sea su camino.
2. Si tenemos una lambda λ_i que está asignada a un subgrupo $E_i \in E$, cuanto más grande sea este subgrupo E_i menos proporción de nuevos caminos podrán ser asignados para λ_i . Entonces, si asignamos la lambda de la nueva conexión al subgrupo con mayor nodos, aumentaremos las posibilidades de que nuevos caminos se puedan alojar utilizando otras λ .

El algoritmo 1 propuesto en [3] para este propósito. Primero describiremos las estructuras de datos que parecerán más tarde en dicho algoritmo:

- l***: petición de *lightpath*; un array que contiene los nodos que describirán el camino.
- wave***: array que determina la utilización de esta longitud de onda; $wave[i]$ es el número de links donde λ_i ha sido utilizada.
- alloc***: matriz 1/0; $alloc[i,j] = 1$ si λ_j esta siendo usada en el link i .

C. *Least-Used*

Este algoritmo selecciona la lambda menos utilizada, de esta manera trata de balancear el uso de las lambda en toda la red. Este esquema acaba por romper los *lightpaths* largos, ya que solo se asignará una lambda a los caminos considerados cortos. De hecho, este algoritmo actúa peor que el aleatorio, añadiendo información de más en la red para saber que lambdas están siendo utilizadas y escoger la de menor uso. Esto lleva al LU a ser un algoritmo no usado en la práctica.

Algorithm 1 First-Fit

```

establish(l):
  max = -1
  for i to n do
    feasible = 1
    for all links k on l's path do
      feasible and alloc[l[i],k]
    end for
    if feasible and max < wave[i] then
      max = wave[i], w = i
    end if
  end for
  if max > 0 then
    for i = 0 to len do
      alloc[l[i],w] = 1
    end for
    wave[w] = wave[w] + len
  else
    the lightpath is blocked
  end if
terminate(l,w):
  for i = 1 to l do
    alloc[l[i],w] = 0
  end for
  wave[r] = wave[r] - len

```

D. Most-Used

Esta heurística es la contraria a Least-Used. Selecciona la lambda más utilizada en la red. Actúa mejor que LU, a pesar de que la información cursada por la red, el almacenamiento de información y el costo del cálculo es el mismo que en LU. También actúa mejor que FF al agrupar las conexiones en menos lambdas.

E. Min-Product

Pensado para redes multi-fibra. Para el caso de redes mono-fibra MP se comporta como First-Fit. El objetivo del algoritmo MP es juntar las longitudes de onda en las fibras, de esta manera se minimiza el uso de fibras en la red. El algoritmo, primeramente computa

$$\prod_{l \in \pi(p)} D_{lj},$$

donde,

- L numero de links.
 - W numero de lambdas por fibra.
 - $\pi(p)$ conjunto de links que conforman el path p .
 - D es una matriz $L \times W$ y D_{lj} indica el numero de enlaces asignados en el link l y lambda j .
- para cada j , i.e, $1 \leq j \leq W$. Sea X el conjunto de lambdas j que minimizan el valor anterior, entonces MP escoge la lambda de menor número en X .

F. Least-Loaded

Se trata de otra heurística pensada para redes multi-fibra. LL trata de seleccionar la lambda que tiene la mayor

capacidad residual en el link más cargado a lo largo de la ruta p . Cuando se utiliza en mono-fibra, la capacidad residual es 1 ó 0, entonces la heurística escoge la lambda con menor índice con capacidad residual 1. Entonces en redes mono-fibra, actúa como FF. LL selecciona la lambda con menor índice j en S_p que consigue

$$\max_{j \in S_p} \min_{l \in \pi(p)} (M_l - D_{lj}),$$

donde,

- M_l es el numero de fibras en el link l .

G. MAX-SUM ($M\Sigma$)

$M\Sigma$ fue inicialmente propuesta para redes multi-fibra, pero también se puede aplicar para redes mono-fibra. $M\Sigma$ considera todos los posibles caminos (lightpaths con rutas preseleccionadas) en la red y trata de maximizar las capacidades restantes después del establecimiento de caminos. Se asume que la matriz de tráfico se conoce de antemano, y que la ruta para cada conexión está preseleccionada.

Introduciremos nueva notación para describir la heurística. ψ denota el estado de la red que especifica los lightpaths existentes. En $M\Sigma$, la *link capacity* en el link l y longitud de onda j en el estado ψ , $r(\psi, l, j)$, se define como el número de fibras en las cuales la lambda j no se usa en el link l ,

$$r(\psi, l, j) = M_l - D(\psi)_{lj},$$

donde $D(\psi)$ es la matriz D en el estado ψ .

La *path capacity* $r(\psi, p, j)$ de la lambda j es el número de fibras en las cuales la longitud de onda j está disponible en el link más congestionado a lo largo del camino p ,

$$r(\psi, p, j) = \min_{l \in \pi(p)} r(\psi, l, j).$$

La *path capacity* del camino p en el estado ψ es la suma de las capacidades de todos los caminos en todas las longitudes de onda,

$$R(\psi, p) = \sum_{j=1}^W \min_{l \in \pi(p)} c(\psi, l, j).$$

Sea:

- $\Omega(\psi, p)$ el conjunto de todas las lambdas posibles que están disponibles para el lightpath que es enrutado por el camino p .
- $\psi'(j)$ el siguiente estado de la red si la lambda j es asignada a la conexión.

$M\Sigma$ escoge la lambda j que maximiza la cantidad

$$\sum_{p \in P} R(\psi'(j), p),$$

donde P es el conjunto de todos los caminos potenciales para la petición de respuesta en el estado actual. Una vez se ha establecido el camino para la conexión entrante, el estado de la red se actualiza y una nueva petición de conexión puede ser procesada.

IV. ALGORITMOS DE *Routing* O ENCAMINAMIENTO

Los algoritmos de routing de encargan de decidir en qué línea de salida debe ser asignado un paquete que viene por la entrada. Estos pueden ser agrupados en dos subclases (se puede ver que dependerá de la asunción del tráfico entrante)[2]:

1. Algoritmos no-adaptativos: no basan sus decisiones de routing en métricas y estimaciones del tráfico actual o topología. Por contra, la elección de una ruta entre dos nodos es calculada de antemano, fuera de línea y descargada a los routers durante la secuencia de arranque, cuando la red esta iniciándose. Esta metodología también es llamada *Static Routing*. Por ejemplo, los algoritmos que se basan en encontrar el camino más corto, como el algoritmo de Dijkstra. Se basa en calcular el camino mas corto, computando los parámetros de coste. La idea es construir un subgrafo de la red actual, donde cada nodo del grafo es un router, y los arcos entre nodos representan los enlaces físicos. Para encontrar una ruta entre un par de nodos dados, el algoritmo escoge el camino más corto entre dos nodos del grafo. El camino es el más corto, considerando los parámetros de coste en base al número de saltos o la distancia geográfica en kilómetros. Otro es el algoritmo *k-shortest paths*, que calcula más de un camino óptimo como resultado a cada petición.

Podemos añadir una subclase en este tipo de algoritmos. Se trata de los denominados *fixed-alternate route*. La idea detrás de este tipo de algoritmo es mantener una tabla en cada uno de los nodos donde se ordene una serie de rutas posibles hacia los diferentes destinos. Así pues podemos tener el camino más corto, el segundo camino más corto, el tercer camino más corto, y así consecutivamente. Existe una ruta desde el nodo s hasta el nodo d definida en primera posición en la tabla de rutas hacia s en el nodo d . Para que una ruta sea alternativa, tiene que estar construida por *links* diferentes a los que pasa la primera ruta.

Cuando llega una nueva petición de conexión, el nodo origen trata de establecer la conexión por la primera ruta prefijada de la lista. Si no hay posibilidad de asignar una lambda para esta conexión, se tratará de proceder de la misma manera para la siguiente ruta de la tabla, y así consecutivamente hasta que haya una conexión establecida o se retorne el aviso de bloqueo para esa conexión.

2. Algoritmos adaptativos: se basan en cambiar sus decisiones de routing dependiendo de los cambios de topología y de tráfico.

El uso de algoritmos adaptativos esta pensado para el caso de redes con topología arbitraria y con un tráfico dinámico. Los no-adaptativos funcionan bien para redes pequeñas, ya que el calculo de las rutas se transforma en una tarea ardua en cuanto el tamaño de la red aumenta.

Algunos estudios reflejan que el uso de los algoritmos adaptativos son mucho más eficientes que los estáticos [7]. Vamos a ver a continuación dos algoritmos

propuestos por diferentes autores, nos remitimos a los documentos originales para la evaluación de resultados así como el calculo de probabilidades de bloqueo. A pesar que constituyen una pieza básica en el análisis de algoritmos, estos resultados son demasiado extensos dadas las diversas simulaciones y los resultados obtenidos.

A. Algoritmo *WLCR-FF*

El propósito del algoritmo *Weighted Least-Congestion Routing and First-Fit Wavelength Assignment* [4], no solo es mostrar que los algoritmos adaptativos obtienen mejores resultados que los estáticos, sino que los autores también tratan el problema de la colocación de *wavelength converters* en la red. Para ello proporcionan además un algoritmo de colocación de conversores. Este algoritmo propuesto, se basa en considerar conjuntamente la distribución de lambdas libres con la longitud de los caminos. Explicaremos la lógica del algoritmo y nos remitimos directamente al artículo original para una evaluación de los resultados.

La clave de este algoritmo se basa en el cálculo del peso de la rutas según un parámetro. Para ello, cada nodo parte con un conjunto de rutas pre-calculadas para cada destino. Según hemos dicho, el dinamismo de este algoritmo se basa en que si el estado de la red cambia, también lo tienen que hacer las rutas por las cuales se encaminarán los datos y posteriormente se deben asignar las longitudes para cada camino. Al ser un algoritmo adaptativo, este tratará de mantener una probabilidad de bloqueo baja.

Sean $\{R_a^{(1)}, R_a^{(2)}, \dots, R_a^{(M_a)}\}$ el conjunto de ruta precalculadas por el nodo a . Para la llegada de una petición de conexión, el nodo a deberá escoger de entre un número M_a de rutas candidatas. El algoritmo WLCR-FF tomará una decisión según el peso de cada ruta. Este peso viene determinado por la función $W(R)$, que se define de la siguiente manera:

$$W(R) = \frac{F(R)}{\sqrt{h(R)}}$$

donde se define:

- $F(R)$: retorna el número de lambdas libres en la ruta R .
- $h(R)$: longitud de la ruta R .

Después del cómputo de los pesos, se elige el camino con peso máximo. Posteriormente se asignarán las lambdas mediante FF.

Como vemos, el núcleo del algoritmo se centra en el cálculo del peso de las rutas. Los autores se basan en la siguiente observación: para tomar la decisión de la mejor ruta hay que tener en cuenta el número de lambdas libres y la longitud de las rutas.

En el caso de que no haya conversores, es lógico que la ruta con mas lambdas libres sea la mejor opción y que además esta ruta no sea demasiado larga. La intuición nos dice que como más larga es la ruta, más probabilidades tendremos que las lambdas estén siendo usadas. El algoritmo funcionará bien para redes sin conversores.

Por otra parte, si ponemos el caso de una red convertible, puede ser que una ruta larga tenga mas lambdas libres

al encontrar un conversor por el camino. Esto nos lleva a una ineficiencia por un lado, y por otro lado nos aumenta la probabilidad de bloqueo. La misión de la función de calculo de peso, es ponderar la rutas proporcionalmente con la cantidad de lambdas e inversamente proporcional con la longitud de la ruta.

En [1] Ahmad *et al.* proponen una mejora a este algoritmo modificando el cálculo de la función de peso de la siguiente forma:

$$W(R) = \frac{F(R)}{\sqrt{h(\overline{R})} \times Cr(R)^2}$$

$$Cr(R) = \sum_{i=1}^n Cr(L_i)$$

$$Cr(L) = \sum (k - Rank_R + 1)$$

Para cualquier ruta que incluye el enlace L Donde:

- n es el numero total de links en la ruta n.
- L_i es el iésimo enlace de la ruta R.
- $Cr(R)$ peso crítico de la ruta R.
- $Cr(L)$ peso crítico del enlace L.
- k numero de rutas alternativas.
- Rank es la posición en la lista ascendente según número de saltos respecto a las rutas alternativas.

El propósito de esta mejora, es calcular el peso crítico de la ruta. Esto se puede entender como un índice de cuan usada esta ruta puede ser por los otros nodos. Si una ruta tiene este índice muy alto, intentaremos encontrar otra alternativa, teniendo en cuenta aún el número de saltos.

B. Algoritmos LCLNR y DTWR

El objetivo de los autores de estos algoritmos, vuelve a ser el mismo: maximizar la utilización de las longitudes de onda y reducir la probabilidad de bloqueo de la red. Proponen un *Dynamic Routing Algorithm*[5] que esta compuesto de dos subalgoritmos: *Least Congestion with Least Nodal-degree Routing algorithm* (LCLNR) (algoritmo 2) y *Dynamic Two-end Wavelength Routing algorithm* (DTWR)(algoritmo 3).

La notación para seguir el algoritmo es la siguiente:

- $N = \{n_i\}$ conjunto de routers.
- s nodo origen.
- d nodo destino.
- n_s son los nodos adyacentes a s.
- n_d son los nodos adyacentes a d.
- $L = \{l_i\}$ conjunto de enlaces en la red, donde $i \in \{1 \dots L\}$.
- l_{mn} es el enlace entre n_m y n_n .
- W número de lambdas por fibra.
- $K = \{K_{sd}^i\}$ lista de *k-shortest paths* en orden ascendente en número de saltos.
- K_{sd}^i es el iésimo candidato para la ruta de s a d.
- $h_i = |K_{sd}^i|$ número de saltos para el camino K_{sd}^i .
- $w_{common}^{K_{sd}^i}$ número de lambdas en común en el camino K_{sd}^i .

- w_{sd}^i número de lambdas en común en el camino K_{sd}^i , donde $w_{sd}^i = |w_{common}^{K_{sd}^i}|$
- $w_x^{K_{sd}^i}$ denota que la lambda x ha sido escogida en el camino K_{sd}^i .
- w_{mn} número de lambdas disponibles en l_{mn} .
- w_{mn}^x conjunto de de lambdas libres en l_{mn} .
- $deg(n_i)$ es el grado en el nodo n_i
- $Path_{sd}$ es el camino determinado para ir de s a d.
- W_{mn} es la suma de todas las lambdas disponibles en todos los links entre n_m y sus nodos vecinos n_m .

El objetivo del algoritmo LCLNR, es evitar enrutar los datos por los enlaces congestionados, reduciendo así la probabilidad de bloqueo. Como en el ejemplo del WLCR-FF, todos los nodos tienen una lista de rutas pre-calculadas para cada destino (los autores proponen una variante del algoritmo de Dijkstra). El procedimiento será parecido al del WLCR-FF, ya que calculamos una función objetivo para todas las rutas definida como: $O(K_{sd}^i) = w/h$ donde se pondera el número de lambdas libres w con la longitud de la ruta en número de saltos h. La ruta con mayor número será la seleccionada. Los autores introducen aquí una variante, ya que si la función del peso retorna dos valores iguales para dos rutas diferentes, se calcula la suma de los grados de los nodos intermedios y se escoge la ruta con menor valor: $\sum_{i=n_s}^{n_d} deg(n_i)$.

La razón de esta variante, es que los autores han visto mediante pruebas y simulaciones, que para dos rutas con la misma longitud, aquella que tenga unos nodos más colapsados, será la que se bloquee con una probabilidad mayor.

Algorithm 2 LCLNR

Require: A physical topology $G(N, L)$, and a set of K pre-determined paths, $K_{sd}^i, i = 1 \dots k$, of connection $t = (s, d)$.

Ensure: $Path_{sd} = K_{sd}^i, i \in \{1 \dots k\}$.

Calculate the objective value consumed by path K_{sd}^i :
 $O(K_{sd}^i) = w_i/h_i, \forall i \in \{1 \dots k\}$

if $O(K_{sd}^i) = 0, \forall i \in \{1 \dots k\}$ **then**

The call is denied; **RETURN**

else

$Path_{sd} = \max_{i \in \{1 \dots k\}} O(K_{sd}^i), P = |Path_{sd}|$;

end if

if $P \geq 2$ **then**

Calculate: $Path_{sd} = \min_{p \in \{1 \dots P\}} \sum_{n_i \in Path_{sd}} deg(n_i), P = |Path_{sd}|$

if $P \geq 2$ **then**

randomly select a route $Path_{sd}$

end if

end if

The request is accepted. Assign a wavelength $w_x^{K_{sd}^i}$ to $Path_{sd}$ using First-fit technique; $w_x^{K_{sd}^i} = \min(w_{common}^{K_{sd}^i})$.

El algoritmo 3 solo se usará cuando una petición no pueda ser asignada por el algoritmo 2. El objetivo del algoritmo DTWR es maximizar la utilización de las lambdas cuando hay un bloqueo provocado por la falta de enlaces entre el nodo origen y sus vecinos, o el destino y sus vecinos. Pri-

mero de todo el algoritmo trata de averiguar si el bloqueo es debido a esta situación. Si es así, se genera un nuevo grafo $G'(N, L')$ a partir del grafo original $G(N, L)$ donde se retiran los nodos que cumplen $w_{sn_s} = 0$ y $w_{n_d} = 0$. Ahora mediante el algoritmo modificado de Dijkstra se recalculan las rutas desde $n_s - n_d$.

Algorithm 3 DTWR

Require: $|Path_{sd}| = 0$.

Ensure: K_{sd}^i .

```

while the call request is denied do
  Calculate  $w_{sn_s}, w_{n_d}, W_{sn_s}, W_{n_d}$ 
  if  $(W_{sn_s} = 0)$  or  $(W_{n_d} = 0)$  then
    The call request is blocked; RETURN
  else if  $\{w_{sn_s}^i, \forall i\} \cap \{w_{n_d}^j, \forall j\} = 0$  then
    The call request is blocked; RETURN
  else
    Remove those links  $w_{sn_s} = 0$  and  $w_{n_d} = 0$  on
     $G(N, L)$  and generate  $G'(N, L')$ 
  end if
  Calculate a set of paths  $K_{n_s n_d}^i$  by using the modified
  Dijkstra's shortest path algorithm,  $t' = (n_s, n_d)$ 
  if  $|K_{n_s n_d}^i| = 0$  then
     $|K_{sd}^i| = 0$ , the call request is blocked; RETURN
  else
     $K_{sd}^i \leftarrow K_{n_s n_d}^i$ , Go to LCLNR
  end if
end while

```

- [4] Xiaowen Chu, Bo li, and Zhensheng Zhang, "A Dynamic RWA Algorithm in a Wavelength-Routed All-Optical Network with Wavelength Converters"
- [5] Kungmeng Lo, Daryoush Habibi, Quoc Viet Phung, and Hoang Nghia Nguyen, "Dynamic Wavelength Routing in All-Optical Mesh Networks"
- [6] Suresh Subramaniam, Murat Azizoglu, and Arun K. Somani, "All-Optical Networks with Sparse Wavelength Conversion" IEEE/ACM Transactions on Networking, Vol. 4, No. 4, pages 544-557, August 1997
- [7] Hui Zangy and Jason P. Juez and Biswanath Mukherjee, "A Review of Routing and Wavelength Assignment Approaches for Wavelength-Routed Optical WDM Networks"

Francesc Guitart Nacido en Lleida, en 1986. En 2007 recibió el título de Ingeniero Técnico en Informática de Sistemas con una especialización en Aplicaciones Telemáticas, Seguridad e Internet. En 2007 empieza el Master en Ingeniería Telemática cursado en la UPC. Actualmente colabora como becario en el grupo de Redes Inalámbricas del Departamento de Ingeniería Telemática de la UPC.

V. CONCLUSIONES

Después de haber valorado el problema del RWA, y según las soluciones propuestas, podemos ver que la mejor solución para encaminar la información dentro de una red totalmente óptica depende de varios factores:

- Tamaño de la red (en numero de nodos).
- Topología de la red.
- Tipo de red (convertible o no).
- Tipo de tráfico.

Incluso después de haber considerado estos parámetros, es importante ver la red desde varios planos. Así pues, el ingeniero que dimensione la red puede pensar en un algoritmo genético que coloque de manera muy eficiente las lambda en la red, pero desde el plano del administrador un algoritmo secuencial y sencillo puede ayudarle a solucionar incidencias que puedan ocurrir. Desde el plano usuario, también es muy importante que sea cual sea el algoritmo general, no añada retrasos ni construya una red inestable.

REFERENCIAS

- [1] Jawwad Ahmad, S.M.H.Zaidi and Shahid Nawaz "Dynamic Routing in Wavelength Convertible WDM Networks"
- [2] Rambla Ahmad "Routing and Wavelength Assignment Algorithms in IP over DWDM Networks" Students Conference, ISCON '02. Proceedings IEEE.
- [3] Imrich Chlamtac, Aura Ganz, and Gadi Karmi. "Purely optical networks for terabit communication" INFOCOM '89. Proceedings of the Eighth Annual Joint Conference of the IEEE Computer and Communications Societies. Technology: Emerging or Converging, IEEE Publication Date: 23-27 Apr 1989