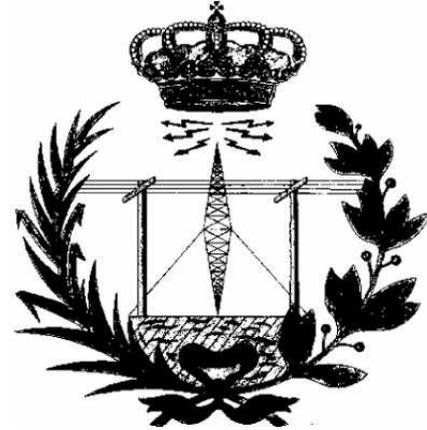


UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Escuela Técnica Superior de Ingeniería de Telecomunicación

Proyecto Fin de Carrera

**HERRAMIENTA *MATPLANWDM* PARA OPTIMIZACIÓN
EN REDES WDM DE CONMUTACIÓN DE *LIGHTPATHS***

D. Gonzalo J. Moreno Muñoz

Director: Dr. D. Pablo Pavón Mariño

Septiembre 2006

A mi madre y hermanas,



HERRAMIENTA *MATPLANWDM* PARA OPTIMIZACIÓN EN REDES WDM DE CONMUTACIÓN DE *LIGHTPATHS*

Autor	D. Gonzalo J. Moreno Muñoz - gonzalo.moreno@laavalancha.es
Director	Dr. D. Pablo Pavón - pablo.pavon@upct.es
Título	"Herramienta MatPlanWDM para optimización en redes WDM de conmutación de <i>lightpaths</i> "
Descriptores	Planificación de redes WAN Diseño de topologías virtuales en redes WDM Algoritmos heurísticos
Resumen	El presente Proyecto incluye el estudio del paradigma de las redes WDM en su vertiente teórica, y el diseño y planificación de topologías virtuales en su parte práctica. El problema de la complejidad se aborda mediante un conjunto de heurísticos propuestos en la literatura e implementados en Matlab. La integración de los algoritmos en un entorno gráfico GUI para su evaluación y pruebas en distintos escenarios, proporciona una herramienta inédita para la docencia en el diseño y planificación de las redes WDM.
Titulación	Ingeniería de Telecomunicación
Intensificación	Planificación y Gestión de Telecomunicaciones
Departamento	Tecnologías de la Información y las Comunicaciones
Fecha de presentación	Septiembre de 2006

Agradecimientos

En septiembre de 2005, el Dr. Pablo Pavón nos recibió en su despacho, a Ramón Aparicio y a mí, para exponernos las ideas que tenía, con respecto a nuestros proyectos fin de carrera. Aunque después de cinco años de carrera, el instrumental conceptual adquirido nos permitió comprender cuales eran los objetivos y el alcance de los PFCs, realmente la tecnología que se nos proponía como objeto de estudio, era desconocida. El conocimiento del medio óptico en nuestra titulación, no lleva asociado el estudio de este tipo de redes a niveles superiores, situación, por otra parte, que evidencia una carencia en el Plan de Estudios. Fue el entusiasmo de nuestro director, en las posibilidades de esta tecnología el acicate necesario para implicarnos de lleno en el estudio de las redes WDM.

De aquí parte, en primer lugar, mi agradecimiento personal a Pablo Pavón, por su rigor, su entrega y su dedicación. Pensar un poco más allá, arriesgarse en la toma de decisiones, no desfallecer jamás, y sobre todo el trabajo incondicional, han sido lecciones inolvidables. La tutorización de este Proyecto contó con un inconveniente adicional, la distancia. El desarrollo fundamental del Proyecto lo realicé en la Fachhochschule de St. Pölten (Austria), pero dirigido, exclusivamente, desde la Universidad Politécnica de Cartagena, por el profesor Pavón. Esto requería del director un esfuerzo extra, y su disponibilidad, a lo largo de nueve meses, fue realmente ejemplar.

Como he dicho, este proyecto tiene un hermano gemelo, realizado por mi compañero y amigo Ramón Aparicio, con el que compartí teoría, práctica, inconvenientes, pequeños logros y hasta el piso en Austria. El peso de la carga se reduce al compartirlo y agradezco a Ramón sus horas de conversación y consejos.

Por último, y consciente de que la consumación del Proyecto Fin de Carrera cierra un periodo fecundo de aprendizaje, quiero dar las gracias a todos los que han hecho posible la consecución de este título: familia, compañeros, amigos y profesores.

Gonzalo J. Moreno Muñoz
Cartagena, septiembre de 2006



Índice

AGRADECIMIENTOS	- 5 -
ÍNDICE	- 6 -
ÍNDICE DE FIGURAS	- 9 -
CAPÍTULO 1. REDES WDM	- 11 -
1.1 INTRODUCCIÓN.....	- 11 -
1.2 REDES ÓPTICAS.....	- 12 -
1.3 WAVELENGTH DIVISION MULTIPLEXING.....	- 13 -
1.4 EVOLUCIÓN DE LAS REDES WDM	- 17 -
1.4.1 Sistemas WDM punto a punto	- 17 -
1.4.2 <i>Wavelength Add/Drop Multiplexer (WADM)</i>	- 18 -
1.4.3 Fibras y matrices de conmutación de longitudes de onda, estrellas pasivas, <i>routers</i> pasivos y <i>switches</i> activos.....	- 19 -
1.5 ARQUITECTURAS DE REDES ÓPTICAS WDM.....	- 21 -
1.5.1 Redes ópticas locales WDM <i>Broadcast-And-Select</i>	- 21 -
1.5.2 Redes ópticas de área extendida <i>wavelength-routed</i>	- 22 -
1.6 PARADIGMA DE LAS REDES WDM.....	- 25 -
1.7 PROBLEMA RWA (ROUTING AND WAVELENGTH ASSIGNMENT).....	- 27 -
1.7.1 Tráfico estático o dinámico	- 27 -
1.7.2 Control centralizado o distribuido.....	- 28 -
1.7.3 Métodos de selección de ruta y longitud de onda.....	- 29 -
1.7.4 Diseño de Topologías Virtuales	- 30 -
1.8 OBJETIVOS Y DISTRIBUCIÓN DEL DOCUMENTO.....	- 31 -
CAPÍTULO 2. PROBLEMA RWA. DISEÑO DE TOPOLOGÍAS VIRTUALES	- 34 -
2.1 PROBLEMA RWA (ROUTING WAVELENGTH ASSIGNMENT)	- 34 -
2.1.1 Clasificación de los algoritmos RWA	- 34 -
2.1.2 Algoritmos RWA.....	- 35 -
2.2 DISEÑO DE TOPOLOGÍAS VIRTUALES	- 41 -
2.2.1 Introducción a las topologías virtuales	- 41 -
2.2.2 Arquitectura y notaciones.....	- 43 -
2.2.3 Limitaciones de la topología virtual	- 48 -
2.2.4 Optimización de la topología virtual	- 48 -
2.2.5 Formulación del Problema	- 50 -



2.3	ALGORÍTMOS HEURÍSTICOS	- 55 -
2.3.1	Subproblemas	- 56 -
2.3.2	Límites.....	- 57 -
2.3.3	Aproximaciones heurísticas y técnicas	- 60 -
2.4	ENFOQUES RELACIONADOS CON EL DISEÑO DE LA TOPOLOGÍA VIRTUAL.....	- 66 -
2.4.1	Beneficios incrementales	- 66 -
2.4.2	Límites de la conversión de longitud de onda.....	- 67 -
2.4.3	<i>Traffic Grooming</i>	- 69 -
2.4.4	<i>Lighpath</i> generalizados	- 69 -
2.4.5	Reconfiguración	- 70 -
CAPÍTULO 3. ALGORITMOS HEURÍSTICOS		- 72 -
3.1	DESARROLLO DE LOS ALGORÍTMOS.....	- 72 -
3.1.1	HLDA (<i>Heuristic Logical Design Algorithm</i>)	- 72 -
3.1.2	MLDA (<i>Minimum-Delay Logical Design Algorithm</i>)	- 73 -
3.1.3	SHLDA (<i>Shortest-Hop Heuristic Logical Design Algorithm</i>).....	- 74 -
3.1.4	LEMA (<i>Link Elimination via Matching Algorithm</i>)	- 75 -
3.1.5	TILDA (<i>Traffic Independent Logical Design Algorithm</i>).....	- 77 -
3.1.6	RLDA (<i>Random Logical Design Algorithm</i>).....	- 77 -
3.2	EJEMPLOS TEÓRICOS	- 78 -
3.2.1	HLDA.....	- 78 -
3.2.2	MLDA	- 79 -
3.2.3	LEMA	- 81 -
3.3	ALGORÍTMOS PROGRAMADOS	- 84 -
3.3.1	<i>HEURISTIC LOGICAL DESIGN ALGORITHM</i> (HLDA) - v1	- 85 -
3.3.2	<i>HEURISTIC LOGICAL DESIGN ALGORITHM</i> (HLDA) - v2	- 88 -
3.3.3	<i>MINIMUM-DELAY LOGICAL TOPOLOGY DESIGN ALGORITHM</i> (MLDA).....	- 90 -
3.3.4	<i>TRAFFIC INDEPENDENT LOGICAL DESIGN ALGORITHM</i> (TILDA).....	- 92 -
3.3.5	<i>RANDOM LOGICAL DESIGN ALGORITHM</i> (RLDA)	- 94 -
CAPÍTULO 4. DESCRIPCIÓN DE LA HERRAMIENTA.....		- 96 -
4.1	INTRODUCCIÓN A LA HERRAMIENTA	- 96 -
4.2	ESCENARIOS	- 96 -
4.2.1	Topologías físicas	- 97 -
4.2.2	Patrones de tráfico.....	- 99 -
4.3	FUNCIONES AUXILIARES.....	- 102 -
4.3.1	<i>PLOT TOPOLOGY</i>	- 102 -
4.3.2	<i>CONGESTION</i>	- 103 -
4.3.3	<i>STATUSBAR</i>	- 104 -
4.4	MANUAL DE LA HERRAMIENTA. MATPLANWDM	- 105 -
4.4.1	Programas designer.m y trainer.m.....	- 105 -
4.4.2	Aplicación en GUI de Matlab: MatPlanWDM	- 111 -
4.4.3	Modos de funcionamiento	- 112 -
4.4.4	Componentes de la interfaz gráfica	- 113 -
4.4.5	Resultados del MatPlanWDM. Informes y gráficas.....	- 119 -
CAPÍTULO 5. RESULTADOS		- 128 -
5.1	TOPOLOGÍAS VIRTUALES	- 128 -
5.1.1	Ejemplo 1. SIMPLENET 4 nodos. Patrón uniforme. HLDA y MLDA.....	- 128 -
5.1.2	Ejemplo 2. POELTENET 9 nodos. TILDA.....	- 130 -
5.1.3	Ejemplo 3. NSFNET 14 nodos. Patrón propio. RLDA.....	- 132 -
5.2	EVALUACIÓN DE ALGORITMOS.....	- 132 -
5.2.1	HLDA.....	- 132 -



5.2.2	MLDA	- 134 -
5.2.3	TILDA y RLDA.....	- 136 -
5.3	COMPARATIVA DE HEURÍSTICOS	- 137 -
5.4	HEURÍSTICOS VS. PROGRAMACIÓN LINEAL. EJEMPLO	- 140 -
5.4.1	Ejemplo AustriaNET de 7 nodos	- 140 -
CAPÍTULO 6. CONCLUSIONES		- 146 -
BIBLIOGRAFÍA.....		- 151 -



Índice de figuras

FIGURA 1. IMPLEMENTACIÓN TDM	- 12 -
FIGURA 2. IMPLEMENTACIÓN CDM	- 12 -
FIGURA 3. VENTANA ÓPTICA DE BAJA ATENUACIÓN.....	- 14 -
FIGURA 4. VARIOS CANALES WDM SE PROPAGAN POR UNA ÚNICA FIBRA.....	- 14 -
FIGURA 5. DIAGRAMA DE UN SISTEMA SIMPLE WDM	- 15 -
FIGURA 6. SISTEMA DE TRANSMISIÓN WDM PUNTO A PUNTO CON REGENERACIÓN	- 16 -
FIGURA 7. SISTEMA DE TRANSMISIÓN WDM CON AMPLIFICADORES Y CUATRO CANALES PUNTO A PUNTO	- 18 -
FIGURA 8. WAVELENGTH ADD/DROP MULTIPLEXER (WADM).....	- 19 -
FIGURA 9. ESTRELLA PASIVA 4 × 4.....	- 20 -
FIGURA 10. ROUTER PASIVO 4 × 4 (CUATRO LONGITUDES DE ONDA)	- 21 -
FIGURA 11. SWITCH PASIVO 4 × 4 (CUATRO LONGITUDES DE ONDA).....	- 21 -
FIGURA 12. ESTRELLA PASIVA BASADA EN UNA RED WDM	- 22 -
FIGURA 13. RED WDM WAVELENGTH-ROUTED DE ÁREA EXTENDIDA.....	- 23 -
FIGURA 14. POSIBLES CAPAS EN UNA RED DE TRANSPORTE WDM.....	- 25 -
FIGURA 15. RED DE <i>LIGHTPATHS</i> SOBRE UNA RED WDM (LOS CÍRCULOS NEGROS INDICAN EXTREMOS DE UN <i>LIGHTPATH</i>).....	- 26 -
FIGURA 16. EFECTO DE LA RESTRICCIÓN DE CONTINUIDAD DE LONGITUD DE ONDA EN EL BLOQUEO.....	- 35 -
FIGURA 17. ILUSTRACIÓN DE UNA TOPOLOGÍA FÍSICA Y VIRTUAL. (A) TOPOLOGÍA FÍSICA. (B) UNA POSIBLE TOPOLOGÍA VIRTUAL.....	- 41 -
FIGURA 18. UNA RED WDM. LOS NODOS DE ENRUTAMIENTOS ESTÁN INTERCONECTADOS POR FIBRAS PUNTO A PUNTO. LAS LÍNEAS PUNTEADAS INDICAN LOS <i>LIGHTPATHS</i>	- 42 -
FIGURA 19. EVOLUCIÓN DE LA MATRIZ DE TRÁFICO EN EL HLDA.....	- 78 -
FIGURA 20. TABLA DE ITERACIONES. ASIGNACIÓN DE <i>LIGHTPATHS</i> PARA EL EJEMPLO DEL HLDA	- 79 -
FIGURA 21. EJEMPLO TEÓRICO HLDA. (A) TOPOLOGÍA FÍSICA. (B) TOPOLOGÍA VIRTUAL.....	- 79 -
FIGURA 22. EVOLUCIÓN DE LA MATRIZ DE TRÁFICO EN EL MLDA	- 80 -
FIGURA 23. TABLA DE ITERACIONES. ASIGNACIÓN DE <i>LIGHTPATHS</i> PARA EL EJEMPLO DEL MLDA.....	- 80 -
FIGURA 24. EJEMPLO TEÓRICO MLDA. (A) TOPOLOGÍA FÍSICA. (B) TOPOLOGÍA VIRTUAL	- 81 -
FIGURA 25. EJEMPLO TEÓRICO LEMA. (A) TOPOLOGÍA FÍSICA. (B) MATRIZ DE TRÁFICO	- 81 -
FIGURA 26. TOPOLOGÍA FÍSICA DE 4 NODOS. SIMPLINET	- 98 -
FIGURA 27. TOPOLOGÍA FÍSICA DE NUEVE NODOS. POELTENET	- 98 -
FIGURA 28. TOPOLOGÍA FÍSICA DE 14 NODOS. NSFNET	- 99 -
FIGURA 29. (A) TOPOLOGÍA FÍSICA DE LA RED AUSTRIANET. (B) TOPOLOGÍA VIRTUAL	- 103 -
FIGURA 30. TOPOLOGÍA VIRTUAL OBTENIDA CON EL PROGRAMA DESIGNER.M	- 107 -
FIGURA 31. CURVAS DE CONGESTIÓN OBTENIDAS CON EL PROGRAMA TRAINER.M.....	- 109 -
FIGURA 32. CURVAS DE CONGESTIÓN PARA LOS ALGORITMOS HEURÍSTICOS OBTENIDAS CON EL PROGRAMA TRAINER.M	- 111 -
FIGURA 33. MODOS DE FUNCIONAMIENTO DEL MATPLANWDM	- 113 -
FIGURA 34. INTERFAZ GRÁFICA DEL MATPLANWDM	- 113 -
FIGURA 35. CAMPOS DEL CUADRANTE <i>SIMULATION DATA</i>	- 114 -
FIGURA 36. MENÚS DESPLEGABLES EN EL CUADRANTE <i>SIMULATION DATA</i> . (A) TOPOLOGÍA FÍSICA.....	- 116 -
FIGURA 37. SEGUNDO CUADRANTE DE LA INTERFAZ GRÁFICA. TOPOLOGÍA FÍSICA.	- 116 -
FIGURA 38. RESULTADOS PRESENTADOS EN EL SEGUNDO Y CUARTO CUADRANTE DE LA INTERFAZ DEL MATPLANWDM. (A) RESULTADO DEL DISEÑO DE UNA TOPOLOGÍA VIRTUAL. (B) RESULTADOS DEL LA COMPARACIÓN DE LOS ALGORITMOS.....	- 117 -
FIGURA 39. MENUS DESPLEGABLES. (A) <i>CLOSE</i> (B) <i>MODE</i> (C) <i>HELP</i>	- 118 -
FIGURA 40. (A) VENTANA FLOTANTE DE CRÉDITOS DEL MATPLANWDM. (B) FICHERO WEB DE AYUDA DEL HLDA	- 119 -
FIGURA 41. RED DE <i>LIGHTPATHS</i> . RESULTADO GRÁFICO DEL PRIMER MODO DE FUNCIONAMIENTO.....	- 120 -
FIGURA 42. INTERFAZ GRÁFICA AL DISEÑAR UNA TOPOLOGÍA VIRTUAL	- 122 -
FIGURA 43. CURVAS DE CONGESTIÓN SEGÚN LONGITUDES DE ONDA.....	- 123 -



FIGURA 44. INTERFAZ GRÁFICA AL EVALUAR UN ALGORITMO HEURÍSTICO.....	- 124 -
FIGURA 45. REPRESENTACIÓN GRÁFICA DE LAS CURVAS DE CONGESTIÓN	- 125 -
FIGURA 46. INTERFAZ GRÁFICA AL COMPARAR LOS ALGORITMOS HEURÍSTICOS	- 127 -
FIGURA 47. TOPOLOGÍAS FÍSICAS Y LÓGICAS DEL EJEMPLO 1. SIMPLINET DE 4 NODOS. PATRÓN UNIFORME. HLDA Y MLDA..	- 129 -
FIGURA 48. TOPOLOGÍA VIRTUAL EJEMPLO2. POELTENET DE 9 NODOS. TILDA	- 131 -
FIGURA 49. TOPOLOGÍA VIRTUAL DEL EJEMPLO 3. NSFNET DE 14 NODOS. RLDA.....	- 132 -
FIGURA 50. CURVAS DE CONGESTIÓN EN LA CARACTERIZACIÓN DEL HLDA. ESCENARIO 1.....	- 133 -
FIGURA 51. CURVAS DE CONGESTIÓN EN LA CARACTERIZACIÓN DEL HLDA. ESCENARIO 2.....	- 134 -
FIGURA 52. CURVAS DE CONGESTIÓN PARA LA CARACTERIZACIÓN DEL MLDA. (A) PATRÓN UNIFORME U(0,100) (B) PATRÓN NO- UNIFORME 50 % (C) PATRÓN NO-UNIFORME 25 %.....	- 136 -
FIGURA 53. CURVAS DE CONGESTIÓN PARA LA CARACTERIZACIÓN DEL TILDA (A) Y RLDA (B)	- 137 -
FIGURA 54. CURVAS DE CONGESTIÓN PARA ALGORITMOS. (A) POELTENET 9 NODOS. PATRÓN UNIFORME. (B) NSFNET 14 NODOS. PATRÓN NO UNIFORME.	- 139 -
FIGURA 55. TOPOLOGÍA FÍSICA DE LA RED AUSTRIANET DE 7 NODOS	- 141 -
FIGURA 56. TOPOLOGÍA VIRTUAL OBTENIDA CON EL SOLVER PARA AUSTRIANET	- 142 -
FIGURA 57. TOPOLOGÍA VIRTUAL OBTENIDA CON EL HLDA PARA AUSTRIANET	- 143 -
FIGURA 58. TOPOLOGÍA VIRTUAL OBTENIDA CON EL TILDA PARA AUSTRIANET	- 143 -
FIGURA 59. TOPOLOGÍA VIRTUAL OBTENIDA CON EL RLDA PARA AUSTRIANET	- 144 -



CAPÍTULO 1

Redes WDM

1.1 INTRODUCCIÓN

Una de las exigencias de la Sociedad de la Información es el acceso a las fuentes de datos de una forma más veloz, manteniendo los requerimientos de lugar, de tiempo y de formato. En nuestros días la información se nos ofrece a través de redes globales de comunicación que empiezan a no satisfacer las demandas de ancho de banda en sus implementaciones actuales.

Las tecnologías de fibra óptica tienen una capacidad potencialmente ilimitada en comparación con otras tecnologías existentes: gran ancho de banda (cercano a los 50 terabits por segundo (Tbps)), baja atenuación de la señal (0.2 dB/km), baja distorsión, bajo requerimiento de energía, requiere poco espacio y material, y presenta un bajo coste.

El ancho de banda potencial de una fibra monomodo está cercano a los 50 Tbps, valor superior en casi cuatro órdenes de magnitud a las actuales tasas de tecnología electrónica que se sitúan entorno a unos pocos gigabits por segundo (Gbps). Parece claro que todos los esfuerzos se deben de orientar a explotar esta diferencia abismal entre el ancho de banda ofrecido por el dominio eléctrico y el óptico. Conviene notar que el techo de velocidad se situará en los usuarios finales, que pueden ser estaciones de trabajo o puertas de enlace con interfaces a subredes de baja velocidad. Estos usuarios acceden a la red limitados por la velocidad electrónica (Gbps), por lo que la clave del diseño de las redes de comunicaciones ópticas para explotar adecuadamente su ancho de banda estará en introducir transmisiones de múltiples usuarios en la arquitectura de red y los protocolos. En una red de comunicaciones ópticas la concurrencia puede implementarse mediante multiplexación de longitud de onda-frecuencia (WDM), ranuras temporales (TDM) o forma de onda [espectro ensanchado, multiplexación por división de código (CDM)], o una combinación de las mismas.

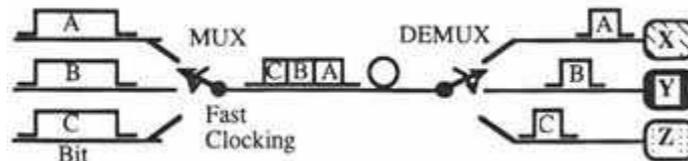


Figura 1. Implementación TDM

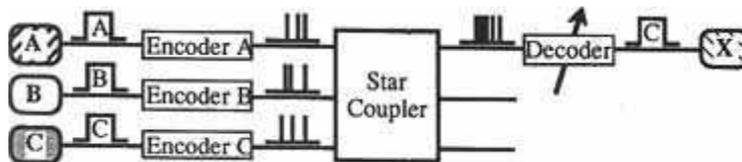


Figura 2. Implementación CDM

Mediante una tecnología TDM, cada usuario final sería capaz de sincronizarse dentro de una ranura (“slot”) temporal. La tasa de bit de una multiplexación TDM es la tasa agregada sobre todos los canales TDM del sistema, mientras que la tasa de chip CDM puede ser mucho más alta que la suma de cada usuario. Como resultado, las tasas de ambas tecnologías serán mucho más altas que la velocidad de procesamiento electrónico, lo que significa que alguna de las partes de la red deben operar a una velocidad más alta que la velocidad eléctrica. Además TDM y CDM son relativamente menos atractivas que WDM, como consecuencia de no presentar esta última esa restricción.

Como consecuencia de lo anterior es WDM la tecnología óptica de multiplexación actualmente mejor posicionada en las redes de comunicaciones ópticas debido a que todos los equipos finales requieren trabajar sólo a la tasa de bit de un canal WDM, que puede ser elegida arbitrariamente, e.g., velocidad de pico en el procesado electrónico. Por consiguiente, los esfuerzos más significantes se están haciendo en el desarrollo y aplicación de WDM y en su comercialización.

1.2 REDES ÓPTICAS

Las redes ópticas, donde los datos se convierten en unidades de luz llamadas fotones y son transmitidos a través de fibras, son sensiblemente más veloces que las redes tradicionales, donde el formato de los datos son electrones que viajan a través de cables de cobre, debido a que el peso de los



fotones es menor que el de los electrones y a diferencia de los electrones, los fotones no se interceptan durante el movimiento a través de la fibra por la ausencia de carga eléctrica y no ser vulnerables a las interferencias externas a la fibra. La luz se propaga a frecuencias más altas y como consecuencia a longitudes de onda más bajas. Más bits de información “cabén” en la misma longitud de fibra que de cobre.

Las fibras de vidrio ópticas se basan en el *principio de la reflexión total*, que fue descubierto en torno a 1850 y fue desarrollado con los primeros endoscopios a principios del siglo XX. La utilización de fibras de vidrio para comunicarse fue propuesta por primera vez por Kao y Hockham en 1966. La manufactura de fibras ópticas comenzó en la década de los 70. A finales de los 80 y principios de los 90 aparecieron una gran variedad de redes ópticas que utilizaban fibras ópticas en sustitución de cables de cobre para alcanzar velocidades más altas. Diversas interconexiones de computadores como ESCON (*Enterprise Serial Connection*), *Fiber Channel* y HiPPI (*High Performance Parallel Interface*) para la interconectar computadores entre sí y con otros sistemas periféricos, utilizan componentes ópticos de baja tasa de bit a bajo coste. FDDI (*Fiber Distributed Data Interface*) utiliza ambos, *token rings* de fibra óptica para proporcionar velocidades de 100-200 megabits por segundo (Mb/s) en las transmisiones entre estaciones de trabajo. SONET/SDH (*Synchronous Optical NETWORK* en América del Norte, *Synchronous Digital Hierarchy* en Europa y Asia)¹ –que conforma la base de las actuales redes troncales de alta velocidad y es uno de los estándar con más implantación en la industria de redes– permite tasas de transmisión similares a OC-192², entorno a 10 gigabits por segundo (Gb/s).

1.3 WAVELENGTH DIVISION MULTIPLEXING

El ancho de banda teórico de la fibra óptica se sitúa entorno a los 25 terahercios (THz) en la banda de baja atenuación de 1,55 micras como se muestra en la Figura 1.1. A pesar de ello las velocidades alcanzadas (tasa de datos) son de unos pocos gigabits por segundo (Gb/s) por la limitación impuesta en la tecnología electrónica por los usuarios finales. Este “cuello de botella” electrónico hace muy difícil aprovechar el ancho de banda potencial que ofrece una fibra mediante un solo canal (longitud de

¹ Conjunto de estándares para la transmisión digital de información sobre redes ópticas

² OC-*n* [*Optical Carrier-n*] especifica una tasa electrónica de $n \times 51.84$ Mb/s aproximadamente; por lo tanto OC-48 y OC-192 se corresponde con velocidades de 2.5 Gb/s y 10 Gb/s respectivamente. OC-768 [40 Gb/s] es el próximo objetivo para el techo de las velocidades electrónicas,



onda). Los avances recientes que han permitido alcanzar velocidades de terabits por segundo (Tb/s) son el resultado de dos desarrollos diferentes: por un lado la tecnología *wavelength division multiplexing* (WDM), que envía simultáneamente haces de luz a diferentes longitudes de onda a través del núcleo de una fibra óptica; y por otro, *erbium-doped fiber amplifier* (EDFA), que amplifica simultáneamente un conjunto de señales a diferentes longitudes de onda, sin tener en cuenta su velocidad o tipo de modulación.

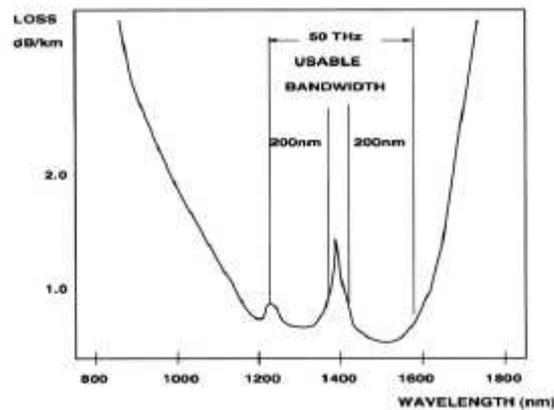


Figura 3. Ventana óptica de baja atenuación

Conceptualmente se puede trazar un paralelismo entre WDM y FDM (*frequency division multiplexing*), ya que varias señales de información modulan una señal óptica a diferentes longitudes de onda y las resultantes se combinan y transmiten a través de la misma fibra óptica.

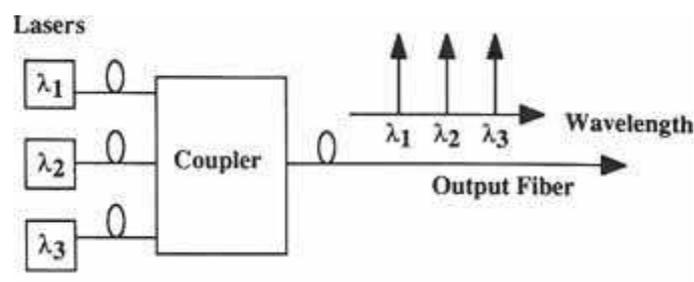


Figura 4. Varios canales WDM se propagan por una única fibra

Los prismas y las rejillas de difracción conocidos se utilizan para combinar (multiplexar) o separar (demultiplexar) las longitudes de onda en distintas señales. A diferencia de las usadas en FDM, las rejillas de difracción en un sistema óptico WDM son totalmente pasivas y mucho más fiables. Además las ondas portadoras de cada canal WDM tienen una frecuencia un millón de veces mayor que uno FDM (THz vs MHz). Dentro de cada canal WDM es posible introducir un canal FDM mediante la subdivisión del ancho de banda en diferentes canales de radio, cada uno a su frecuencia



correspondiente. Esta técnica se denomina *subcarrier multiplexing*. Una única longitud de onda también puede ser dividida entre varios nodos de la red mediante *time division multiplexing* (TDM). Se puede comprobar como WDM elimina el “cuello de botella” electrónico dividiendo el espectro de transmisión (banda de 1.55 micras) en un conjunto de canales de forma que las longitudes de onda no se solapen y coexistan en una misma fibra; cada longitud de onda soporta un canal de comunicación que opera a la velocidad de pico electrónica. Gracias a WDM se puede disponer de un incremento sustancial de ancho de banda sin la necesidad de grandes inversiones en el despliegue de nuevas fibras.

La revolución de WDM se puede apreciar con ejemplo sencillo: para transmitir 40 Gb/s sobre 600 Km en un sistema tradicional se requerían 16 pares de fibra y 224 regeneradores ($[600/40-1]16$), necesitando uno cada 40 Km. Con un sistema de transmisión WDM se puede usar un único par de fibra óptica con 16 canales y 4 amplificadores ($600/120 - 1$), colocados cada 120 Km.

Sólo con introducir dos, tres o cuatro longitudes de onda adicionales en los sistemas de transmisión punto a punto ya instalados, WDM ha mejorado de forma notable la capacidad. Ya están disponibles sistemas WDM de 16 longitudes de onda en *OC-48* y 32 longitudes de onda en *OC-192*, que proporcionan tasas agregadas de 40 y 320 Gb/s respectivamente. La técnica *dense wavelength division multiplexing* (DWDM) permite incrementar de forma efectiva el número de canales en la fibra mediante la explotación intensiva de la banda espectral, es decir disminuyendo la distancia entre canales (aumento de la densidad). El espaciado entre canales tradicional oscila de 0.4 nanómetros (nm) a 4 nm, equivalente en frecuencia a un rango 50 – 500 GHz.

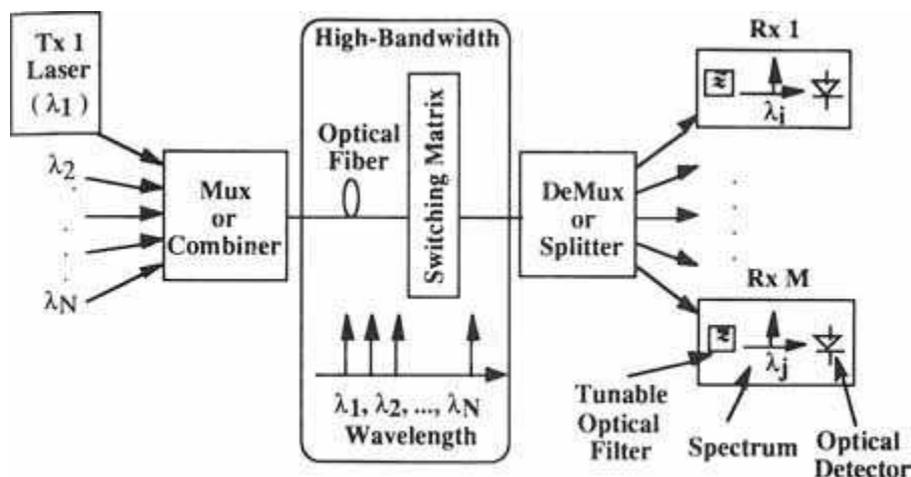


Figura 5. Diagrama de un sistema simple WDM

Como es sabido por la teoría óptica, cuando los problemas de atenuación son resueltos, los efectos de la dispersión se hacen importantes. De forma similar, cuando se controlan los problemas asociados a la dispersión, los efectos no lineales³ se hacen dominantes, éstos limitan potencialmente la tasa de transmisión máxima de los sistemas de comunicación WDM. Las causas de los efectos no lineales son la interacción entre la señal óptica y las moléculas de sílice de la fibra o entre aquella y las ondas acústicas, también es importante la dependencia del índice de refracción con la intensidad óptica de las señales que atraviesan la fibra. Estas no linealidades se pueden controlar en las fibras con la elección cuidadosa de la potencia y el espaciado de los canales.

La llegada de EDFA permite la comercialización de los sistemas WDM, encontrando una vía para amplificar todas las longitudes de onda al mismo tiempo, independientemente de sus tasas de transmisión individuales, tipos de modulación o niveles de potencia. Antes de la aparición de los EDFAs, los efectos de las pérdidas ópticas eran compensados cada 10 Km por regeneradores electrónicos, que requerían una conversión opto-electrónica y viceversa. La limitación fundamental proviene de que los regeneradores electrónicos trabajan sólo para la tasa especificada a una longitud de onda, sin considerar retardos y consumo. El amplificador EDFA contiene varios metros de fibra de vidrio de sílice que ha sido dopada con iones de erbio, metal lantánido ubicado en las “tierras raras”. A través de un láser de bombeo óptico se activan los iones de erbio, que amplifican las señales ópticas que atraviesan la fibra. Es una providencial coincidencia que la banda de amplificación del EDFA (1,53 – 1,54 micras) sea la misma que la banda de transmisión de las fibras ópticas (1,55 micras).

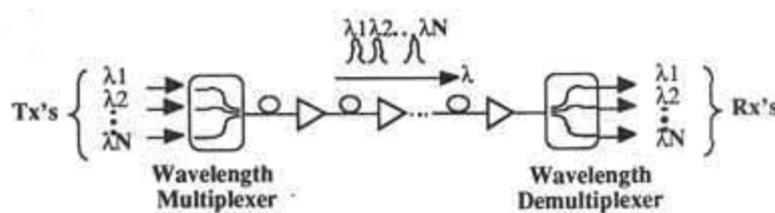


Figura 6. Sistema de transmisión WDM punto a punto con regeneración

En la práctica existen tres tipos de regeneradores de señal: los regeneradores de 3R, que excluye las operaciones de *reshaping* y *reclocking*⁴; los regeneradores de 2R, que sólo ejecuta la operación de

³ Dispersión estimulada Raman, dispersión estimulada Brillouin, modulación self- y cross-phase y el mezclado de cuatro ondas.

⁴ *Reshaping* de la señal reproduce el pulso original de cada bit, eliminando la mayoría del ruido. *Reclocking* de la señal sincroniza la señal con el patrón de tiempo original y la tasa de bit.



reshaping y los regeneradores de 1R, que sin operaciones de *reshaping* y *reclocking*, llevan a cabo una amplificación simple usando EDFAs u otros amplificadores.

Se llama transponedor a un elemento de red que combina el receptor óptico, varios niveles de regeneración y un transmisor óptico. Si un transponedor es capaz de transmitir a una longitud de onda diferente a la de recepción, se lleva a cabo una conversión de longitud de onda como resultado de la regeneración.

1.4 EVOLUCIÓN DE LAS REDES WDM

1.4.1 Sistemas WDM punto a punto

La tecnología WDM está siendo desplegada por varias empresas de telecomunicación en forma de comunicaciones punto a punto. Este despliegue está siendo impulsado por la incremento en la demanda de ancho de banda. Cuando los niveles de demanda superen las capacidades ofrecidas por las fibras existentes, WDM se convertirá en la alternativa más rentable. Así, un estudio comparó los costes relativos de actualizar los enlaces de transmisión punto a punto de OC-48 a OC-192 con tres alternativas diferentes.

1. Solución “multi-fibra”. Instalación de fibras y equipamientos adicionales.
2. Solución WDM de “cuatro-canales” (ver Figura 1.7). Se trata de un multiplexador WDM que combina cuatro fuentes de datos, cada una en una única longitud de onda y los envía a través de la fibra. En recepción un demultiplexador separa los cuatro canales a sus flujos originales.
3. Solución OC-192 “alta velocidad electrónica”

Diversos análisis demuestran que para distancias menores de 50 Km, la solución “multifibra” es la menos cara, pero para distancias del enlace de transmisión superiores a 50 Km, el coste de la solución WDM es el menor, seguido de cerca por la solución “alta velocidad electrónica”.



Los mux/demux en los enlaces punto a punto los comercializan diversos vendedores IBM, Pirelli y AT&T. En estos productos ya comercializados, el número máximo de canales disponibles por fibra es de 64, aunque se espera incrementarlos en breve.

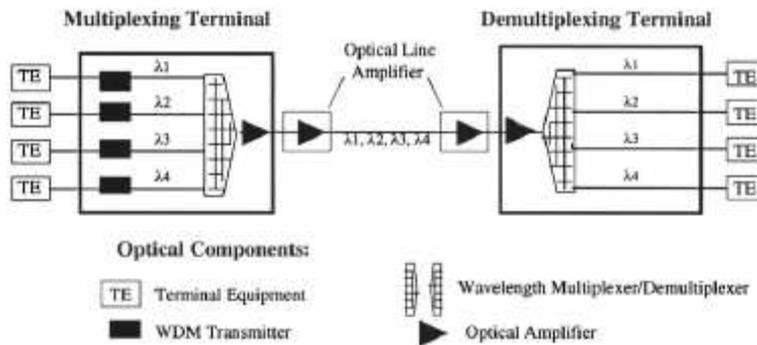


Figura 7. Sistema de transmisión WDM con amplificadores y cuatro canales punto a punto

1.4.2 Wavelength Add/Drop Multiplexer (WADM)

En la Figura 1.8 se muestra la estructura de un *Wavelength Add/Drop Multiplexer (WADM)*. Consiste en un demultiplexador, seguido de una etapa de *switches* 2×2 , habiendo un *switch* por longitud de onda y terminando en un multiplexador. El WADM se puede insertar en un enlace físico de fibra. Si todos los *switches* 2×2 están en estado no activo, todas las longitudes de onda atraviesan el WADM sin alteración. En cambio, en el caso de estar alguno de los *switches* en estado activo (como la longitud de onda λ_i en la Figura 1.8), la señal de la correspondiente longitud de onda es “extraída” de forma local y un flujo de datos entrante se puede “añadir” con la misma longitud de onda en el nodo donde se ubica el WADM. En el caso de que la interfaz del WADM tenga el hardware necesario y la capacidad de procesamiento se pueden “añadir y extraer” más de una longitud de onda.

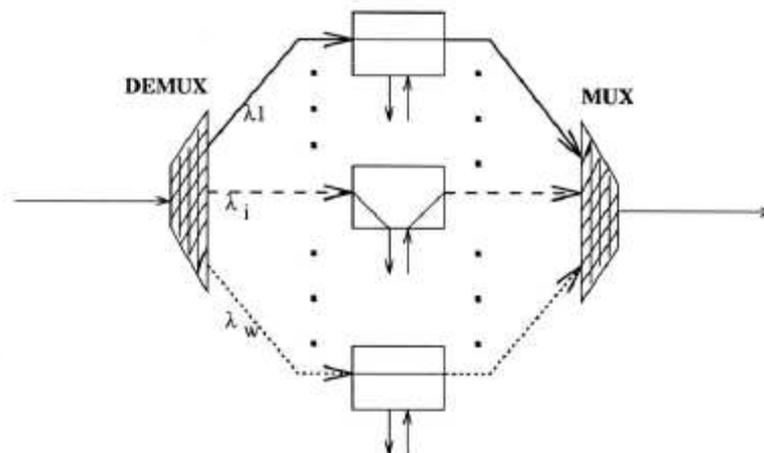


Figura 8. Un Wavelength Add/Drop Multiplexer (WADM)

1.4.3 Fibras y matrices de conmutación de longitudes de onda, estrellas pasivas, *routers* pasivos y *switches* activos

Para manejar una red de enlaces de fibra con distintas longitudes de onda, se requieren dispositivos de interconexión adecuados, que se agrupan en tres categorías:

- Estrella pasiva (ver Figura 1.9)
- *Router* pasivo (ver Figura 1.10)
- *Switch* activo (ver Figura 1.11)

La “estrella pasiva” recibe una señal en uno de sus puertos de entrada y a una longitud de onda determinada, divide la potencia de la señal entre todos los puertos salida y a la misma longitud de onda que la recibió la envía por todos ellos. En la Figura 1.9 las señales con longitudes de onda λ_1 y λ_4 de los puertos *Input Fiber 1* y *Input Fiber 4* respectivamente, son enviadas a todos los puertos de salida en una operación de *broadcast*. Es posible dar lugar a una colisión en el caso de que dos o más señales de entrada salgan de la estrella a la misma longitud de onda. Si se asume que hay tantas longitudes de onda como puertos, una estrella pasiva $N \times N$ puede enrutar N conexiones simultáneas.

Un “*router* pasivo” puede enrutar de forma separada cada una de las longitudes de onda entrantes en distintos puertos de salida a la misma longitud de onda, e.g., las longitudes de onda λ_1 , λ_2 , λ_3 y λ_4 entrantes en el *Input Fiber 2* se enrutan de forma alternada en los cuatro puertos de salida (ver Figura 1.10). El “*router* pasivo” permite la reutilización de longitudes de onda, es decir, la misma longitud de



onda se puede usar en diversas conexiones a través de *router*. La “matriz de enrutamiento” caracteriza al *router* y determina que longitud de onda entrante se enruta al correspondiente puerto de salida, esta matriz está conformada por las conexiones internas entre el demultiplexador y multiplexador. La “matriz de enrutamiento” es estática y no admite cambios. Esta familia de *routers* están disponibles en el mercado y son conocidos como *Latin routers*, *waveguide grating routers* (WGRs), *wavelength routers* (WRs), etc. Si se asume que hay tantas longitudes de onda como puertos, un “*router* pasivo” puede enrutar N^2 conexiones simultáneas (en comparación con las N que podía enrutar la “estrella pasiva”), si bien carece de la capacidad *broadcast* de aquélla.

De forma similar el “*switch* activo” permite la reutilización de longitudes de onda y puede enrutar N^2 conexiones simultáneas, pero tiene una ventaja importante sobre el “*router* pasivo”, su “matriz de enrutamiento” puede ser reconfigurada mediante control electrónico según la demanda, presentando como inconveniente su necesidad de ser alimentado por una fuente externa. El “*switch* activo” se denomina de forma mayoritaria *wavelength-routing switch* (WRS) aunque también es referido como *wavelength selective crossconnect* (WSXC).

Se puede añadir una funcionalidad adicional al “*switch* activo”, la posibilidad de convertir la longitud de onda antes de entrar en la etapa de multiplexado, de forma que se la señal se transmite a la salida con una longitud de onda diferente a la entrante. Los “*switches* activos” que presentan esta propiedad se denominan *wavelength-convertible switch* o *wavelength interchanging crossconnect* (WIXC).

Las aplicaciones de los tres dispositivos vistos son diferentes: la “estrella pasiva” se usa en entornos locales WDM, el “*router* pasivo” tiene distintas aplicaciones como dispositivo mux/demux y el “*switch* activo” se emplea en redes ópticas de área extensa.

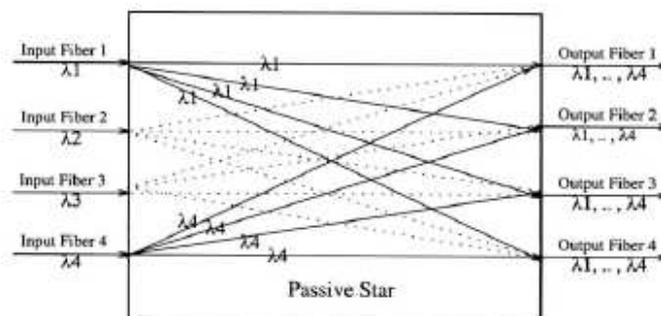


Figura 9. Estrella pasiva 4 × 4

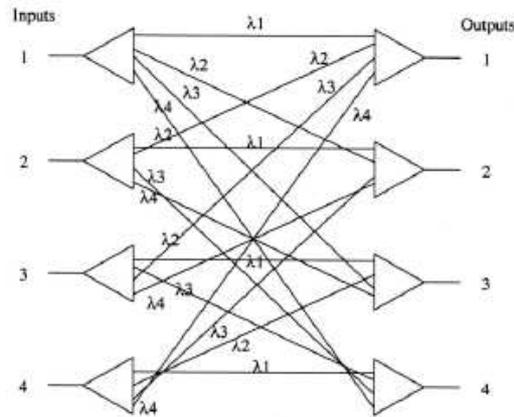


Figura 10. Router pasivo 4 × 4 (cuatro longitudes de onda)

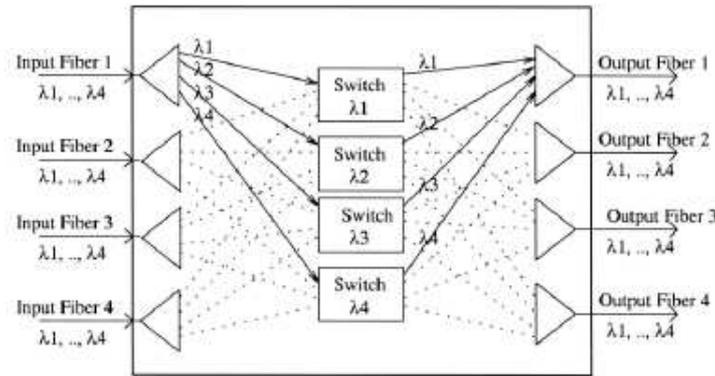


Figura 11. Switch pasivo 4 × 4 (cuatro longitudes de onda)

1.5 ARQUITECTURAS DE REDES ÓPTICAS WDM

1.5.1 Redes ópticas locales WDM *Broadcast-And-Select*

Una red óptica local WDM se puede construir mediante fibras bidireccionales en forma de estrella pasiva, como se muestra en la Fig. 1.12. Un nodo envía su transmisión a la estrella en una longitud de onda disponible, usando un láser que produce la señal óptica. Las señales de información provenientes de distintas fuentes se combinan en la estrella y la potencia de cada una es dividida mediante ecualización y reenviada a todos los nodos destino, mediante sus respectivos receptores ópticos. Los receptores de los nodos se sintonizan a una determinada longitud de onda utilizando filtros ópticos. Las comunicaciones entre las fuentes y los nodos pueden seguir dos métodos: (1) mono-salto (2) multi-

salto. Cuando una fuente transmite a una longitud de onda determinada λ_1 , más de un receptor se puede sintonizar a esa misma longitud onda y por lo tanto, todos serían receptores de la misma señal óptica. De esta forma la arquitectura de estrella-pasiva soportaría servicios *multicast*.

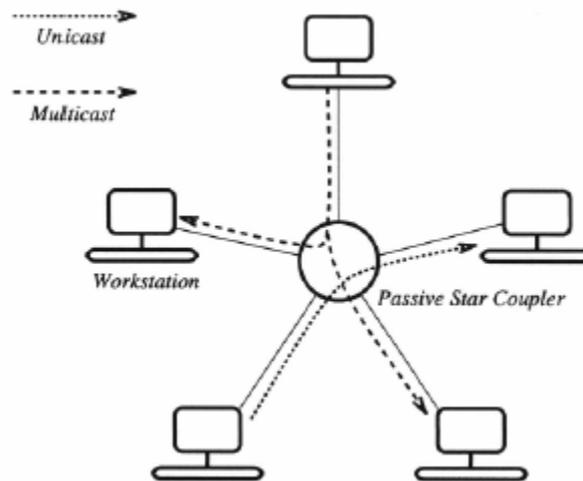


Figura 12. Estrella pasiva basada en una red WDM

1.5.2 Redes ópticas de área extendida *wavelength-routed*

Las redes WDM *wavelength-routed* presentan la peculiaridad de evitar los tres problemas principales de las redes *broadcast* vistas en el apartado anterior: reutilización de longitudes de onda, pérdidas de potencia por división de señal y escalabilidad a redes de área extensas (WANs). En esencia una red *wavelength-routed* consiste en un conjunto de nodos de enrutamiento (WXC) interconectados por enlaces de fibra punto a punto dentro de una topología de red arbitraria. Cada nodo final está conectado a un WXC por un enlace de fibra. El conjunto de nodos finales y su correspondiente WXC se referencia como nodo de la red. Cada nodo está equipado con un conjunto de transmisores y receptores, susceptibles de ser sintonizados a la longitud de onda deseada, para enviar y recibir datos a/de la red.

En una red *wavelength-routed* los mensajes se envían de un nodo a otro usando una longitud de onda continua denominada camino de luz o *lightpath*, que no requiere ningún mecanismo de reconversión opto-electrónica ni *buffers* en los nodos intermedios. Este proceso se llama *enrutamiento de longitud de onda* y en él, los nodos intermedios encaminan los *lightpaths* en el dominio óptico mediante sus respectivos WXC. Los nodos finales de los *lightpath* acceden usando transmisores y receptores sintonizados a la longitud de onda que opera.



Un *lightpath* se puede definir como una vía de comunicación en el dominio óptico entre dos nodos, establecida con la misma longitud a través de la ruta que sigue la señal transmitida. Por lo tanto constituye una vía de gran ancho de banda, capaz de transmitir datos a varios gigabits por segundo y está caracterizada de forma unívoca por una ruta física y una longitud de onda. La restricción de que sea la misma longitud de onda utilizada a lo largo de toda la ruta física se conoce como *restricción de continuidad de longitud de onda*. Conviene destacar sin embargo que dos *lightpaths* pueden usar la misma longitud de onda si utilizan un conjunto de enlaces disjuntos. Esta propiedad se conoce como *reutilización de longitud de onda*.

En comparación con otras arquitecturas de redes ópticas, las redes *wavelength-routed* presentan ventajas considerables. La propiedad de reutilizar las longitudes de onda en conjuntos disjuntos de enlaces asegura una mayor escalabilidad de la red, respecto a las redes *broadcast-and-select* vistas en el apartado anterior. Una característica importante, que permite a las redes *wavelength-routed* extenderse hasta grandes distancias, es que la potencia de los *lightpaths* no se divide en destinos irrelevantes.

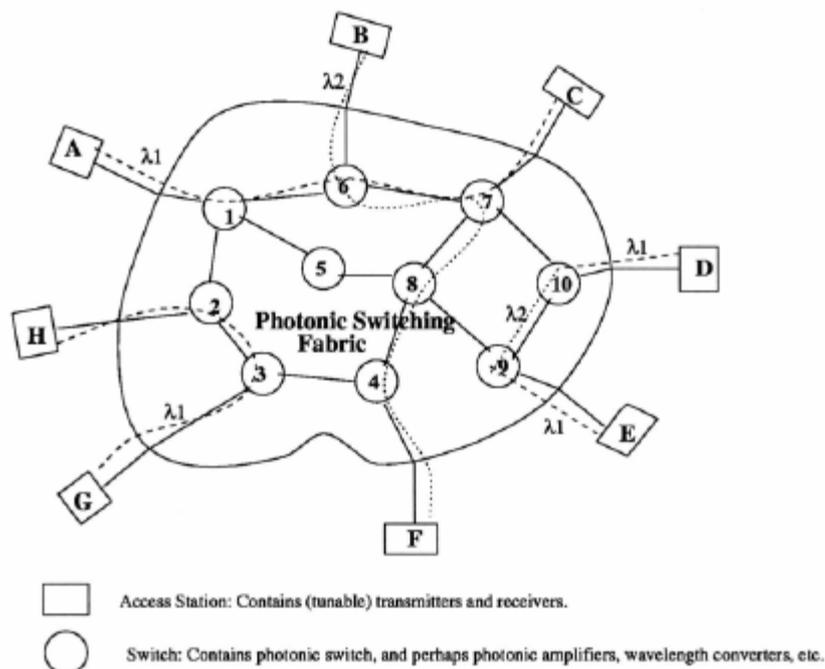


Figura 13. Red WDM Wavelength-routed de área extendida

En este escenario, dada una red WDM, el problema fundamental se centra en el enrutamiento y la asignación de longitudes de onda a los *lightpaths*. Existen diversos métodos y algoritmos que abordan la resolución de ambos problemas y a la vez minimizan el número de longitudes de onda utilizadas. El número de longitudes de onda que admite una fibra óptica está entre 4 y 32, aunque las líneas actuales



de investigación aseguran hasta 100 longitudes de onda por fibra, y esta variable juega un papel fundamental en la resolución de ambos problemas en función de los recursos disponibles.

La conmutación de paquetes sobre redes *wavelength-routed* se puede modelar mediante sistemas mono-salto o multi-salto, de forma similar a las redes *broadcast-and-select*. En la aproximación multi-salto, una topología virtual⁵ se sobrepone sobre la topología física mediante los WXC's en los nodos. Sobre esta topología virtual los paquetes se enrutan a través de los nodos intermedios antes de alcanzar el destino final. En cada uno de los nodos intermedios el paquete se convierte al dominio electrónico para su procesado, siguiendo la operación inversa y su retransmisión en otra longitud de onda.

Una red de transporte WDM se puede descomponer en tres capas, a saber: capa de medios físicos, capa óptica y capa del cliente, como se muestra en el esquema de la Fig. 1.14. El desarrollo de las tecnologías WDM ha introducido la capa óptica entre la inferior netamente física y la superior de acceso a los usuarios finales. Como se ha comentado anteriormente un conjunto de *lightpaths* forma la capa óptica (topología virtual); proveyendo a los clientes de la capa superior transparencia de protocolos y conmutación de circuitos. Esto es posible por la flexibilidad que presentan los *lightpaths* de transportar los paquetes a distintas tasas de bit y sobre distintos protocolos y, consecuentemente, a distintos clientes de forma concurrente. Por ejemplo, dentro de una misma red algunos *lightpath* pueden transportar flujos de SONET, mientras que otros lo harán con paquetes IP o celdas ATM. Una red con una capa óptica puede ser configurada para reenrutar los *lightpaths* por vías alternativas de forma automática, en caso de haber un fallo en el sistema, mecanismo que proporciona fiabilidad a la red.

A su vez, y de acuerdo con Recomendación G.872 de la Unión Internacional de Telecomunicaciones – Sector de Estandarización de la Telecomunicaciones (UIT-T), la capa óptica se puede descomponer en tres subcapas: la capa de canal óptico, la capa de multiplexación óptica y la capa de transmisión óptica. El objetivo de la capa de canal óptico es proveer la conectividad entre puntos finales de los canales ópticos, de forma que sea transparente a los clientes. La capa de multiplexación óptica está relacionada con el agregado a la red de las señales ópticas con distintas longitudes de onda. La capa de transmisión óptica se refiere a comunicación de las señales ópticas por distintos medios físicos, como por ejemplo las transmisiones monomodo o multimodo.

⁵ Conjuntos de *lightpaths* establecidos de forma lógica. También se denomina *red de lightpaths* o *topología lógica*



Las características mencionadas –reutilización de longitud de onda, transparencia de protocolos y alta fiabilidad- hacen de las redes *wavelength-routed* idóneas para las redes de área extensa. Estas redes exigen el tratamiento de tres cuestiones fundamentales: diseño, reconfiguración, corrección de fallos. El diseño de la capa óptica parte de la demanda de tráfico y es fundamental para aprovechar eficientemente los recursos de la red (fibras y transmisores/receptores) así como para proveer calidad de servicio a los usuarios. La reconfiguración de la capa óptica se deriva de las variaciones de las demandas de tráfico. El hecho de que grandes cantidades de tráfico se transporten por la capa óptica, obliga de forma crítica a disponer de un servicio de recuperación en caso de fallos. El objeto fundamental de estudio de este Proyecto será el diseño de la capa óptica.

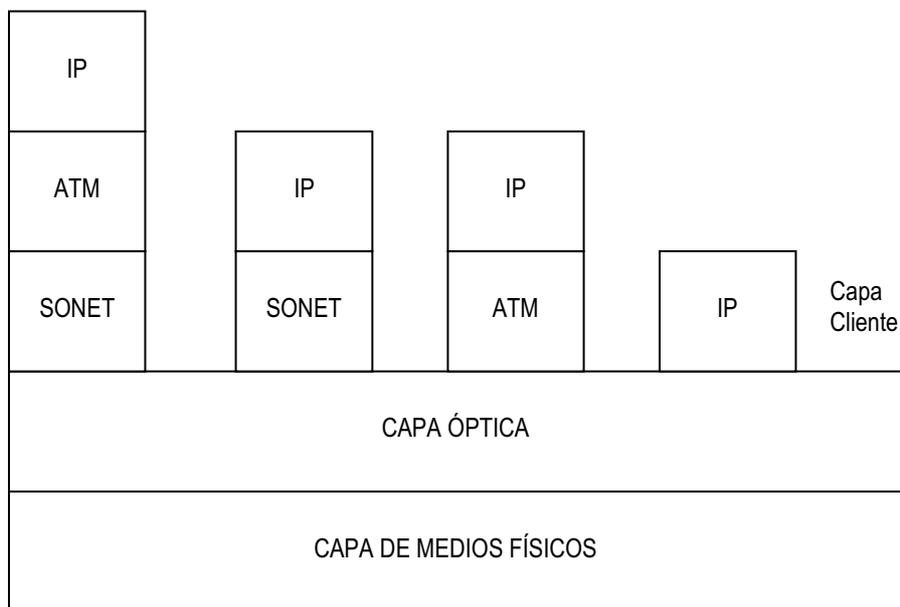


Figura 14. Posibles capas en una red de transporte WDM

1.6 PARADIGMA DE LAS REDES WDM

Como se ha descrito en la sección 1.5, los usuarios finales en una red troncal basada en fibra se comunican entre ellos por medio de canales en el dominio óptico que hemos llamado *lightpaths*. Un *lightpath* atraviesa varios enlaces de fibra para interconectar dos nodos con alta carga de tráfico mediante una técnica de “conmutación de circuitos”, además el *lightpath* se establece con independencia de la proximidad física de los nodos. Cada nodo intermedio actúa de catalizador en el

dominio óptico para el establecimiento y mantenimiento de los *lightpaths*. En la Fig. 1.15 se muestra un ejemplo de red de *lightpaths*.

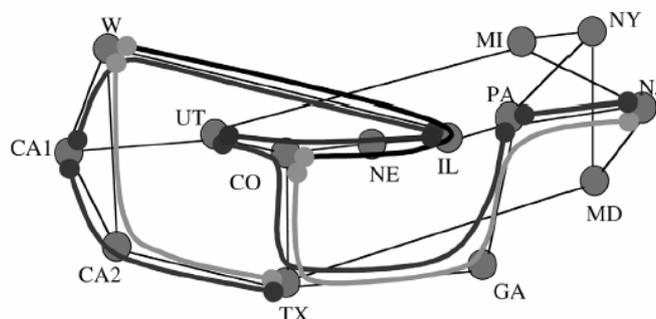


Figura 15. Red de *lightpaths* sobre una red WDM (los círculos negros indican extremos de un *lightpath*)

En una red de N nodos, si cada nodo está equipado con $N-1$ transceptores⁶ y hay disponibles suficientes longitudes de onda en cada enlace de fibra, todos los nodos se pueden conectar mediante *lightpaths* sin requerir resolución de problema alguno. Las limitaciones de recursos conocidos en todos los sistemas de comunicaciones también se aplican a las redes ópticas: los transceptores son caros por lo que se limita su número en cada nodo, y el número de canales disponibles en cada enlace de fibra se limita a W (como ya hemos dicho pueden llegar a 100). En consecuencia, tan sólo unos pocos *lightpaths* se pueden establecer en la red, de lo que se deriva un problema de optimización.

De forma simplificada y bajo las condiciones citadas, el problema se describe así: un conjunto finito de *lightpaths* se tienen que establecer sobre una topología física de red y bajo una demanda determinada de tráfico. Para ello se tienen que determinar las rutas física que atravesarán los *lightpaths* y qué longitud de onda ocupará en cada una; todo con el objetivo de que se establezcan el máximo número *lightpaths* posibles. Desde una visión parcial de cada *lightpath* es deseable que se elijan las rutas con menor número de saltos, aunque este criterio se puede varias para posibilitar que se establezcan un número mayor de *lightpaths*. Los *lightpaths* que no se puedan establecer debido a las restricciones sobre las rutas y las longitudes de onda, se bloquean; por lo que el problema de optimización se centra en minimizar la probabilidad de bloqueo.

A las condiciones del problema se añade la *restricción de continuidad de longitud de onda*, descrita en el apartado 1.5.2, de forma que si dos *lightpaths* comparten el mismo enlace de fibra no pueden usar la misma longitud de onda. Esta restricción desaparece en el caso de introducir nodos con

⁶ El término transceptor se refiere a los transmisores (láseres) y los receptores (filitros)



capacidad de conversión de longitud de onda, de forma que se pueda conmutar la longitud de onda a lo largo de la ruta del *lightpath*.

Este problema particular se denomina RWA por sus siglas en inglés: *Routing and Wavelength Assignment* y engloba todo el planteamiento descrito.

Los diseñadores de la próxima generación de redes ópticas tienen que ser conscientes de las propiedades y limitaciones de fibras y dispositivos para que los protocolos y algoritmos diseñados aprovechen las redes WDM en todo su potencial. Con frecuencia los profesionales han trabajado bajo asunciones ideales o con una visión tradicional de las redes ópticas que arrojan resultados imprácticos o incluso irrealizables.

1.7 PROBLEMA RWA (ROUTING AND WAVELENGTH ASSIGNMENT)

Como se ha comentado en apartados anteriores, las conexiones en las redes WDM *wavelength-routed* se establecen mediante *lightpaths*. Para establecer una conexión virtual entre cada par de nodos origen-destino es necesario encontrar una ruta con una longitud de onda continua. Cualquier algoritmo que resuelva la ruta a seguir y la longitud de onda a asignar para establecer el *lightpath* se denomina algoritmo RWA (*Routing and Wavelength Assignment*). Muchos problemas que se plantean en el diseño de redes WDM *wavelength-routed* tienen al RWA como subproblema. Será un imperativo del diseñador encontrar algoritmos RWA que resuelvan de forma eficiente las dos instancias del problema a la hora de establecer los *lightpaths*. De las dos instancias -ruta y longitud de onda- que aborta un algoritmo RWA será solamente la segunda la que diferencia a las redes WDM *wavelength-routed* de las redes convencionales.

1.7.1 Tráfico estático o dinámico

La demanda de tráfico sobre una red puede ser estática o dinámica. En el caso de tráfico estático, los requerimientos de las conexiones a establecer se conocen a priori. Este tráfico se puede explicitar en términos de pares de nodos origen-destino. Los pares se eligen en función de una estimación del tráfico generado entre el par de nodos. El objetivo se focaliza en asignar rutas y longitudes de onda a toda la demanda de tráfico, minimizando el número de longitudes de onda utilizadas y maximizando el



número de demandas satisfechas. Las dos instancias del problema RWA para tráfico estático se encuadran dentro del problema SLE (*static lightpath establishment*). Se ha demostrado que el problema SLE es NP-completo⁷ por lo tanto se opta por los algoritmos de tiempo polinomial cuya solución se aproxima a la óptima.

En caso de demanda de tráfico dinámica, las peticiones de conexión llegan y salen de la red de forma aleatoria. Los *lightpaths* establecidos lo hacen de forma temporal. La demanda de tráfico dinámica modela diversos escenarios en las redes de transporte. De esta forma se puede hacer necesario sustituir un *lightpath* por otro nuevo en respuesta a las modificaciones de los patrones de tráfico o como consecuencia de fallos en los componentes de la red. A diferencia del problema RWA estático, cualquier solución para un escenario de tráfico dinámico tiene que ser computacionalmente simple, ya que las peticiones y modificaciones tienen que ser procesadas de forma instantánea. Cuando se produce una nueva petición, se asigna una longitud de onda y ruta teniendo como objetivo maximizar el número de peticiones aceptadas (o, análogamente, minimizar el número de peticiones rechazadas). Los resultados de los algoritmos RWA dinámicos arrojan peores resultados que los estáticos ya que carecen de información sobre las peticiones de conexión futuras, mientras en los escenarios estáticos esta información se conoce *a priori*. Otra diferencia importante es el orden de procesamiento de las peticiones de conexión que en el caso de RWA dinámico será por estricto orden de llegada y en el RWA estático se decidirá en función del algoritmo heurístico aplicado. Un enfoque heurístico puede ser asignar longitudes de onda en orden decreciente al número de saltos, ya que según aumente el número de saltos entre un par de nodos es menos probable encontrar una longitud de onda libre común a toda la ruta.

1.7.2 Control centralizado o distribuido

Los algoritmos RWA adoptan mecanismos de control de forma centralizada o distribuida para la selección de rutas y longitudes de onda. En caso de control centralizado, un punto de control central asume las funciones de asignación de rutas y longitudes de onda y envía las señales de control apropiadas para el establecimiento y suspensión de *lightpaths*. Cuando el control es distribuido la información de actualización que describe el estado de la red es conocida por cada nodo y éstos usan una búsqueda distribuida para seleccionar rutas y longitudes de onda y así cursar las peticiones de conexión. Se puede incluso hacer un procesamiento previo de rutas y buscar las longitudes de onda

⁷ Computacionalmente intratable. Los algoritmos conocidos necesitan tiempo exponencial en el peor caso para encontrar una solución óptima.



libres en los enlaces que atraviesan. Los mensajes de control se envían a varios nodos para reservar las longitudes de onda de los enlaces que atraviesa la ruta. Una vez que se encuentra la ruta apropiada y las longitudes de onda están reservadas, se envían las señales apropiadas de control a varios nodos para configurar los *switches* en los nodos de enrutamiento para que quede establecido el *lightpath*. De forma análoga, para liberar un *lightpath*, las señales de control se envían a un conjunto de nodos por el nodo origen. Por norma general los algoritmos centralizados son apropiados para redes pequeñas y no son escalables a redes extensas. Cuando una red requiere de escalabilidad y simplicidad conviene utilizar protocolos de control distribuido.

Otro problema que presentan las redes *wavelength-routed* es la “imparcialidad” entre conexiones con distinto número de saltos. Las peticiones de conexión con mayor número de saltos tienen menos probabilidad de ser aceptadas que las que presentan un número de saltos menor. Este desequilibrio se acentúa en el caso de usar control distribuido, debido al incremento de las posibilidades de conflicto entre distintas peticiones a la hora de reservar las longitudes de onda. Por esta razón es fundamental seleccionar el protocolo de control apropiado para mejorar la imparcialidad entre conexiones en escenarios con número de saltos dispares, y que afecte lo menor posible al rendimiento y los recursos de la red.

1.7.3 Métodos de selección de ruta y longitud de onda

Los métodos más importantes de selección de ruta que se incluyen mayoritariamente en la literatura son: *enrutamiento fijo*, *enrutamiento alternado* y *enrutamiento agotado*. En el método de enrutamiento fijo, sólo se considera una ruta por cada par de nodos. Generalmente se escoge la ruta más corta. Cuando llega una petición de conexión para un par de nodos, se consulta la disponibilidad de longitudes de onda en la ruta prefijada. En el método de enrutamiento alterno, se proporciona dos o más rutas posible para cada par de nodos. Estas se utilizan de forma secuencial que muchas veces viene determinada según orden decreciente del número de saltos. El método de enrutamiento agotado busca todas las rutas posibles entre cada par de nodos. En este caso el estado de la red se representa por un grafo y se usa un algoritmo de búsqueda del camino más corto. Este último método proporciona las mejores prestaciones en comparación con los otros dos, aunque presenta una complejidad mayor. Consecuentemente, el más método más simple pero que peores resultados arroja es el enrutamiento fijo.



En función del orden en el que se buscan las longitudes de onda, los métodos de asignación se clasifican en: *la más usada*, *la menos usada*, *orden fijo* y *orden aleatorio*. En el primer método, las longitudes de onda se examinan en orden decreciente a su utilización en la red. Este método trata de comprimir el rango de longitudes de onda usadas para dejar libre rutas continuas y poder disponer de recursos (en términos de longitudes de onda) para cursar las peticiones de conexión tardías. Cuando el método utilizado prima la menos usada, las longitudes de onda se seleccionan en orden creciente de utilización. Este método reparte los *lightpaths* establecidos entre varias longitudes de onda. La idea aquí es que una petición nueva puede encontrar una ruta más corta y una longitud de onda disponible en ella. El tercer método (orden fijo) las longitudes de onda se buscan en un orden prefijado que puede ser el mismo índice nominal de cada una, de forma que se utilizan antes las longitudes de onda de menor índice. Por último se pueden seleccionar en orden aleatorio entre las longitudes de onda que estén disponibles. Los métodos *la más usada* y *la menos usada* son preferibles en redes con control centralizado. Los otros dos métodos son más adecuados para redes con control distribuido. Los resultados numéricos de la literatura indican que las redes con el primer método (más usada) obtienen un rendimiento mayor que con el segundo (menos usada) y la elección de forma fija es preferible a la elección aleatoria.

Los algoritmos RWA pueden aplicar primero el método de selección de longitud de onda sobre la ruta o viceversa; aunque también las rutas y longitudes de onda pueden ser tratadas de forma conjunta. Para cada par ruta-longitud de onda se asocia un índice de coste que se utilizará en los métodos dinámicos donde se seleccionan las rutas menos congestionadas. La ruta menos congestionada es la que tiene el máximo número de longitudes de onda libres. Este método posibilita una ruta continua en longitud de onda para las peticiones de conexión tardías.

1.7.4 Diseño de Topologías Virtuales

El diseño de la topología virtual en la capa óptica de una red de transporte consiste en establecer un grupo de *lightpaths* para un subconjunto de nodos de la red. Los *lightpaths*, como se ha dicho anteriormente, se eligen en función de las demandas de tráfico para cada par de nodos. El patrón de conectividad por medio de *lightpaths* conforma la topología virtual o lógica. En el equivalente físico de una topología virtual, un nodo se corresponde con un nodo de enrutamiento y un enlace equivale a un *lightpath*. De forma intuitiva si dos nodos están conectados por un *lightpath*, se pueden comunicar en un solo salto de luz. Debido a las limitaciones tecnológicas y de recursos en longitudes de onda y transmisores/receptores respectivamente, no es posible establecer *lightpaths* entre todos los nodos de



la red como sería deseable para enrutar todo el tráfico. A pesar de ello, si dos nodos no están conectados de forma directa por un *lightpath* pero si por una secuencia de ellos, se pueden comunicar entre ellos en una conexión multi-salto (óptico). En este caso el reenvío de mensajes entre dos *lightpaths* consecutivos se implementa mediante procesamiento electrónico.

Como ejemplo de lo anterior, si dos *routers* IP están conectados por un *lightpath* son vecinos en la topología virtual independientemente de que estén conectados en la topología física por un enlace. El tráfico IP entre dos *routers* que no tenga *lightpath* entre ellos tiene que ser procesado electrónicamente en los *routers* intermedios.

En definitiva el tráfico entre los pares de nodos se enruta sobre la topología virtual en uno o más saltos. La topología virtual transporta el tráfico de la red y se diseña para optimizar alguna métrica, como el retardo medio de un mensaje o la congestión de la red. El retardo de un mensaje se define en función del número de *lightpaths* que atraviesa. La congestión de la red se define como la máxima carga en cualquier *lightpath*, medida esta en términos de la cantidad de tráfico transportado por ese *lightpath*. Conviene notar que el flujo que transporta un *lightpath* puede provenir de pares de nodos diferentes.

En el siguiente capítulo se profundizará en el diseño de topologías virtuales para redes WDM de forma teórica, de los conceptos que implica y las variables de diseño. El establecimiento de la topología virtual en la red de *lightpaths* supone la solución al problema RWA descrito. En los capítulos que abordan la herramienta práctica de este Proyecto se implementan diversos algoritmos heurísticos que se aproximan a la solución óptima en el diseño de la topología virtual.

1.8 OBJETIVOS Y DISTRIBUCIÓN DEL DOCUMENTO

Como se ha visto en el presente capítulo introductorio, la tecnología óptica WDM es un conjunto de protocolos, algoritmos, estándares y medios físicos que se articulan en un modelo de capas y requiere la convergencia de diversas áreas de conocimiento en el estudio clásico de redes además de conceptos propios e inherentes a las redes ópticas. La estandarización teórica de las redes WDM se encuentra en su etapa primera, donde las líneas actuales de investigación apuntan incluso a unas



expectativas de desarrollo superiores a las iniciales, pero que su extrema juventud hace que la tecnología se encuentre en continua evolución.

En este proceso se enmarca el presente Proyecto Fin de Carrera que si bien encontrará su justificación en el contexto de las redes ópticas WDM, introducirá elementos específicos en ciertas aplicaciones para evaluar el rendimiento y las posibilidades de diversos algoritmos propuestos en la literatura.

Como se ha explicado en los apartados anteriores es en el diseño de la capa óptica -transparente a los medios ópticos e independiente del acceso a la red- donde se centra el grueso de la investigación de las redes ópticas WDM. El diseño de la capa óptica gira en torno al problema RWA y sus implicaciones, habiéndose abordado de dos formas diferentes a consecuencia de su complejidad computacional –ya que se trata de un problema NP-completo-, la resolución por Programación Lineal o mediante Algoritmos Heurísticos. Ambos métodos tienen el objetivo encontrar la red de *lightpaths*, o topología virtual, que minimice alguna de las métricas propuestas.

El presente Proyecto aborda la implementación en lenguaje Matlab de diversos algoritmos heurísticos propuestos en la literatura de investigación que resuelven el problema RWA de una manera subóptima. Su integración en una herramienta programada en un entorno GUI permite realizar diseños de topologías virtuales, evaluar la capacidad de un algoritmo en un escenario simulado y comparar los algoritmos entre en las coordenadas del mismo problema. Este Proyecto es complementario de otro PFC que aborda la resolución del mismo problema mediante programación lineal, pero que como sabemos impone limitaciones en sus dimensiones en atención a su complejidad exponencial. La comparación de los resultados de ambos trabajos, en aquellos escenarios que lo permitan, es una medida de la aproximación de los algoritmos heurísticos aquí presentados a la solución óptima.

En el segundo capítulo se define el RWA y los tipos de problemas e instancias que incluye, siendo la principal el diseño de topologías virtuales. La resolución de la topología virtual incluye en este capítulo las notaciones, definiciones y variables necesarias para su posterior implementación.

El tercer capítulo formula teóricamente los algoritmos heurísticos seleccionados en la literatura e incluye las fichas de las funciones implementadas en Matlab. En el capítulo siguiente se describe la



herramienta programada para el manejo de los algoritmos: sus funciones, variables de entrada y salida, funciones auxiliares y complementarias.

En el capítulo 5 se presentan los resultados obtenidos sobre distintos escenarios, donde se ha querido incluir un amplio abanico de topologías físicas y patrones de tráfico que permita jugar con una rica casuística. Por último, y para enmarcar el Proyecto en las líneas de investigación citadas al comienzo se obtendrán las conclusiones de este trabajo y su idoneidad como aplicación docente, al mismo tiempo que se indicará su evolución futura.



CAPÍTULO 2

Problema RWA. Diseño de Topologías Virtuales

2.1 PROBLEMA RWA (ROUTING WAVELENGTH ASSIGNMENT)

2.1.1 Clasificación de los algoritmos RWA

Las redes WDM *wavelength-routed* de conmutación de circuitos están concebidas para redes de área extensa. Una petición de conexión implica una comunicación entre el nodo origen y el nodo destino, por la que se produce la transmisión de datos, y se libera cuando esta ha finalizado. En este tipo de redes la comunicación se implementa por medio de un *lightpath*, identificado de forma unívoca por una ruta física y una longitud de onda. La *restricción de continuidad de longitud de onda* impone que la longitud de onda utilizada en todos los enlaces que el *lightpath* atraviesa tiene que ser la misma. Esta restricción es privativa de las redes WDM y degrada considerablemente el rendimiento de la red, incrementando la probabilidad de bloqueo. La probabilidad de bloqueo se define como el porcentaje de peticiones rechazadas.

En un entorno de tráfico dinámico, donde las peticiones de conexiones entran y salen de la red de forma aleatoria, puede ocurrir la siguiente situación: una petición de conexión para el par de nodos $\langle s, d \rangle$ encuentra una ruta disponible, pero se bloquea al no encontrar una longitud de onda común a toda ella. Se ilustra en la Fig. 2.1. La figura muestra dos *lightpaths* p_1 y p_2 con longitudes de onda w_0 y w_1 , respectivamente. Se asume que hay dos longitudes de onda por fibra. Supongamos que llega una petición al nodo 0 para conectarse al nodo 2 y que, por lo tanto, existe una ruta disponible, pero para esta ruta en el primer enlace (0-1) está libre w_1 y en el segundo enlace (1-2) está libre w_0 . La petición es rechazada por no existir continuidad de longitud de onda. Si el nodo 1 estuviera equipado con capacidad de conversión de longitud de onda, la petición podría ser aceptada. En el caso de que todos



los nodos de la red tuvieran esta capacidad, el funcionamiento sería el mismo que una red de conmutación de circuitos convencional. En este Proyecto no consideraremos redes con capacidad de conversión.

Con la llegada de una petición de conexión, un algoritmo de enrutamiento de longitud de onda (WR – *wavelength routing*) se encarga de encontrar un *lightpath* para satisfacer esa demanda. El diseño de un algoritmo WR es crítico para mejorar el rendimiento de la red en términos de probabilidad de bloqueo de las conexiones. Un algoritmo WR también se denomina algoritmo RWA (*routing and wavelength assignment*) y tiene dos componentes: selección de ruta y de longitud de onda. La ruta se elige en atención a criterios de coste como el número de saltos y la longitud de onda en función del factor de utilización en toda la red. Se han propuesto en la literatura diversos algoritmos para la elección del mejor dúo de ruta y longitud de onda, aunque difieren en sus políticas de selección, lo que determinará su clasificación. La resolución del problema se puede iniciar por una de las dos instancias o de forma conjunta, ya que el orden de selección no altera el funcionamiento.

Una red WDM con N nodos y W longitudes de onda por fibra se puede representar como un grafo $G(V,E)$, donde V es un conjunto de vértices y E un conjunto de enlaces. Los vértices se corresponden a los nodos de la red y los enlaces a los canales de longitudes de onda en los enlaces físicos entre nodos. El grafo G tiene W subgrafos (capas), G_i , $0 \leq i < W$ cada uno correspondiente a una longitud de onda. Un vértice de un subgrafo G_i se corresponde a un nodo en la red y un enlace se corresponde a una longitud de onda i en la fibra que conecta los nodos extremos del enlace. En un subgrafo G_i se puede encontrar de forma sencilla una ruta de un nodo a otro a una longitud de onda i .

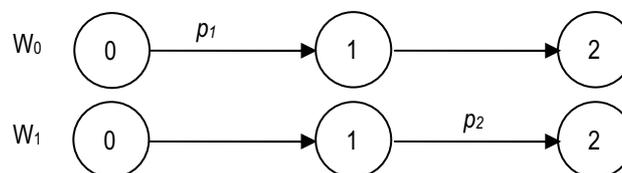


Figura 16. Efecto de la restricción de continuidad de longitud de onda en el bloqueo

2.1.2 Algoritmos RWA

En el primer capítulo se introdujeron varios algoritmos para la selección de ruta y longitud de onda, y en este apartado se va a tratar algunos de los algoritmos de enrutamiento de longitud de onda (WR) presentes en la literatura. Los primeros cuatro algoritmos seleccionan la ruta según alguna medida de



coste y después asignan una longitud de onda de las disponibles usando un método de selección. El último algoritmo WR que se presenta, selecciona ruta y longitud de onda de forma conjunta. En este caso, el coste de una ruta depende de la longitud de onda considerada; o en otras palabras, el coste se asocia a cada par ruta-longitud de onda y se elige el que presente menor coste.

1. Enrutamiento fijo (EF)
2. Enrutamiento fijo alterno (EFA)
3. Enrutamiento agotado (EA)
4. Enrutamiento de mínima congestión (EMC)
5. Enrutamiento conjunto ruta-longitud de onda (JWR – *joint wavelength-route*)

2.1.2.1 Enrutamiento fijo

Este enrutamiento está considerado como el más simple de todos los algoritmos. Para cada par de nodos p en la red, se procesa una ruta fija R^p de forma previa a las peticiones de conexión, esta ruta es normalmente la más corta. Cuando llega a la red una petición de conexión entre los nodos del par p , el algoritmo comprueba si se dispone de una longitud de onda común en todos los enlaces de la ruta R^p . Si hay más de una longitud de onda disponible, se puede usar un algoritmo de selección de longitud de onda para determinar la más apropiada. Las ventajas de este algoritmo radican en su rapidez de configuración y tiempo de procesado. Si H es la longitud de la ruta fija más larga para cualquier par de nodos, entonces tardaría $O(HW)$ unidades de tiempo.

El principal inconveniente de este algoritmo es que presenta un rendimiento muy pobre, ya que sólo tiene en cuenta una ruta entre un par de nodos p , incluso si existen varias disponibles. Se puede dar la situación de no haber longitudes de onda disponibles en la ruta fija dispuesta pero si haberlas en una ruta alternativa entre los dos nodos, en este caso la petición sería rechazada con la consiguiente degradación de las prestaciones de la red. En condiciones de baja carga este algoritmo funciona sensiblemente mejor ya que las probabilidades de encontrar una longitud de onda libre en la ruta fija son altas. En consecuencia, conforme aumenta la carga en la red el algoritmo tiene peor funcionamiento. Otros factores que influyen en el desempeño de este algoritmo son la topología y la conectividad. En una red con mucha densidad de conexiones, donde hay disponibles muchas rutas entre cada par de nodos, los resultados de este algoritmo son pobres.



2.1.2.2 Enrutamiento fijo alternativo

El algoritmo de enrutamiento fijo alternativo es una extensión del EF. Para cada par de nodos, se proveen un conjunto de rutas válidas (más de una). Los elementos de este conjunto se denotan por $R_0^p, R_1^p, \dots, R_{K-1}^p$. Estas rutas como en el caso del EF se computan previamente y sólo con los datos de la topología física. El conjunto de rutas seleccionadas es sólo un subconjunto de todas las rutas posibles para cada par de nodos. Cuando llega una petición de conexión para el par p , se consultan las rutas candidatas y se selecciona la primera de ellas con un coste finito. Las métricas de coste más usadas son el número de saltos o el retardo. Si no es posible encontrar una ruta de coste finito, o bien una longitud de onda común a los enlaces, entonces la petición se rechaza. Igual que en el EF, si hay más de una longitud de onda disponible para la ruta seleccionada, se puede usar un método de selección para determinar la más adecuada.

Aunque este algoritmo es más complejo que el EF, también tiene como ventaja su simplicidad y su corto tiempo de configuración. Si H es la longitud de la ruta válida más larga para cualquier par de nodos, el algoritmo se resuelve en $O(KHM)$ unidades temporales. En general, y al no incluir todas las rutas posibles entre un par de nodos, los resultados de este algoritmo son mejorables.

2.1.2.3 Enrutamiento agotado

Los algoritmos de enrutamiento agotado funcionan de forma más eficiente que los algoritmos EF y EFA vistos hasta ahora. En lugar de predeterminar las rutas válidas para cada par de nodos, guarda la información sobre el estado de la red dinámicamente, actualizándolo con las variaciones de la demanda de tráfico. Cuando llega una petición de conexión para un par p , elige la mejor ruta entre todas posibles, basándose en algunos criterios de coste. De esta forma, al incluir todas las rutas posibles en la decisión, se mejora la tasa de aceptación de peticiones.

La red se puede modelar como un grafo con W subgrafos, cada uno correspondiente a una longitud de onda. Un algoritmo convencional de búsqueda de la ruta más corta (por ejemplo, el Algoritmo de Dijkstra) determina la ruta de mínimo coste en cada uno de los subgrafos, de forma que se puede elegir la mejor ruta en un momento determinado. Como es natural, y al considerar todas las rutas posibles, el algoritmo EA tiene un rendimiento mejor que los dos vistos con anterioridad. A pesar de ello, presenta algunos inconvenientes como la lentitud. En el peor caso la complejidad de este algoritmo es $O(N^2W)$.



De forma general el EA es apropiado para implementaciones de red centralizadas y menos aconsejable para arquitecturas distribuidas.

2.1.2.4 Algoritmo de Dijkstra

El Algoritmo de Dijkstra se propone para encontrar la ruta más corta desde un nodo s a un nodo d sobre un grafo directo. Se asume que los pesos de los enlaces son no negativos y los nodos están numerados de 0 a $N-1$. Se va a describir el algoritmo para encontrar la distancia más corta entre s y d , y puede modificarse fácilmente para encontrar el camino correspondiente.

El algoritmo tiene un máximo de $N-1$ iteraciones, y en cada una de ellas se busca la distancia más corta hasta un nuevo nodo. Se denota como $w_t(y,x)$ el peso de cada enlace desde y a x , si existe. En cada iteración, el valor de $dist(x)$ indica la longitud del camino más corto conocido desde s a x , atravesando sólo los nodos cuyo camino más corto haya sido determinado hasta este momento. Un nodo se etiqueta como *permanente* si se ha encontrado su distancia más corta. En otro caso se etiqueta como *provisional*.

Inicialmente, el nodo s se etiqueta como permanente y el resto como provisionales. El hecho de que el valor de $dist(x)$ sea infinito indica que no hay enlace entre s y x ; en otro caso su valor se establece en $w_t(s,x)$. El nodo con el valor menor de $dist()$ entre todos los nodos etiquetados como provisionales se escoge como permanente y se llama y . Ahora el valor $dist()$ de todos los nodos etiquetado como provisionales se actualiza según:

$$dist(x) = \min[dist(x), dist(y) + w_t(y,x)]$$

El procedimiento anterior se repite iterativamente hasta que el nodo d se etiqueta como permanente. Puede haber como máximo $N-1$ iteraciones y en cada iteración $O(N)$ operaciones. Por lo tanto, en el peor caso la complejidad del algoritmo es $O(N^2)$.

Este algoritmo se puede extender para encontrar el camino más corto hasta todos los nodos desde un nodo origen dado. Siempre que un nodo (llamado x) reciba un nuevo valor de $dist()$ de otro nodo (llamado y), el nodo y se declara padre del nodo x . Cuando el nodo x está etiquetado permanentemente, se puede encontrar su ruta más corta hasta el nodo origen atravesando los nodos padres.



2.1.2.5 Búsqueda Breadth-First

Después de lo aprendido sobre las redes WDM se puede afirmar que utilizar el número de saltos como métrica mejora el rendimiento de la red en términos de la probabilidad de bloqueo de las conexiones. Como se ha comentado anteriormente, esto se debe a que usar pocos saltos, posibilita la utilización de más enlaces para las peticiones tardías. Se puede usar el algoritmo de búsqueda Breadth-First para encontrar la ruta con el mínimo número de saltos en un tiempo $O(|E|)$, siendo $|E|$ el número de enlaces del grafo. En el peor caso, $|E|$ es $O(N^2)$, pero en la mayoría de las redes prácticas el número de enlaces es mucho menor que el cuadrado del número de nodos. A continuación se describe el algoritmo para encontrar el número de saltos de la ruta más corta entre un nodo s y un nodo d , a partir de ese dato es fácilmente calculable aquella.

Llamamos Q a una cola de nodos en una determinada etapa del algoritmo. Inicialmente Q contiene sólo el nodo origen s , cuyo valor $dist()$ está inicializado a cero. Todos los nodos no están marcados de inicio. El procedimiento siguiente se repite hasta que el nodo d aparezca a la cabeza de la cola Q .

Tomar el nodo del primer lugar de la cola Q . Este no nodo se marca como y , y significa que se ha encontrado el coste de la ruta más corta hasta él. En todos los nodos no marcados x para los que exista un enlace desde el nodo y , se actualiza el valor $dist()$ según la siguiente fórmula, por último se añaden a la última posición de Q .

$$dist(x) = dist(y) + 1$$

El algoritmo anterior se puede extender para encontrar el camino más corto a todos los nodos desde un nodo origen dado. Cuando un nodo (llamado x) actualiza el valor $dist()$ desde otro nodo (llamado y) el nodo y se declara padre del nodo x . Cuando se marca el nodo x , se puede encontrar su ruta más corta hasta el nodo origen atravesando la sucesión de nodos padre de la ruta.

2.1.2.6 Enrutamiento de mínima congestión

Este algoritmo RWA selecciona la ruta de mínima congestión entre las rutas posibles que conectan un par de nodos p . La congestión de la ruta la determina el número de longitudes de onda libres en toda la ruta. Cuanto mayor sea el número de longitudes de onda libres, menos congestionada estará la ruta. El algoritmo que se presenta se basa en una aproximación de enrutamiento alterno.



Se seleccionan K rutas de entre todas las posibles computadas previamente para cada par de nodos p , que se denotan por $R_0^p, R_1^p, \dots, R_{K-1}^p$. Cuando llega una nueva petición de conexión, se calcula el coste asociado a cada ruta seleccionada. Este coste viene dado por la congestión de la ruta. En caso de existir un empate de costes, se toma la ruta con menor número de saltos. Una vez que esté determinada la ruta, se puede usar un método de selección de longitud de onda para determinar la más apropiada. La idea intuitiva para seleccionar la ruta con menor congestión se basa en la preservación de camino de continuidad de onda para las peticiones futuras, ya explicada anteriormente. La complejidad temporal de este procedimiento es $O(KHW)$, donde H es el contador de saltos de la ruta más larga seleccionada. Es previsible que este algoritmo funcione mejor que el ER y el ERA. A pesar de ello, y ya que está basado en enrutamiento alterno, presenta una probabilidad de bloqueo mayor que las aproximaciones de enrutamiento agotado.

2.1.2.7 Enrutamiento conjunto ruta-longitud de onda

Todos los algoritmos RWA presentados hasta ahora resolvían las dos instancias del problema de forma independiente. Aunque en todos ellos el problema de la ruta se ha resuelto en primer término, también es posible hacerlo a la inversa, originándose nuevos algoritmos. A diferencia de los anteriores el algoritmo JWE considera el coste asociado a cada par ruta-longitud de onda y selección el de menor coste. La función de coste asociada a cada par en sus dos vertientes tiene en cuenta factores como la utilización de la longitud de onda en la red, el número de saltos de la ruta, o la congestión en la misma.

El algoritmo JWR se basa en una aproximación de enrutamiento alterno. Como se ha presentado en los algoritmos que presentaban este método, es necesaria una selección previa de K rutas candidatas de entre todas las posibles para cada par de nodos p y se denotan por $R_0^p, R_1^p, \dots, R_{K-1}^p$. Se llama $A(\omega_i)$ el número de enlaces en donde la longitud de onda ω_i está actualmente disponible. El número de saltos y el número de longitudes de onda libres en R_j^p se denotan como $L(R_j^p)$ y $F(R_j^p)$ respectivamente. Entonces el coste asociado a un par longitud de onda-ruta viene dado por

$$C(w_i, R_j^p) = \alpha_1 A(w_i) + (1 - \alpha_1) \{ \alpha_2 [W - F(R_j^p)] + (1 - \alpha_2) L(R_j^p) \}, 0 \leq \alpha_1 \text{ y } \alpha_2 \leq 1$$

Se pueden escoger los valores apropiados para las constantes α_1 y α_2 para conformar la función de coste deseada. Como ejemplo, la elección de un valor alto de α_1 (pongamos $\alpha_1=1$) dará



preferencia en primer lugar a las longitudes de onda más utilizadas. Al contrario, un valor de $\alpha_1=0$ preferirá la ruta de mínimo coste, ignorando el nivel de utilización de las longitudes de onda. En resumen y dada su elaboración y complejidad, este algoritmo trata de sintetizar las ventajas de los algoritmos EF, EFA y EMC.

2.2 DISEÑO DE TOPOLOGÍAS VIRTUALES

2.2.1 Introducción a las topologías virtuales

Últimamente el interés por el diseño de topologías lógicas o virtuales para redes WDM y como respuesta al problema RWA ha ido creciendo. El diseño de una topología virtual resuelve las dos instancias del problema. Una topología virtual sobre una red de área extensa WDM se concibe como la combinación más eficiente de recursos ópticos y electrónicos presentes en la misma. La arquitectura virtual utiliza transparencia de canales entre nodos llamados *lightpaths*, porque atraviesan varios enlaces físicos pero la información que transportan es transmitida en dominio óptico de extremo a extremo. La forma corriente de implementar un *lightpath* es mediante la elección de una longitud de onda determinada en cada uno de los enlaces físicos que atraviesa. Como se introdujo en el Capítulo I la condición que obliga al mantenimiento de la longitud de onda a lo largo del *lightpath* se conoce como *restricción de continuidad de longitud de onda*. Debido a las limitaciones en hardware y número de longitudes de onda por fibra, no es posible establecer *lightpaths* entre todos los pares de nodos de la red. El conjunto limitado de *lightpaths* que se establecen sobre la topología física constituye la topología virtual (o lógica).

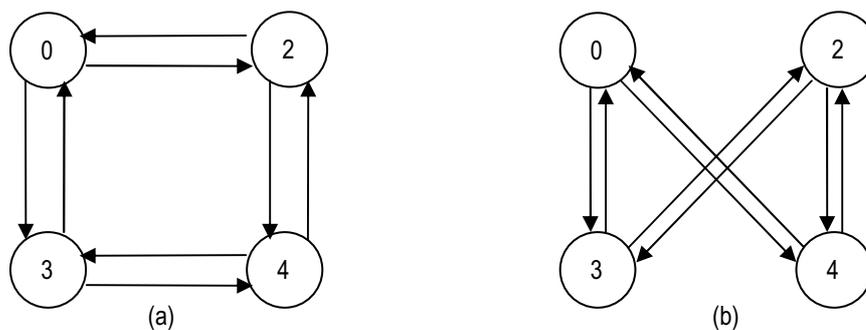


Figura 17. Ilustración de una topología física y virtual. (a) Una topología física. (b) Una posible topología virtual.

Como es sabido las redes WDM en su conjunto implican conjuntamente recursos ópticos y electrónicos, y será el equilibrio en la utilización de ambos lo que determine un buen diseño de topología virtual. El establecimiento de un *lightpath* no implica cambio del dominio óptico al electrónico, por lo que el tráfico que transporta no se verá afectado por la limitación de ancho de banda que impone el último.

El uso tradicional de la fibra óptica consistía en un enlace punto a punto con una longitud de onda, pero ha sido a partir de las redes WDM cuando se aumentado de forma espectacular el ancho de banda y se han reducido sensiblemente los retardos. La Fig. 2.2 muestra una red donde se han establecido ciertos *lightpaths* (marcados con trazo discontinuo) para permitir una comunicación transparente entre nodos que no están directamente conectados en la topología física.

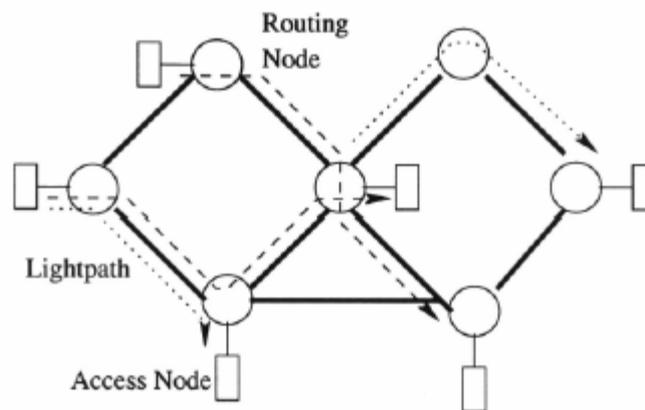


Figura 18. Una red WDM. Los nodos de enrutamientos están interconectados por fibras punto a punto. Las líneas punteadas indican los *lightpaths*.

El concepto de la topología virtual introduce cierta independencia sobre la topología física debido a las muchas que se pueden diseñar sobre el mismo escenario. Será entonces labor del diseñador encontrar la topología virtual que mejor rendimiento tenga sobre una topología física dada, y bajo ciertas condiciones de tráfico.

En general los problemas de diseño de topologías virtuales se pueden formular como problemas de optimización maximizando el *throughput* o cualquier métrica de interés. La solución exacta a este problema de optimización es NP-completa y se necesitan algoritmos heurísticos para encontrar soluciones aproximadas. Para ello el problema se descompone en cuatro subproblemas que se abordan de forma independiente



A parte de lo dicho, hay al menos dos cuestiones relacionados con el diseño de topologías virtuales. La primera de ellas tiene que ver con el coste que requiere establecer y operar la red y supone una consideración muy importante desde el punto de vista práctico. Por eso, aunque una topología virtual presente un retardo más bajo y un rendimiento más alto que otra, puede ser que involucre muchos recursos de red (conmutadores, transmisores, receptores...) y por lo tanto su coste asociado sea mucho mayor. La elección de una topología virtual como “mejor” tiene que tener en cuenta, no sólo las variables de optimización, sino también aspectos fundamentales que implica su planificación.

El segundo aspecto que influye en la planificación y diseño de las redes ópticas que usan topologías virtuales es su reconfiguración. La adaptabilidad es uno de los puntos fuertes de las redes ópticas en general y de la topología virtual en particular. Una topología virtual se diseña en función de los patrones de tráfico de la topología física. Cuando se posibilita diseñar una nueva topología virtual en un escenario cambiante se dota a la red adaptabilidad (cuando cambian los patrones de tráfico), autoreparación (cuando se producen cambios en la topología física debido a fallos de los componentes de red) y escalabilidad (cuando cambia la topología física por la adición o eliminación de componentes). La propiedad de rediseñar topologías virtuales es fundamental en la solución del problema y su flexibilidad como solución dinámica.

Las ventajas de la tecnología óptica radican en la conmutación y la transmisión, no en el procesado o el almacenamiento. Por eso, la conmutación y la transmisión en el dominio electrónico son más apropiadas en las redes de acceso donde no se requieren grandes anchos de banda, pero si alto rendimiento en el procesado. Consecuentemente el contexto del diseño de topologías virtuales son las redes de transporte.

2.2.2 Arquitectura y notaciones

La multiplexación por división de longitud de onda (WDM por sus siglas en inglés, wavelength division multiplexing) permite el uso de diferentes longitudes de onda sobre una fibra óptica para implementar canales separados. En un nodo de enrutamiento de la fibra se usa un add/drop multiplexer (ADM) para modificar el flujo a través de una fibra. En ciertas longitudes de onda el ADM pasa el tráfico sin interrumpirlo, mientras otras se añaden o eliminan al transportar tráfico de origen o destino en el nodo. Un Wavelength Router (WR) es un dispositivo más completo que el ADM, ya que permite tomar las señales de los puertos de entrada y enrutarlas a cualquier puerto de salida, con independencia de



su longitud de onda. Un WR con N puertos de entrada y N de puertos de salida, manejando W longitudes de onda hace las veces de W conmutadores $N \times N$ independientes. Estos conmutadores están precedidos de un multiplexador y seguidos de un demultiplexador para implementar un WR. Un conversor de longitudes de onda es un dispositivo óptico que se puede usar en un router óptico, y que permite cambiar la longitud de onda por la que se transporta el flujo actual a otra sin intervención de procesado electro-óptico. La conversión de longitudes de onda permite crear canales transparentes sobre distintas longitudes de onda y distintos enlaces físicos. Es posible una capacidad de conversión limitada para ciertos nodos. La conversión completa implica que todas las longitudes de onda de entrada se pueden convertir a cualquier longitud de onda de salida. La conversión limitada permite que un conjunto de longitudes de onda de entrada se puedan convertir a algunas longitudes de onda de salida, que no pueden ser todas las disponibles por lo menos para una longitud de onda de entrada. Si un nodo tiene capacidad de convertir longitudes de onda, su configuración entrará dentro del diseño global de la topología virtual.

Poder convertir longitudes de onda implica que la rigidez introducida por la restricción de continuidad de longitud de onda, desaparece. Por eso la administración de longitudes de onda es más eficiente, a pesar de que el coste de los conversores aumenta y también la complejidad del problema. Se puede mantener un equilibrio del coste dotando a los nodos de conversión limitada y limitando el número de nodos que tenga capacidad de conversión. Esto introduce nuevas variables al problema de la topología virtual: especificar el grado de conversión para cada nodo y la ubicación de los conversores.

La topología virtual diseñada e implementada no sólo condiciona el comportamiento de la red en términos de la métrica optimizada, sino también acarrea un coste de componentes que debe de ser tenido en cuenta a la hora de abordar el problema. Para ello es necesario un modelo de coste de red. El primer objetivo de los estudios realizados es conseguir una medida del coste relativo parcial de algunos componentes en el coste total del sistema, y proveer un índice de eficiencia económica que permita determinar el coste de la implementación de una topología virtual.

2.2.2.1 Notaciones y definiciones generales

En esta sección vamos definir la terminología y notaciones genéricas que se adoptan para abordar el problema de la topología virtual y que se utilizarán en la implementación de los algoritmos de los



capítulos siguientes. Después de la formulación del problema será necesario definir otro conjunto de variables específicas que se adaptarán al tipo de función objetivo y optimización abordada.

1. **Topología Física.** Un grafo $G_p(V, E_p)$ en donde cada nodo de la red es un vértice y cada enlace de fibra entre dos nodos es un arco. Cada enlace de fibra es un enlace físico, o simplemente un enlace. Se asume que el grafo es no dirigido, porque los enlaces son bidireccionales. El peso asociado a cada arco del grafo se relaciona con la distancia o el retardo de propagación entre nodos.
2. **Matriz de Tráfico.** Una matriz que especifica el tráfico medio entre cada par de nodos en la topología física. Si hay N nodos en la red, la matriz de tráfico es una matriz $N \times N$ $T = [t^{s,d}]$ donde $t^{s,d}$ es el tráfico entre el nodo s al nodo d en las unidades apropiadas, como paquetes por segundo ofrecidos o ancho de banda requerido. La matriz de tráfico provee en términos numéricos como se distribuye el tráfico sobre la red entre cada par de nodos origen-destino, o lo que es lo mismo el patrón de tráfico de la red.
3. **Lightpath.** Un *lightpath* es un canal óptico transparente entre dos nodos. El tráfico que enruta el *lightpath* no se convierte al dominio electrónico en el trayecto que media entre los nodos, es una señal óptica continua entre los nodos origen y destino. La *restricción de continuidad de longitud de onda* impone que el *lightpath* sea una secuencia de enlaces físicos a la misma longitud de onda a lo largo de la ruta.
4. **Indicador de Enlace.** En el caso de exista un enlace en la topología física desde un nodo x a otro nodo y , se denota por $I_{x,y}$ y será 1 cuando exista el enlace y 0 en caso contrario.
5. **Indicador de Lightpath.** En el caso de exista un enlace en la topología lógica desde un nodo i a otro nodo j , se denota por p_{ij} y será 1 cuando exista el *lightpath* y 0 en caso contrario. Se asume que puede existir más de un *lightpath* entre el mismo par de nodos.
6. **Topología Virtual.** Un grafo $G_v(V, E_v)$ en donde el conjunto de nodos es el mismo que el grafo $G_p(V, E_p)$ de la topología física, y cada *lightpath* es un arco. También se utilizan los términos topología lógica y enlaces lógicos, respectivamente. Se asume que el grafo es dirigido, ya que



los *lightpaths* existen desde un nodo A hasta otro B pero no en sentido contrario. Este grafo también está ponderado, y cada peso indica la *distancia del lightpath*.

7. **Distancia de Lightpath.** El retardo de propagación sobre el *lightpath*, se denota por d_{ij} para el *lightpath* desde el nodo i al nodo j . Es la suma de los retardos de propagación en los enlaces físicos que conforman el *lightpath* en la topología virtual.
8. **Grado Físico.** El grado físico de un nodo es el número de enlaces que se conectan a ese nodo provenientes de otros nodos.
9. **Grado Virtual.** El grado virtual (o lógico) es el número de *lightpaths* que se conectan a ese nodo provenientes de otros nodos. El número de *lightpath* de entrada y salida, al considerarse unidireccionales, pueden ser diferentes; por lo que se denotan respectivamente *grado virtual de salida* y *grado virtual de entrada*. Como usualmente serán iguales, se denotará en general como *grado virtual*. Si este grado es el mismo para todos los nodos de la red, se dice que es el *grado virtual de la red*. El grado virtual está determinado en parte por el grado físico, pero también está influido por la capacidad del nodo.
10. **Carga de Tráfico Virtual.** Cuando se establece una topología virtual sobre la topología física, se tiene que enrutar sobre los *lightpath* el tráfico desde cada nodo origen hasta cada nodo destino. El tráfico agregado resultante sobre un determinado *lightpath* es la carga ofrecida a ese enlace lógico. Si existe un *lightpath* desde el nodo i hasta el nodo j , la carga ofrecida a ese *lightpath* se denota por λ_{ij} . La componente de esta carga debida al tráfico desde el nodo origen s al nodo destino d se denota por $\lambda_{ij}^{(sd)}$. El máximo de las cargas lógicas se llama **congestión** y se denota por $\lambda_{\max} = \max_{ij} \lambda_{ij}$.

2.2.2.2 Arquitectura

En esta sección vamos a describir en detalle la red WDM *wavelength-routed* explicada anteriormente e ilustrada en la Fig 2.3. La red consiste en varios nodos de enrutamiento conectados entre si mediante fibras ópticas, especificada por la topología física. Cada nodo de enrutamiento puede tener un nodo de acceso conectado a él aunque para el diseño de topología virtual sólo importa el tráfico agregado entre los nodos de enrutamiento (suma de los *lightpaths*). Si asumimos que cada nodo



de enrutamiento tiene un nodo de acceso conectado a él podemos referirnos a ambos simplemente como nodos. La matriz de tráfico especifica el tráfico agregado desde todos los nodos a cada uno de los demás.

Los enlaces de fibra conectados a cada nodo permiten un número específico de longitudes de onda, W . Cada nodo está equipado con un WR capaz de enrutar estas W longitudes de onda. En general no se asume capacidad de conversión de longitud de onda en ningún nodo.

Los *lightpaths* se establecen sobre la topología física estableciendo la topología virtual. Un *lightpath* se forma configurando los nodos origen y destino en una transmisión a una longitud de onda determinada, escogiendo una ruta entre ellos y configurando los WR de cada nodo intermedio para reenviar la señal óptica sin cambiar la longitud de onda al nodo siguiente. Dos *lightpaths* que comparten un mismo enlace físico deben de funcionar a distintas longitudes de onda. El número total de longitudes de onda utilizadas en todos los enlaces debe de ser igual o menor que W . Es usual que todos los nodos de la red tengan el mismo número de *lightpaths* de entrada y salida, por lo que se habla de *grado lógico* (o virtual) de la red.

El tráfico se enruta desde cada nodo origen a destino sobre un solo *lightpath* en caso de que exista entre ellos, o sobre una secuencia de más de uno en varios saltos lógicos. Para simplificar el problema de optimización normalmente se asume que el tráfico para un solo par origen-destino puede ser bifurcado sobre distintas rutas. El objetivo de crear una topología virtual es asegurar que se pueda enrutar el máximo tráfico posible con los mínimos conversores opto-electrónicos a lo largo de la ruta. El caso extremo sería establecer *lightpaths* entre todos los pares, aunque normalmente esto estaría limitado por el número de longitudes de onda disponibles. En esta situación tenemos la topología virtual que es exactamente igual a la topología física, por lo que la conversión opto-electrónica se produce en cada nodo intermedio. Con un diseño adecuado se puede conseguir que el número de conversiones opto-electrónica no sea excesivamente elevado. El hecho de que en las redes WAN el retardo de propagación domine sobre el retardo de encolamiento (ya que la carga de los enlaces no se aproxima al límite de su capacidad), este último se desprecia en la formulación del problema de optimización.

El objetivo del diseño de la topología virtual es optimizar una determinada métrica. Por eso, cuando se formula matemáticamente el problema se tiene en cuenta una variable para minimizar, e.g. la congestión de la red o el retardo medio del paquete. El número de longitudes de onda se toma como



constante. Si las dos minimizaciones se quieren llevar a cabo de forma conjunta, una de ellas se establece como constante en función de las características de la red física. En general ambas son importantes porque la relajación respecto a la congestión origina una topología virtual muy parecida a la topología física, y demasiado poco énfasis en el retardo puede producir poca relación entre ambas topologías y en consecuencia existir *lightpaths* muy largos que incrementen excesivamente el retardo global de red.

2.2.3 Limitaciones de la topología virtual

De forma ideal se desea un grafo completamente conectado como topología virtual. Si una topología virtual está completamente conectada, el tráfico se puede enrutar de todos los nodos en un sólo salto lógico. Así la topología virtual aumentará la capacidad de procesamiento y reducirá los costes por hardware, también mejorará la red en términos de utilización del ancho de banda, ya que se reduciría el flujo de tráfico procedente de distintos pares de nodos para un mismo *lightpath*. A pesar de ello la conectividad completa para una red de N nodos es difícil de alcanzar por las siguientes razones:

1. Todos los nodos tienen que tener $N-1$ transmisores y receptores ópticos para originar y recibir $N-1$ *lightpaths*. Por lo tanto, se requieren $N \times (N-1)$ transmisores y receptores en total. Esto implica un enorme coste de componentes de red.
2. Un nodo requiere suficiente capacidad de procesado para manejar los mensajes provenientes de los $N-1$ *lightpaths* entrantes, lo que provoca un coste excesivo. Generalmente la capacidad de procesamiento se limita al número de puertos que el *switch* electrónico puede soportar.
3. Se requiere un elevadísimo número de longitudes de onda para establecer los $N \times (N-1)$ *lightpaths*. Si N es de orden 100, el número de longitudes de onda requeridas será del orden de 100 o 1000 en función del grado de conectividad de la red. No es económicamente rentable ni tecnológicamente posible usar miles de longitudes de onda.

2.2.4 Optimización de la topología virtual

En un problema de diseño de topología virtual se asume como dados la topología física y la matriz de tráfico medio entre los distintos pares origen-destino. Los flujos de tráfico atraviesan una secuencia de *lightpaths* hasta alcanzar su destino. Las métricas más importantes para optimizar una red óptica a



la hora de diseñar una topología virtual son: (1) número de saltos (virtuales) requeridos para enrutar el tráfico desde el nodo origen al nodo destino, (2) tráfico ofrecido a un *lightpath*, (3) flujo de tráfico de un solo salto, (4) retardo experimentado por los paquetes. Por lo tanto se definen cuatro funciones objetivo para los problemas de topologías virtuales.

1. Minimizar la media ponderada del número de saltos. La media ponderada del número de saltos (virtuales) se define como el número medio de *lightpaths* atravesados por una unidad de tráfico. Si $t^{s,d}$ es el tráfico ofrecido entre el nodo s y el nodo d y $h^{s,d}$ es el número de saltos entre s y d en la topología virtual, entonces el número ponderado de saltos en el par $\langle s,d \rangle$ viene dado por $h^{s,d} \times t^{s,d}$. Haciendo la media de este valor sobre todos los pares origen-destino se obtiene la media ponderada del número de saltos, que se debe de minimizar por dos razones fundamentales.

- Primera. Los paquetes experimentarán el mínimo número de conversiones opto-electrónicas en los nodos. Esto reducirá el retardo del tráfico globalmente. Además, permitirá utilizar de forma más eficiente los recursos disponibles: transmisores, receptores y puertos.
- Segunda. Cuanto menor es el número de *lightpaths* atravesado, menor es el tráfico ofrecido a un *lightpath* proveniente de distintos pares origen-destino. Consecuentemente el retardo de cola en los nodos se reduce, y la red será capaz de soportar futuras ampliaciones de tráfico.

2. Minimizar la congestión de la red. La congestión de la red se define como el máximo tráfico ofrecido a un *lightpath* (en cualquiera de ellos). El tráfico ofrecido sobre un *lightpath* se define como la suma de los flujos de tráfico entre todos los pares origen-destino que usan ese *lightpath*. La congestión de una red debe de reducirse lo más posible. En caso de que el tráfico se escale, debido a ampliaciones de la red, un nivel bajo de congestión permitiría enrutarlo sin modificaciones de los recursos. Al minimizar la congestión, de forma indirecta, también se reduce el número medio de saltos de los paquetes; por lo que estas dos funciones objetivo están muy relacionadas.



3. **Maximizar el flujo de tráfico de salto único.** El tráfico de salto único se refiere a aquel flujo que se transporta de nodo origen a destino en único salto virtual (es decir, por un solo *lightpath*). Para maximizar el flujo de salto único, se tiene que maximizar la utilización del ancho de banda para que se reduzca el tráfico compartido en un solo *lightpath* debido a distintos pares. Esta función objetivo tiene una relación estrecha con las dos descritas anteriormente.
4. **Minimizar el retardo de los mensajes.** Se llama retardo del mensaje al tiempo que necesita un mensaje para alcanzar el nodo destino desde que se transmite por el nodo origen. El retardo del mensaje se puede concebir con dos fuentes de retardo: el retardo de propagación y el retardo de cola en *lightpath*. El retardo de propagación sobre un *lightpath* es el tiempo que necesita el mensaje para atravesar el *lightpath* de un extremo a otro y depende de la longitud de la fibra. Es normal que en las redes de área extensa, el retardo de propagación domine sobre el tiempo de encolamiento, por lo que este último se considera despreciable. El retardo en cola llega ser significativo sólo cuando el *lightpath* está al límite de su capacidad por exceso de tráfico. En aplicaciones que requieren servicios en tiempo real el retardo de encolamiento influye más que el de propagación, por eso es necesario reducirlo al mínimo. Al minimizar el número medio de saltos o la congestión de la red, se minimiza de forma indirecta el retardo de encolamiento, ya que depende de la carga del *lightpath*.

2.2.5 Formulación del Problema

2.2.5.1 Notación específica

Además de las definiciones generales de la sección 2.2.2.1 se utilizan las siguientes variables y definiciones para abordar la formulación del problema en las cuatro posibles funciones objetivo.

1. **Matriz de saltos.** La matriz de saltos especifica el número máximo de saltos físicos en un *lightpath* entre los extremos de un par de nodos. La matriz de saltos H_p es una matriz bidimensional $N \times N$, donde cada entrada $h_{i,j}^p$ en H se corresponde con el par de nodos $\langle i, j \rangle$ cuyo valor representa el número de saltos permitidos para un paquete que atraviesa el *lightpath* del nodo i al nodo j .
2. **Indicador *Lightpath*-Longitud de Onda.** En caso de que exista un *lightpath* con la longitud de onda determinada el indicador *lightpath*-longitud de onda $p_{i,j,\omega}$ tendrá el valor 1. Esto quiere



decir que hay un *lightpath* del nodo i al nodo j y tiene asignada la longitud de onda ω . En otro caso el valor $p_{i,j,\omega}$ se fija a 0.

3. **Indicador *Lightpath*-Longitud de Onda-Enlace.** Este indicador se usa para constatar la existencia de un *lightpath* a un longitud de onda específica entre un par de nodos usando un enlace físico particular. El valor del indicador *Lightpath*-Longitud de Onda-Enlace se denota como $p_{i,j,\omega}^{x,y}$ y es 1 se hay un *lightpath* del nodo i al nodo j y usa la longitud de onda ω sobre un enlace físico del nodo x al nodo y . En otro caso el valor de $p_{i,j,\omega}^{x,y}$ es 0.
4. **Retardo de Propagación del Enlace.** El retardo de propagación sobre un enlace físico de un nodo x a un nodo y viene dado por la variable $d_{x,y}$. El mismo retardo de propagación pero sobre un *lightpath* se puede considerar como la agregación de los retardos implicados en cada enlace que atraviesa.
5. **Salto Lógicos.** El número de *lightpaths* que tiene que atravesar un paquete de tráfico del nodo origen i al nodo destino j sobre una topología virtual. Se denota como $h_{i,j}$.
6. **Flujo de *Lightpath*.** La cantidad total de tráfico ofrecido sobre un *lightpath* del nodo i al nodo j y se denota por $f_{i,j}$. El componente de tráfico debido al par de nodos $\langle s,d \rangle$ ofrecido sobre un *lightpath* se denota por $f_{i,j}^{(s,d)}$

2.2.5.2 Posibles funciones objetivo

A continuación se formulan las funciones objetivo presentadas en la sección 2.2.5.

1. **Minimizar el número medio de saltos.** Si t_{total} es todo el tráfico ofrecido a la red, el número medio de saltos (virtuales), denotado como h_{med} , vale:

$$h_{med} = \frac{1}{t_{total}} \times \sum_{s,d} h_{s,d} \times t^{s,d} \quad (2.2.1)$$

La función objetivo viene dada por

Minimizar h_{med}



2. **Minimizar la congestión de la red.** Se llama congestión de la red al flujo máximo en cualquier *lightpath* de la red. Se denota por f_{max} y vale:

$$f_{max} = \max_{i,j} f_{i,j} \quad (2.2.2)$$

La función objetivo viene dada por:

Minimizar f_{max}

3. **Maximizar el tráfico de salto único.** La cantidad de tráfico transportada en un solo salto (virtual) se denota por t_{unico} , y vale:

$$t_{unico} = \sum_{s,d} p_{s,d} \times t^{s,d} \quad (2.2.3)$$

La función objetivo viene dada por:

Maximizar t_{unico}

4. **Minimizar el retardo del mensaje.** El retardo que experimenta un mensaje d_{msg} se puede minimizar, minimizando el retardo de propagación (d_p) y el retardo de encolamiento (d_q).

El retardo de propagación d_p se puede calcular como sigue:

$$d_p = \sum_{i,j} \sum_{x,y} \sum_{\omega=0}^{W-1} f_{i,j} \times p_{i,j,\omega}^{x,y} \times d^{x,y} \quad (2.2.4)$$

Si C es la capacidad de un *lightpath*, el retardo de propagación d_q se puede calcular como sigue:

$$d_q = \sum_{i,j} \frac{f_{i,j}}{C - f_{i,j}} \quad (2.2.5)$$

La función objetivo se define como:

Minimizar $(d_p + d_q)$



2.2.5.3 Restricciones

A continuación se explican las restricciones a que está sometido el problema de optimización.

1. **Restricciones de grado virtual.** El número de transmisores y receptores disponibles en un nodo determina el grado de salida (D_{out}^v) y entrada (D_{in}^v), respectivamente, de ese nodo en la topología virtual. En las restricciones que se imponen se asume que todos los nodos tienen el mismo grado de salida y el mismo de grado de entrada. Las restricciones se pueden relajar especificando restricciones diferentes para cada nodo.

$$\sum_j p_{j,i} \leq D_{in}^v \quad \forall i \quad (2.2.6)$$

$$\sum_j p_{j,i} \leq D_{out}^v \quad \forall i \quad (2.2.7)$$

2. **Restricciones de flujo de tráfico.** Las restricciones de flujo de tráfico se refieren al tráfico enrutado sobre los *lightpaths* de la topología virtual. La restricción (2.2.8) describe el hecho de que el tráfico ofrecido a un *lightpath* es la suma de los tráficos ofrecidos a ese *lightpath* por todos los pares de nodos. La congestión de la red viene definida por la restricción (2.2.9). El componente de tráfico ofrecido a un *lightpath* debido a un par de nodos específico puede ser como máximo el flujo de tráfico entre ese par. El flujo de tráfico originado por un par de nodos origen-destino se puede bifurcar entre distintas rutas de la topología virtual. Este fenómeno se recoge en la restricción (2.2.10). La conservación del flujo de tráfico en los nodos finales del *lightpath* se define en la restricción (2.2.11). El componente de tráfico de un *lightpath* debido a un par de nodos concreto se añade al nodo origen del *lightpath* si es el mismo nodo que el origen del par. De igual forma, si el nodo final del par es el mismo que el nodo destino del *lightpath*, el tráfico se extrae del mismo. En otro caso el componente de tráfico se conmuta del *lightpath* de llegada a este *lightpath* en su nodo origen y después se vuelve a conmutar a un *lightpath* de salida en su nodo destino.

$$f_{i,j} = \sum_{s,d} f_{i,j}^{(s,d)} \quad \forall (i,j) \quad (2.2.8)$$

$$f_{i,j} \leq f_{max} \quad \forall (i,j) \quad (2.2.9)$$

$$f_{i,j}^{(s,d)} \leq p_{i,j} \times t^{s,d} \quad \forall (i,j)(s,d) \quad (2.2.10)$$



$$\sum_j f_{i,j}^{(s,d)} - \sum_j f_{j,i}^{(s,d)} = \left. \begin{cases} t^{(s,d)} & \text{si } s = i \\ -t^{(s,d)} & \text{si } d = i \\ 0 & \text{si } s \neq i \text{ y } d \neq i \end{cases} \right\} \forall (s,d), i \quad (2.2.11)$$

3. **Restricciones de longitud de onda.** Las restricciones sobre la longitud de onda se refieren a la asignación de longitudes de onda en los *lightpaths*. La restricción (2.2.12) se usa para asegurar que la longitud de onda usada por un *lightpath* es única. La restricción de continuidad de longitud de onda se expresa en (2.2.13). Como se ha venido diciendo repetidamente, esta restricción asegura que todos los enlaces atravesados por el mismo *lightpath* usen la misma longitud de onda. En la restricción (2.2.14) se expresa que cómo máximo un *lightpath* puede usar una longitud de onda sobre un enlace, o lo que es lo mismo, que dos *lightpaths* no pueden usar la misma longitud de onda en el mismo enlace físico. La conservación de longitudes de onda en los nodos finales de la topología física sobre un *lightpath* se expresa en la restricción (2.2.15). En el nodo origen de un *lightpath*, no existe ningún enlace físico de entrada con la longitud de onda asignada al *lightpath* igual que en el nodo destino de un *lightpath*, tampoco existe ningún enlace físico de salida con la longitud asignada al *lightpath*. En un nodo intermedio del *lightpath*, existe igual número de enlaces físico de entrada y salida con la longitud de onda asignada al *lightpath*.

$$p_{i,j} = \sum_{\omega=0}^{W-1} p_{i,j,\omega} \quad \forall (i, j) \quad (2.2.12)$$

$$p_{i,j,\omega}^{x,y} \leq p_{i,j,\omega} \quad \forall (i, j), (x, y), \omega \quad (2.2.13)$$

$$\sum_{i,j} p_{i,j,\omega}^{x,y} \leq 1 \quad \forall (x, y), \omega \quad (2.2.14)$$

$$\sum_{\omega=0}^{W-1} \sum_x p_{i,j,\omega}^{x,y} \times l^{x,y} - \sum_{\omega=0}^{W-1} \sum_x p_{i,j,\omega}^{y,x} \times l^{y,x} = \left. \begin{cases} p_{i,j} & \text{si } y = j \\ -p_{i,j} & \text{si } y = i \\ 0 & \text{si } y \neq i \text{ y } y \neq j \\ \forall (i, j), y \end{cases} \right\} \quad (2.2.15)$$



4. **Restricciones de salto.** Las siguientes restricciones aseguran que el número de enlaces físicos atravesados por un *lightpath* es como máximo el valor indicado en la matriz de saltos físicos H_p .

$$\sum_{x,y} p_{i,j,\omega}^{x,y} \leq h_{i,j}^p \quad \forall (i,j), \omega \quad (2.2.16)$$

5. **Restricciones de valor variable.** Las siguientes restricciones aseguran que el flujo de tráfico en un *lightpath*, el flujo de tráfico en un *lightpath* debido a un par de nodos y la congestión de la red son cantidades positivas.

$$f_{i,j} \geq 0 \quad \forall (i,j) \quad (2.2.17)$$

$$f_{i,j}^{(s,d)} \geq 0 \quad \forall (i,j), (s,d) \quad (2.2.18)$$

$$f_{max} \geq 0 \quad (2.2.19)$$

Además que hay asegurar valores binarios para: *lightpaths*, enlaces, *lightpath*-longitud de onda, *lightpath*-longitud de onda-enlace. Esto se garantiza mediante las siguientes restricciones.

$$p_{i,j} \in \{0,1\} \quad \forall (i,j) \quad (2.2.20)$$

$$l^{s,d} \in \{0,1\} \quad \forall (s,d) \quad (2.2.21)$$

$$p_{i,j,\omega} \in \{0,1\} \quad \forall (i,j), \omega \quad (2.2.22)$$

$$p_{i,j,\omega}^{s,d} \in \{0,1\} \quad \forall (i,j), (s,d), \omega \quad (2.2.23)$$

2.3 ALGORÍTMOS HEURÍSTICOS

El problema formulado en la sección 2.2.6 y varios de sus subproblemas son computacionalmente intratables, es decir NP-completo. Por eso para redes relativamente grandes no es práctico ni efectivo intentar resolver este problema de una forma exacta. Para obtener buenas aproximaciones se requieren algoritmos heurísticos. Durante toda esta sección se explicarán las aproximaciones



heurísticas al problema del diseño de la topología virtual proporcionando conceptos y definiciones que luego se aplicarán sobre algoritmos heurísticos concretos propuestos en la literatura.

2.3.1 Subproblemas

El problema completo de diseño de la topología virtual definido en la sección 2.2 se puede descomponer de forma aproximada en cuatro problemas parciales. Esta subdivisión es inexacta, pero simplifica la complejidad del problema original. La resolución marginal de los subproblemas de forma secuencial y la combinación de sus soluciones puede que no alcance la solución óptima para el problema completamente integrado. También puede ocurrir que alguno de los problemas parciales no tenga solución que sí tenía para un problema similar anterior, por lo tanto no es seguro encontrar soluciones a todos los problemas. Los cuatro subproblemas son

1. **Diseño de topología.** Determinar la topología virtual que se debe superponer sobre la topología física, determina la red de *lightpaths* en función de nodos origen y destino.
2. **Enrutamiento de *lightpaths*.** Determinar los enlaces físicos que cada *lightpath* atraviesa., esto es, la ruta de cada *lightpath* sobre la topología física.
3. **Asignación de longitudes de onda.** Determinar qué longitud de onda usa cada *lightpath*, es la asignación de longitudes de onda a cada *lightpath* establecido en la topología virtual, las restricciones de longitud de onda están condicionadas por cada enlace físico.
4. **Enrutamiento de tráfico.** Determinar de qué forma se enrutan los paquetes de tráfico entre los nodos origen y destino sobre la topología virtual obtenida.

Puede que el último subproblema, enrutamiento de tráfico sobre la topología virtual, no parezca importante en el diseño de la topología virtual. Efectivamente, una vez que se ha fijado la topología virtual mediante la resolución de los tres primeros subproblemas, el último problema marginal de enrutamiento de tráfico sobre una topología virtual es conocido por ser el mismo que se soluciona mediante varios algoritmos sobre cualquier topología. A pesar de ello, la determinación de los flujos de tráfico sobre la topología virtual y su optimización como métrica de red, hacen necesario la inclusión del cuarto subproblema como parte integral del problema general.



Como se ha comentado anteriormente la descomposición en cuatro subproblemas no es exacta. Tampoco es exacta la solución marginal para cada uno de ellos, porque también algunos de ellos son NP-completo. Por lo tanto la obtención de soluciones subóptimas para cada problema marginal se puede conseguir mediante algoritmos heurísticos. Este hecho puede conducir a la no obtención de una solución para el problema completo. Mediante la relajación de ciertas restricciones, se pueden obtener soluciones de los heurísticos, que serán evaluadas con límites de fiabilidad como se mostrará a continuación. Una de las restricciones más comúnmente relajadas es el número máximo de longitudes de onda que puede contener una fibra. En último término será el criterio del diseñador que verificará si la solución es viable.

La división de problemas presentada no es la mejor ni la única, pero es la que presenta mayor consenso en la literatura. Alternativas a esta descomposición se usan en parte de la bibliografía manejada. A pesar de ello, la consistencia y fiabilidad de los subproblemas, su relativa facilidad de solución y la coherencia de los resultados obtenidos, también en el problema completo, han hecho que nos inclinemos por esta opción.

2.3.2 Límites

Para evaluar una aproximación producida por un heurístico, necesitamos saber lo cerca que está nuestra solución de la solución óptima. Precisamente la razón de usar heurísticos para abordar un problema computacionalmente intratable hace que no podamos disponer de la solución óptima en primer lugar, y por consiguiente de cualquier comparación con ella. De ahí la necesidad de establecer unas fronteras que nos permitan evaluar las soluciones subóptimas obtenidas, basadas en consideraciones teóricas. Estos son los límites de fiabilidad comentados anteriormente y que vamos a desarrollar a continuación.

Un objetivo común en la resolución del problema de la topología virtual es la minimización de la congestión de la red, como se explicó en la sección 2.2. Un límite inferior de congestión obtenido por consideraciones teóricas nos permite saber que no todas las soluciones alcanzan valores pequeños de congestión, a la vez que nos ayuda a evaluar la solución proporcionada por algún heurístico. A continuación se explican varios casos de límite inferior de congestión.



2.3.2.1 Límite independiente de la topología física

Este límite parte del hecho de que la carga en todos los enlace lógicos de la red sería la misma, y de valor igual a la congestión, si el tráfico de total sobre la red fuera igualmente repartido entre todos los *lightpaths*. El valor de esta congestión actuaría entonces como límite inferior en cualquier topología virtual que pudiera ser diseñada bajo las condiciones de tráfico dadas. Este límite tiene en cuenta la demanda total de tráfico, pero no la distribución del tráfico total entre los diferentes pares origen-destino (que es el patrón de tráfico). Por lo tanto asume que el tráfico de un par aleatorio origen-destino se puede asignar a cualquier *lightpath* de la topología virtual, y consecuentemente, ignora la topología física.

Sea \bar{H} el número medio de saltos en la topología virtual. Si E_l denota el número de *lightpaths* de la topología virtual y r indica la tasa total de llegada de paquetes a la red, de forma fácil se ve que

$$\lambda_{max} \geq \frac{r \bar{H}}{E_l}$$

y estableciendo un límite inferior a \bar{H} también se establece límite a la congestión. Para que número medio de saltos sea mínimo, los pares origen-destino con la mayor cantidad de tráfico se deben conectar mediante un número pequeño de saltos. Como tiene que haber un máximo de $N\Delta_1$ pares conectados por un solo salto, asumimos que estos son exactamente los pare más cargados, y de forma similar para dos, tres y sucesivos número de saltos. En este caso el número medio de saltos atravesados por el tráfico de forma ponderada es un **límite inferior**. Este viene dado por la expresión

$$\bar{H} \geq \sum_k k S_k$$

Donde S_k es el número de fracciones de tráfico (con respecto al tráfico total de la red r) que son transportadas en k saltos. Estas fracciones de tráfico se pueden determinar de la siguiente forma: ordenar las fracciones de tráfico en orden descendente de magnitud y dividir las en bloques, donde el bloque i está compuesto por las $N\Delta_i$ fracciones sucesivas en esa lista. Según esto el primer bloque está formado por las primeras $N\Delta_1$ fracciones de tráfico, el segundo bloque por las siguientes $N\Delta_2$, y



así sucesivamente. Por lo tanto el sumatorio anterior S_k es la suma de las fracciones de tráfico que forma el bloque k .

2.3.2.2 Límite del árbol de mínimo flujo

Este límite se obtiene partiendo de consideraciones similares al anterior, pero con restricciones sobre los pares de nodos. En este caso, tenemos en cuenta la restricción de que cada nodo origen puede sólo ser fuente de Δ_i *lightpaths* en total, además de las consideraciones del apartado anterior. Este conjunto restrictivo de condiciones hace que el límite de mínimo flujo marque una frontera muy exigente. El número medio de saltos lógicos \bar{H} se limita asumiendo que cada nodo está conectado por un solo salto lógico a los Δ_i nodos con mayor carga de tráfico, por dos saltos a los Δ_i^2 nodos con siguiente mayor carga de tráfico, etc.

2.3.2.3 Límite iterativo

El límite iterativo se obtiene a partir de la formulación de programación lineal, resolviendo el problema mediante la relajación de ciertas restricciones. Para esto es necesario añadir una restricción adicional, que asegure que la congestión tiene que ser mayor que el límite inferior de congestión conocida *a priori*, este límite puede ser por ejemplo el límite de mínimo flujo del apartado anterior. Para mejorar el ajuste, el valor obtenido para la congestión por la resolución del problema relajado se puede usar ahora como nuevo límite de congestión conocido *a priori* y volver a resolver el problema de programación lineal para obtener un límite de congestión más ajustado. Este proceso iterativo se puede repetir para conseguir una progresiva precisión en el límite de la congestión.

2.3.2.4 Límite con independencia de la topología

Este límite se presenta como un método de obtención teniendo en cuenta la topología física de la red. Se derivan las sucesivas topologías para maximizar el tráfico de uno, dos y tres saltos; y estas topologías no influyen sobre el resto. Finalmente la congestión resultante se toma como el límite. Los autores⁸ de este límite indican que, en caso tráfico uniforme, su precisión es sólo algo mayor que el límite presentado en 2.3.2.2; pero en caso de tráfico no uniforme llega a ser mucho más estricto.

⁸ Banerjee et al., 1997



2.3.2.5 Límite inferior para el número de longitudes de onda

Durante el diseño de topologías virtuales es común hacerlo usando el mínimo número de longitudes de onda posible, y en la práctica existe un límite en cuanto al número de longitudes de onda que puede contener una fibra óptica. El límite se usa con frecuencia en la formulación del problema en la sección 2.2.6 pero no es tan común encontrarlo en las aproximaciones heurísticas. Un límite inferior sobre el número de longitudes de onda que se necesitan para un problema determinado es útil a la hora de evaluar la solución obtenida de un heurístico. Este límite fácil de computar ayuda a dar una respuesta rápida en cuanto a la viabilidad de una topología virtual, simplemente desechándola con una negativa. Según esta lógica se pueden definir dos límites. El primero parte del hecho de que el nodo con el menor número de enlaces (grado del nodo) δ_p puede ser origen de Δ_l *lightpaths*. Según esto el número de longitudes de onda requeridas obedece a $W \geq \lceil \delta_p / \Delta_l \rceil$. El segundo se deriva de que cada nodo es origen de un número de *lightpaths* igual a aquellos nodos que se pueden alcanzar con el menor número de saltos físicos.⁹

Se pueden encontrar distintos límites para el número de longitudes de onda requeridas asumiendo ciertas condiciones en la soluciones a los diferentes subproblemas. Por ejemplo, suponiendo que la topología virtual forma un hipercubo o calculando el número de longitudes de onda necesarias para escenarios con y sin *restricción de continuidad de longitud de onda*.

2.3.3 Aproximaciones heurísticas y técnicas

Durante el diseño de algoritmos heurísticos o soluciones aproximadas al problema de la topología virtual, los diversos autores ponen énfasis en diferentes aspectos. En la mayoría de casos se resuelven algunos de los problemas marginales estratégicos, pero no todos ellos. El punto de partida para muchos algoritmos heurísticos son las suposiciones que se adoptan sobre las características de la topología virtual a implementar. En las siguientes secciones se describen técnicas heurísticas clasificadas en tres apartados; en primer lugar, se asume que la topología virtual a implementar es una topología regular y conocida, como un hipercubo o una red de arrastre. En un segundo grupo, se asumen conocidos los *lightpaths* de la topología virtual en términos de nodos origen y destino para cada instancia del problema, y se abordan los subproblemas de enrutamiento de *lightpaths* y asignación de longitudes de onda. En la última categoría no se contemplan simplificaciones especiales.

⁹ El desarrollo de este límite se puede encontrar en [Ramaswami and Sivarajan, 1996]



El interés por usar topologías regulares en el contexto de las topologías virtuales para redes WAN se deriva de que algunas topologías virtuales extensas pueden condicionar la topología física, es decir se pueden añadir enlaces de fibra suplementarios a la topología física antes de implementar la topología virtual. Conforme se han desplegado más fibras para formar parte de la redes ópticas, usadas como enlaces punto a punto, el interés se ha ido desplazando hacia el aprovechamiento de estas mediante WDM y topologías virtuales, en lugar profundizar en el despliegue de medios físicos. También los estudios relacionados con las topologías arbitrarias han ido adquiriendo importancia en los últimos tiempos.

2.3.3.1 Topologías regulares

Las topologías regulares como los hipercubos o las redes de arrastre tienen diferentes ventajas como topologías virtuales. Su estudio está afianzado y todos sus aspectos son bien conocidos, por ello los resultados obtenidos son relativamente fáciles de derivar para la obtención de límites y medias. El enrutamiento del tráfico sobre las topologías regulares es también muy simple y sus resultados están disponibles en la literatura, por lo tanto el problema marginal de enrutamiento se convierte en trivial. Además, las topologías regulares cuentan de forma inherente con características de balanceo de carga.

Una vez que una topología regular se implementa como topología virtual, es necesario decidir la correspondencia entre los nodos de la topología física y los nodos de la topología regular (nos referiremos a este subproblema como *mapeo de nodos*) y cual será la secuencia de enlaces físicos que se asociarán a los enlaces de topología regular, es decir, los *lightpaths* (nos referiremos a este subproblema como *mapeo de ruta*). Este procedimiento también se llama embeber la topología regular en la topología virtual. Siguiendo la división en los cuatro subproblemas del diseño de heurísticos, la elección de topología regular junto al mapeo de nodos se corresponden con el diseño de la topología (primer subproblema), y el mapeo de rutas se corresponde con el enrutamiento de *lightpaths*.

Obviamente el número de nodos en la topología virtual (regular) no puede ser elegido con completa libertad, ya que obedece a las restricciones de la topología regular. En caso de que la topología física tenga menos nodos que la topología regular, se pueden añadir nodos ficticios antes de establecer la correspondencia entre ellos. Si el exceso de nodos se encuentra en la topología física, aquellos nodos con menos carga de tráfico se pueden combinar para establecer la correspondencia, a pesar de que se introduce una aproximación adicional que tendrá su reflejo en el resultado de la topología virtual. En



general, tanto el mapeo de nodos como el de rutas no toman en consideración el patrón de tráfico, y usan métricas como la distancia de la fibra y el reuso de longitudes de onda para enrutar los *lightpaths* sobre la topología física. Por esta razón en las topologías regulares existe una suposición tácita sobre el empleo de patrones uniformes o casi uniformes al usarlas como topologías virtuales. Los dos mapeos definidos tienen que estar libres de conflictos de longitudes de onda y deben de ceñirse a límites predefinidos en número de longitudes de onda por fibra. Todas estas limitaciones alcanzan rango de problemas parciales y está demostrado que son NP-completo, por lo que se necesitan algoritmos heurísticos para su resolución.

A continuación se enumeran algunas técnicas heurísticas disponibles en la literatura que abordan alguna instancia de las topologías regulares:

- La topología física se convierte primero en una cadena equivalente, preservando algunas de sus características, para posteriormente embeber la topología regular seleccionada en la cadena obtenida.¹⁰
- Se pueden comparar diversas topologías regulares para la misma topología virtual y determinar cual es más apropiada. La comparación se basa en el número de saltos lógicos entre nodos y el número necesario de longitudes de onda.¹¹
- Se desarrollan heurísticos para el mapeo de nodos en la topología regular (hipercubos) y la solución es tenida en cuenta en el mapeo de rutas y en la asignación de longitudes de onda. Por lo tanto se abordan el mapeo de nodos de la topología virtual, el enrutamiento de longitudes de onda y la asignación de longitudes de onda a los *lightpaths*. Para ello se especifican un primer algoritmo agresivo y un segundo más suave para obtener un primer mapeo y después refinarlo basándose en criterios de reducción del retardo global de la red.¹²
- Otro heurístico menos efectivo se presenta, con la diferencia de que problema del enrutamiento de tráfico se soluciona mediante un método desviación de flujo. El método de desviación de flujo es una buena alternativa a la solución óptima obtenida para el enrutamiento de tráfico mediante programación lineal. Este método arranca con una asignación inicial de flujo y de

¹⁰ Chlamtac et al., 1993

¹¹ Marsan et al., 1993

¹² Mukherjee et al., 1994



forma iterativa va desviando los flujos por distintas rutas alternativas, evitando aquellos enlaces con la mayor carga de tráfico.¹³

2.3.3.2 Topologías prediseñadas

Cuando se habla de topologías prediseñadas la atención se focaliza en el subproblema de enrutamiento de *lightpaths*, pero a la vez posibilita la resolución de la asignación de longitudes de onda y el enrutamiento del tráfico. En otras palabras, la topología virtual, tomada como una lista de *lightpaths* con sus nodos origen y destino, se resuelve para cada instancia del problema.

El patrón de tráfico en la red tendría que ser tenido en cuenta cuando se decide establecer los *lightpaths*. Por eso en algunas de las aproximaciones siguientes, donde los patrones de tráfico no se computan, sí se utiliza la información del tráfico en el enrutamiento de *lightpaths*. En los subproblemas de enrutamiento de *lightpaths* y asignación de longitudes de onda se pueden definir los objetivos de resolución en términos de *lightpaths*, por ejemplo la minimización del número de diferentes longitudes de onda necesarias. Como en el apartado anterior se citan algunas técnicas de las muchas disponibles en la literatura.

- Se asume conocido, aparte de los nodos origen y destino, el enrutamiento de los *lightpaths*. Se resuelve entonces el problema de asignación de longitudes de onda a los *lightpaths*, ahora lo llamamos Establecimiento Estático de *Lightpaths* (EEL). Está demostrado que el citado EEL es equivalente a un problema de grafo coloreado- n , y consecuentemente NP-completo. Este estudio aporta un algoritmo heurístico ambicioso para la asignación de longitudes de onda a un conjunto de *lightpaths*, con el objetivo de usar el menor número posible de estos.¹⁴
- Este estudio asume que el problema marginal de la topología virtual se ha resuelto, y el conjunto de *lightpaths* a establecer está disponible en términos de nodo origen-destino. El objetivo en la asignación de rutas y longitudes de onda es maximizar la utilización de longitudes de onda en los conmutadores. Esto se medirá en función de la utilización de los *Wavelength Routers* (WR) en cada nodo de la red. Para definir formalmente este problema, se introduce el término "*Latin Square*". Estos se pueden identificar como partes parciales de la solución de enrutamiento de *lightpaths* y asignación de longitudes de onda. Para solucionar

¹³ Mukherjee et al., 1996

¹⁴ Chlamtac et al., 1992



estos *Latin Squares* en los nodos individuales se proporcionan dos algoritmos heurísticos, y posteriormente mediante un esquema se combinan las soluciones parciales para resolver el problema de enrutamiento y asignación a nivel de red.¹⁵

- De forma similar al anterior, esta propuesta asume como conocida la topología virtual, en sus *lightpaths* definidos con nodos origen y destino para cada instancia del problema. El problema de enrutamiento de *lightpaths* concibe el tráfico de los *lightpaths* como un flujo multi-producto, y está demostrado que es NP-completo. Una aportación interesante de este estudio es la propuesta de reducción del tamaño del problema; para ello se aboga por un planteamiento a medida de la formulación (no usar la genérica), recortando la búsqueda en árbol para las rutas de los *lightpaths* y relajando las restricciones enteras. El estudio emplea métodos heurísticos conocidos, incluyendo redondeo aleatorio o coloreo de grafos, con unos buenos resultados contrastados y así abordar los problemas de enrutamiento de *lightpaths* y asignación de longitudes de onda.¹⁶

2.3.3.3 Topologías arbitrarias

Dentro de las técnicas y aproximaciones heurísticas, existen varios estudios en la literatura con métodos heurísticos para topologías virtuales arbitrarias. Estos estudios abordan bien el problema de la topología virtual en sí, o bien los problemas subsiguientes al diseño de topología virtual. La mayoría de estos métodos tienen en cuenta el patrón de tráfico de la red, por eso las topologías virtuales arbitrarias se usan en entornos con patrones de tráfico no uniforme o topologías físicas irregulares. Algunos de los heurísticos presentados presentan rasgos similares.

- El problema se define como el establecimiento de un grafo de conexiones ópticas sobre una red WAN, basada en la demanda media de tráfico, entonces se procede al enrutamiento por las conexiones del grafo según aquella demanda, por lo que disponemos de un conjunto de circuitos virtuales dinámicos que contienen todos los *lightpaths*. Por consiguiente este heurístico, que se reduce a la solución de las conexiones del grafo, se corresponde con los tres primeros subproblemas del diseño de la topología virtual presentados en la sección 2.3.1. El problema se formula como un problema de programación no-lineal y se presenta una separación de instancias aproximada. El heurístico que se propone se basa en una

¹⁵ Chen and Banerjee, 1995

¹⁶ Banerjee and Mukherjee, 1996



aproximación ambiciosa, es utilizar el número de longitudes de onda disponibles para el máximo número de *lightpaths*.¹⁷

- Adoptando una de las técnicas heurísticas más recurrentes, este estudio aboga por crear *lightpaths* entre nodos en orden decreciente a la demanda de tráfico entre ellos. Se establecen primero los *lightpaths* que mayor volumen de tráfico cursan entre ellos. Si se ha computado todo el tráfico pero algunos nodos no han agotado sus recursos de grado (transmisores y receptores), el resto de *lightpaths* se establecen aleatoriamente siguiendo las restricciones. Una versión modificada de este algoritmo establece en primer lugar un par de *lightpaths* en direcciones opuestas, para cada enlace físico, después se sigue el procedimiento descrito por el algoritmo original. Este método asegura que el tráfico se puede enrutar siempre en el menor número de saltos físicos y por tanto es eficiente a lo hora de satisfacer requerimientos de retardo. Esta aproximación heurística centrará los algoritmos HLDA y MLDA presentados en el capítulo siguiente.¹⁸
- Otro heurístico depende del límite iterativo, descrito en la sección 2.3.2.3, que se tiene en cuenta para relajar las restricciones de la formulación de programación lineal y definir los indicadores de los *lightpaths*.¹⁹
- Otros heurísticos no tienen en cuenta el patrón de tráfico en absoluto, pero se concentran en crear *lightpaths* que usan sólo unos pocos enlaces físicos, ya que de esta forma se conservan las longitudes de onda.²⁰
- En algunos estudios heurísticos se intenta maximizar el tráfico de salto único pero también existe el heurístico que busca el objetivo opuesto. El algoritmo trata de maximizar el tráfico multisalto, ya que concentrarse en el salto único puede llevar a congestión. Los resultados de ambas aproximaciones se presentan con un rendimiento muy similar, a pesar de que los detalles sobre la asignación de longitudes de onda se omiten.²¹

¹⁷ Zhang and Acampora, 1995

¹⁸ Ramaswami and Sivarajan, 1996

¹⁹ Ramaswami and Sivarajan, 1996

²⁰ Ramaswami and Sivarajan, 1996

²¹ Banerjee and Mukherjee, 1997



- Otros estudios propuestos posteriormente siguiendo la argumentación del apartado anterior, proponen maximizar el tráfico de salto único, pero concentrándose en los mayores volúmenes de tráfico, aunque también se indica que los componentes de tráfico más pequeños pueden enrutarse de forma arbitraria y producir congestión en algún enlace físico. Se diseña un esquema que describe el mapeo de la red a un grafo bipartito para evitar la descompensación de carga en los enlaces. Por último se propone un algoritmo heurístico conocido para la asignación de longitudes de onda.²²

2.4 ENFOQUES RELACIONADOS CON EL DISEÑO DE LA TOPOLOGÍA VIRTUAL

En este apartado se van a introducir algunas técnicas y algoritmos que se apartan de la definición formal del problema definido hasta ahora, pero que están muy relacionados con el diseño de topologías virtuales para las redes *wavelength-routed*.

2.4.1 Beneficios incrementales

Diversos estudios en el campo de las redes ópticas ponen de manifiesto los beneficios introducidos en la resolución del problema RWA mediante topologías virtuales sobre redes WANs. Se extrajo un patrón de tráfico real de una red *backbone* T1 NSFNET en enero de 1992. Sobre los datos obtenidos se aplican tres esquemas diferentes para escalar el patrón de tráfico. El primer esquema sólo establece un punto de partida respecto a la eficiencia de enrutamiento y los otros dos usan WDM y topología virtual con WDM respectivamente. El índice más representativo fue el factor de escala, que si en los dos primeros esquemas se situó en 49 y 57 respectivamente, en el tercero fue 106. También son significativos los datos relativos a la utilización de los enlaces, cuyo valor mínimo es 32 % y 23 % en los dos primeros esquemas, ascendiendo hasta el 71 % en el tercer esquema (la máxima carga en los enlaces es 99 % para los tres esquemas). Con la ayuda de los citados índices se demuestra fácilmente los beneficios que acarrea para el rendimiento de la red el uso de topologías virtuales, así como su escalabilidad.

²² Banerjee et al., 1997



2.4.2 Límites de la conversión de longitud de onda

Como se dijo en el Capítulo I tiene que existir un equilibrio, basado en criterios de coste, entre dotar una red de capacidad de conversión de longitud de onda total o limitarla, instalando sólo *Wavelength Routers* (WR) en aquellos nodos que se consideren estratégicos. La clave del estudio de esta dicotomía está en el subproblema de la asignación de longitudes de onda. En los estudios realizados los resultados se obtienen en función de la topología física, como por ejemplo una red en forma de anillo.

Al ser la longitud de onda el factor determinante, los algoritmos que trabajan sobre la ubicación de conversores en un escenario de conversión limitada, lo hacen segmentando una ruta completa entre dos nodos origen-destino (un *lightpath*) de forma que cada segmento pueda tener un longitud de onda continua. Por ejemplo, si un *lightpath* de 9 saltos presenta tres segmentos de continuidad de longitud de onda (3-3-3) serán necesarios tres conversores de longitud de onda en los nodos indicados. Por lo tanto la segmentación de la ruta será el objeto de estudio para los algoritmos, que con programación lineal o aproximaciones heurísticas, intentan resolverlo.

Supongamos una ruta de N nodos con K conversores (K segmentos). Este problema es NP-completo y se resuelve atendiendo a diversas métricas como función objetivo. Suponemos una ruta entre un nodo origen y destino que atraviesa diversos enlaces, cada uno con sus longitudes de onda. Teóricamente la colocación de conversores en esta ruta se puede alcanzar de forma óptima ya que la probabilidad de éxito en una longitud de onda de cada enlace es la misma. Sea ρ_i la carga por longitud de onda en el enlace i . Las cargas de los diferentes enlaces no tienen por que ser iguales.

La probabilidad de éxito para una longitud de onda en un segmento S_j que comprende del enlace j al enlace k , es decir, la probabilidad de que esa longitud de onda esté libre en todos los enlaces del segmento, viene dada por:

$$S_j = \prod_{x=j}^k \overline{\rho_x}$$

Dada una ruta con H saltos y K número de conversores (nodos con conversión), no siempre será posible dividir la ruta en $K+1$ segmentos con un probabilidad de éxito igual. Por esta razón, los heurísticos se pueden usar para formar segmentos con probabilidades éxito aproximadamente iguales.



Se etiquetan los enlaces de 0 a $H-1$. Los segmentos se etiquetan de 0 a K y sus probabilidades de éxito se denotan de S_0 a S_k , respectivamente. Sea M la media geométrica de la probabilidad de éxito en los $K+1$ segmentos. El valor de M se puede calcular con la siguiente ecuación:

$$M^{k+1} = \prod_{i=0}^K S_i = \prod_{x=0}^{H-1} \rho_x$$

2.4.2.1 Algoritmo izquierda a derecha

Este algoritmo examina los enlaces de izquierda a derecha. Sea $j-1$ el último enlace del segmento más alejado. El nuevo segmento incluye los enlaces desde j a k , siendo k el índice más pequeño donde la probabilidad de éxito del segmento se encuentra más próxima a M , será entonces el segmento desde j a $k+1$. El inconveniente de este algoritmo es la acumulación de errores de aproximación en los sucesivos segmentos. Consecuentemente, la probabilidad de éxito del último segmento puede tener una variación excesiva respecto al valor deseado M .

2.4.2.4 Algoritmo izquierda a derecha con recomputado de la media

Como en el caso anterior, también los segmentos en este algoritmo se examinan de izquierda a derecha. Pero ahora en la formación de un nuevo segmento, este heurístico recalcula la media geométrica M sobre las probabilidades de éxito de los segmentos remanentes.

2.4.2.5 Algoritmo izquierda o derecha

A diferencia de los dos anteriores, este algoritmo forma los segmentos desde ambos extremos, izquierdo y derecho. Primeramente determina los segmentos probables por cada flanco, y una vez calculados, se escoge aquel que presente una probabilidad de éxito más cercana a M . Este heurístico intenta obtener la mejor selección para que la probabilidad de éxito sea lo más cercana posible a la ideal. Es previsible que este algoritmo presente un error de aproximación menor que en los dos casos anteriores.



2.4.3 *Traffic Grooming*

Los *lightpaths* proporcionan un ancho de banda grande, que no suele ser aprovechado por los usuarios individuales, por eso deben de ser vistos como canales de transporte en la red *backbone*, donde se multiplexa el tráfico de diversas aplicaciones de la red de acceso. En cierto sentido, esta es la razón que justifica la inclusión del problema de enrutamiento de tráfico en el diseño de la topología virtual, ya que el tráfico proveniente de aplicaciones individuales debe de ser enrutado por la topología virtual propuesta, y de esta forma los *lightpaths* transportan el tráfico obtenido por agregación de los flujos de baja de velocidad. El patrón de multiplexación de tráfico sobre los *lightpaths* afecta directamente a la eficiencia del reenvío óptico que se produce en los *Wavelength Routers* (WR), ya que toda la información que contiene un *lightpath* se somete a una conversión electro-óptica y a un posterior enrutamiento electrónico en el caso de que el flujo de tráfico finalice en el nodo intermedio. Este efecto se refleja en el coste de los componentes de red (número y capacidades). Para poder computar los efectos del diseño, se especifican diversas topologías físicas, cuyos resultados se comparan tomando como variables relativas al coste, el número de transceptores por nodo, número de longitudes de onda, número medio de salto para los *lightpaths*, o la caracterización de los patrones de tráfico. Por todo ello el *traffic grooming* se están convirtiendo en una de las áreas dentro del diseño óptico más estudiadas en las líneas de investigación actuales.

2.4.4 *Lighpath generalizados*

El concepto de generalizar *lightpaths* está relacionado con los *árboles de luz*, que se definen semejantes a los *lightpaths* como canales transparentes implementados con una sola longitud de onda y nodo origen, pero con varios nodos destino. Por lo tanto son canales de luz punto-multipunto. En los estudios realizados sobre esta técnica, se pone de manifiesto que los árboles de luz son más óptimos para un escenario dado, dada su enfoque genérico y sus posibilidades para envíos ópticos múltiples.

Como se ha venido definiendo en todo el problema RWA, las soluciones óptimas no son prácticas desde el punto de vista computacional, por lo que en enfoques más genéricos este rasgo se agudiza. Serán necesarias por lo tanto soluciones heurísticas como en los casos anteriores, pero deberán adaptarse a los nuevos espacios de búsquedas, mayores que en los problemas más acotados. Sometidos los problemas punto a punto a las aproximaciones de árboles de luz, se obtienen mejores compromisos entre el ancho de banda, el retardo, la congestión, el número de saltos en la topología física; que en la aproximación de los *lightpaths*. El reto para los árboles de luz será obtener algoritmos



heurísticos que puedan afrontar el incremento de la complejidad, y producir soluciones viables y que demuestren que se puede alcanzar el compromiso entre todas las variables mencionadas.

2.4.5 Reconfiguración

El problema de la reconfiguración de la red, es decir la variación de la topología virtual en un escenario variable, está muy relacionado con el diseño de topologías virtuales. Para su estudio se aportan dos aproximaciones fundamentales.

2.4.5.1 Coste

En esta aproximación se parte de la premisa que tanto la topología virtual actual como la nueva que requiere la red son conocidas, junto con los detalles de topología física. La clave está en minimizar el coste de esa reconfiguración, de ese cambio de topologías virtuales. El coste se puede expresar en términos del número de *Wavelength Routers* necesarios en el nuevo escenario, o el número total de conversiones ópticas que se necesitan cambiar para implementar los nuevos *lightpaths* y eliminar los antiguos. Estas métricas son importantes en la medida que reflejan el tiempo que necesitan *offline* para hacer los cambios, así como el esfuerzo necesario para reconfiguración de la red. Métricas similares a estas se pueden aplicar igualmente, por ejemplo en el caso de que la red no se pueda desconectar por completo, pero una sección intermedia de la topología virtual tenga que ser reconfigurada (grupo de *routers*) rápidamente y posteriormente puesta de nuevo en funcionamiento. Estas métricas más complejas de igual forma reflejan el tiempo total necesario en la reconfiguración así como el esfuerzo requerido para el desarrollo de los cambios en las topologías intermedias.

En la literatura actual no hay estudios que aborden el problema de la reconfiguración en redes *wavelength-routed* WAN, a pesar de haber encontrado algunas aproximaciones para topologías virtuales en redes *broadcast* LAN. Es interesante destacar que las técnicas usadas en estos estudios se pueden extrapolar fácilmente a nuestras redes, nos referimos específicamente a las técnicas de *intercambio de enlaces* e *intercambio de ramas* usadas para minimizar el coste de conversión de una topología virtual a otra.



2.4.5.2 Optimización

Otra aproximación es asumir como dada solamente la topología virtual actual, junto con el patrón de tráfico cambiante y/o la topología física que hace necesaria la reconfiguración. Con esta información de partida el algoritmo de reconfiguración propuesto, implica la solución de la nueva topología virtual por sí misma sin referencia alguna a la topología virtual actual para obtener una nueva solución óptima, con un nuevo valor óptimo para la función objetivo que fue propuesta, valor que es registrado. Entonces el problema del diseño de la topología virtual se reformula con una restricción adicional que fuerza la función objetivo anterior al valor registrado, y la nueva función objetivo implica la minimización del número de *lightpaths* que deben de ser añadidos o retirados.

Mientras que este método garantiza encontrar una topología virtual óptima para las nuevas condiciones, no alcanza un equilibrio entre la búsqueda de un nueva topología virtual óptima y otra que implique el menor número de cambios como sea posible sobre la vieja topología. Según esto es posible que se lleve a cabo una reconfiguración muy costosa de la red para una ligera mejora en su rendimiento. Es posible encontrar para este problema formulaciones más equilibradas, y heurísticos asociados que probablemente funcionen mejor en la práctica.



CAPÍTULO 3

Algoritmos heurísticos

3.1 DESARROLLO DE LOS ALGORÍTMOS

3.1.1 HLDA (*Heuristic Logical Design Algorithm*)

El algoritmo heurístico HLDA tiene por objetivo minimizar la congestión en la red dada. Como variables de entrada se tienen la matriz de tráfico y la topología física. También se asumen conocidos el número de transmisores y receptores por nodo, y el número de longitudes de onda por fibra, que es fijo. Este algoritmo tratará de maximizar el flujo de tráfico por salto único, medio por el que se alcanzará la minimización de la congestión. El algoritmo se describe a continuación.

Paso 1: Se hace una copia de la matriz de tráfico $T = (t^{s,d})$.

$$Q = (q^{s,d}) = T$$

Paso 2: Se selecciona el par de nodos origen-destino (s_{max}, d_{max}) con mayor volumen de tráfico entre ellos.

$$q^{s_{max}, d_{max}} = \max_{s,d} q^{s,d}$$

Si ya se han tenido en cuenta todos los pares con tráfico distinto de cero, ir al paso 4.

Paso 3: Si el nodo s_{max} tiene menos enlaces de salida que Δ_1 número de transmisores, y el nodo d_{max} tiene menos enlaces de entrada que Δ_1 número de receptores; entonces se encuentra la longitud de onda de índice más bajo por la ruta con menor número de saltos en el grafo G_p (topología física). Si hay más de una ruta empatada en el menor número de saltos, se toman secuencialmente.

If (longitud de onda disponible en la ruta seleccionada)

Se crea un *lightpath* entre s_{max}, d_{max}

Se busca el siguiente par s', d' con el volumen de tráfico mayor

$$q^{s',d'} = \max_{s \neq s_{max}, d \neq d_{max}} q^{s,d}$$



$$q^{s_{max,d_{max}}} = q^{s_{max,d_{max}}} - q^{s',d'}$$

Ir al paso 2.

else $q^{s_{max,d_{max}}} = 0$; ir al paso 2.

Paso 4: Si todavía no tenemos N_{Δ_1} *lightpaths*, se colocan los enlaces lógicos remanentes de forma aleatoria mientras que las restricciones de grado no se violen y se encuentre una longitud de onda libre para la ruta más corta.

Notar que HLDA coloca más de un *lightpath* entre nodos con mucho volumen de tráfico, en el caso de que el tráfico "residual" (ver paso 3) entre dos nodos sea mayor que el tráfico entre el siguiente par de nodos con mayor volumen de tráfico.

En resumen, la asignación de un *lightpath* en la topología virtual, una vez identificados el par de nodos origen-destino sobre el que se hará la tentativa, requiere cuatro condiciones, a saber:

1. Existencia de un transmisor libre en el nodo origen.
2. Existencia de un receptor libre en el nodo destino.
3. Existencia de una ruta física entre los nodos.
4. Existencia de una longitud de onda libre y común a todos los enlaces físicos que el *lightpath* atraviesa.

El HLDA resuelve los tres primeros subproblemas (topología virtual, enrutamiento de *lightpaths*, asignación de longitudes de onda) aunque no lo hace de forma independiente. Mediante el siguiente ejemplo se describe su funcionamiento.

3.1.2 MLDA (*Minimum-Delay Logical Design Algorithm*)

El objetivo del MLDA, a diferencia del anterior, es hacer mínimo el retardo de propagación, sin embargo hereda del HLDA su funcionamiento en la parte fundamental. Solamente se define en el caso que el número de transmisores y receptores por nodo Δ_i , sea mayor que el grado de la topología física (enlaces conectados a los nodos) y asegura para cada par de nodos que existe el camino más corto, concepto que se relaciona con el retardo de propagación.

El algoritmo comienza haciendo una copia de la topología física a la topología virtual, por lo que ésta incluye a aquélla. Se establecen *lightpaths* según los enlaces físicos, serán por tanto de un solo salto. El resto de los *lightpaths* se establecen según el algoritmo HLDA, tratados los



pares de nodos en orden decreciente de tráfico. Por esta razón las topologías lógicas creadas por el MLDA son capaces de enrutar todos los paquetes por el camino más corto (*lightpaths* de un solo salto) entre todos los pares de nodos. Esta característica lo convierte en el algoritmo ideal para satisfacer exigentes restricciones de retardo. Como desventaja se puede apuntar, que mientras minimiza el retardo medio de los mensajes, presenta unos resultados más pobres en términos de congestión.

3.1.3 SHLDA (*Shortest-Hop Heuristic Logical Design Algorithm*)

Algunos de los métodos presentados –HLDA y MLDA- están centrados en maximizar el tráfico de salto único, métodos que no son adecuados para el diseño de una topología lógica que tenga que soportar tráfico IP. Para suplir esta carencia, se introduce un nuevo algoritmo heurístico llamado SHLDA (*Shortest-Hop Logical Topology Design Algorithm*) que evita los citados problemas.

Asumimos que una función de enrutamiento actúa sólo a nivel de la capa de red IP. Según esto, la topología virtual debe de diseñarse, incorporando la selección de rutas, que aporta el protocolo de enrutamiento IP. Como es natural, el protocolo de enrutamiento IP, seleccionará la ruta más corta para el reenvío de paquetes. Al hablar de una ruta corta, nos referimos a que el número de *lightpaths* entre los nodos finales del par, es pequeño. De hecho, los retardos de propagación y de encolamiento también influyen en la selección de la ruta. Por ello, el contador de saltos del *lightpath* (i.e. número de *lightpaths* que atraviesa un paquete) debe de reducirse todo lo posible, objetivo este del algoritmo que nos ocupa. Una vez que se considera que un *lightpath* se puede dividir para soportar tráfico debido a distintos pares origen-destino, es necesario un conjunto de *lightpaths* para alcanzar el destino y se debe de tener en cuenta, en cada salto, el retardo de procesamiento introducido por los *routers* electrónicos. En este método se incorpora esta consideración en la configuración final de los *lightpaths*.

SHLDA usa la métrica F_{ij} para el par de nodos ij , que viene dada por la siguiente ecuación,

$$F_{ij} = \gamma_{ij} \times h_{ij}$$

donde γ_{ij} es la demanda de tráfico del nodo i al nodo j , y h_{ij} es el cuenta de saltos de la ruta más corta entre el par de nodos ij de la topología física. Aquí, la cuenta de saltos del *lightpath* se



refiere al número de enlaces físicos que atraviesa el *lightpath*. SHLDA primero usa el contador de saltos como métrica para el cálculo del orden de configuración del *lightpath*. El retardo de propagación y el contador de saltos se tienen en cuenta a la hora de determinar la ruta del *lightpath*. Para la determinación de la ruta del *lightpath* desde el nodo i al nodo j , usamos una métrica R_{ij} , que viene dada por la siguiente ecuación,

$$R_{ij} = D_{ij} \times h_{ij},$$

donde D_{ij} es el retardo de propagación total de la ruta desde el nodo i al nodo j . SHLDA selecciona la ruta que presenta un valor de métrica R_{ij} menor, como la ruta más corta entre el par de nodos ij . Eso nos permite establecer los *lightpaths* minimizando el número de *routers* electrónicos utilizados. La descripción en detalle del algoritmo se presenta a continuación.

- **Paso 1:** Calcular la métrica F_{ij} para cada par de nodos ij de la matriz de tráfico $T=t_{ij}$. En el cálculo de F_{ij} , h_{ij} se toma de forma trivial como el número de saltos de la ruta física más corta.
- **Paso 2:** Colocar un *lightpath* entre los nodos si hay recursos disponibles.
- **Paso 3:** Seleccionar el par de nodos i',j' donde i' y j' son los índices obtenidos de $\max_{i',j'} F_{i'j'}$. Si $F_{i',j'}=0$, ir al paso 5. En otro caso, continua al paso siguiente.
- **Paso 4:** Encontrar la ruta más corta entre el par de nodos i',j' , y comprobar la disponibilidad de longitudes de onda para configurar el *lightpath*. Si hubiera varias disponibles, usar la longitud de onda con el menor índice. Fijar ahora $F_{i',j'}=0$ y volver al paso 3.
- **Paso 5:** Si todavía existen longitudes de onda sin utilizar, los *lightpaths* se configuran aleatoriamente tanto como sea posible.

3.1.4 LEMA (*Link Elimination via Matching Algorithm*)

Los algoritmos como HLDA y TILDA intentan maximizar el tráfico de salto único, consideran un subconjunto de pares para el diseño de la topología virtual, los de volumen de tráfico mayor. Como consecuencia, el tráfico de los nodos no tenidos en cuenta tiene que enrutar por otros *lightpaths*. Esto incluso puede llegar a causar congestión en algunos *lightpaths*, precisamente cuando la congestión es la función objetivo para minimizar en estos algoritmos.



Para evitar esta situación, se propone tener en cuenta todos los pares de nodos, independientemente de su tráfico, durante el diseño de la topología virtual. Esto es lo que hace el algoritmo heurístico *Link Elimination Via Matching Algorithm* (LEMA), que parte una topología virtual completamente conexa y elimina enlaces de forma iterativa basándose en los pesos de tráfico de los enlaces mediante un grafo bipartito. El algoritmo se describe a continuación.

LEMA toma como variables conocidas la topología física, la matriz de tráfico T , y el número de transmisores y receptores de los nodos. Como se ha comentado el elemento $t^{s,d}$ de la matriz de tráfico representa el flujo de tráfico medio entre los correspondientes nodos s y d . De inicio, se crea un grafo bipartito completo, donde cada partición contiene los N nodos de la red. Los enlaces se crean desde todos los nodos de la partición 1, a todos los nodos de la partición 2. Al enlace del nodo i de la partición 1 al nodo j de la partición 2, se le asigna un peso igual al flujo de tráfico $t_{i,j}$. La topología virtual correspondiente a este grafo está completamente conectada (conexa) donde el grado lógico de cada nodo es $N-1$, siendo N el número de nodos en la topología física. Los *lightpaths* se transportan al medio físico tomando la ruta más corta (menor número de saltos). Si se satisfacen las restricciones de grado, es decir hay suficientes transmisores y receptores para establecer todos los *lightpaths*, el algoritmo se da por concluido; en otro caso la topología virtual se mejora iterando.

Ahora, del grafo bipartito, se seleccionan los N enlaces menos congestionados mediante el algoritmo *via minimum perfect matching*, y se eliminan si es posible. Un enlace marcado como congestionado se elimina siempre y cuando no desconecte el grafo de la topología virtual. El tráfico que se enrutaba por estos enlaces se vuelve a enrutar por los enlaces remanentes, seleccionando aquellos rutas con menor congestión. En cada iteración se vuelve a comprobar si se cumplen las restricciones de transmisores y receptores, cuando así sea, el algoritmo termina; en caso contrario se repite el procedimiento.

La segunda parte del algoritmo, que aquí se omite, tiene por objeto embeber la topología virtual obtenida, y que ya asegura que no viola las restricciones de grado, en la topología física. En esta etapa, se dará resolución a los otros dos problemas marginales, ruta física y longitud de onda para cada *lightpath*.



3.1.5 TILDA (*Traffic Independent Logical Design Algorithm*)

Hasta aquí se han presentado algoritmos que tomaban la matriz de tráfico como dato conocido y a partir de ahí elaboraban un procedimiento, para tomar decisiones en función de la demanda o la congestión. También consideramos otros algoritmos como el TILDA, que no tienen en cuenta la distribución de tráfico, a la hora de calcular la topología virtual, que puede ser necesario en escenarios donde se desconozca *a priori*. Como es natural, la complejidad de este algoritmo es mucho menor respecto a los tres anteriores, ya que no cuenta con variables de cómputo numéricas, tan sólo con la información aportada por la topología física. Su funcionamiento se describe a continuación.

Establece en primer lugar *lightpaths* entre todos los nodos vecinos de un solo salto en la topología física, después hace lo mismo con vecinos separados dos saltos (comprobando que no hay ya un enlace lógico entre ellos), con vecinos separados tres, cuatro, etc. siempre comprobando que no haya *lightpaths* previos entre el mismo par y que no se violan las restricciones de grado (número de transmisores y receptores). Ya que los *lightpaths* utilizan el menor número posible de enlaces físicos, TILDA minimiza el número de longitudes de onda requeridas para la topología virtual, y puede ser un algoritmo apropiado para redes con patrones de tráfico desconocidos o uniformes.

3.1.6 RLDA (*Random Logical Design Algorithm*)

El RLDA no se trata de un algoritmo en rigor, ya que el procedimiento para establecer los *lightpaths* no obedece a ningún criterio de optimización, pero resulta muy útil para funciones de comparación del rendimiento de la red respecto a los hasta ahora estudiados. El RLDA establece los *lightpaths* se forma absolutamente aleatoria, sólo sujeto a encontrar *lightpaths* en cada enlace y sin violar las restricciones de grado, pero ignorando la matriz de tráfico como en el caso anterior.



3.2 EJEMPLOS TEÓRICOS

3.2.1 HLDA

Considerar la topología física de la Fig. 21(a) y la matriz de tráfico de la Fig. 19. Se asume que todos los nodos tienen dos transmisores y dos receptores; y las fibras contienen dos longitudes de onda ω_0 y ω_1 . La Fig. 19 describe la evolución de la matriz de tráfico en cada iteración del algoritmo. Las iteraciones se han etiquetado con letras desde a hasta i. El par de nodos con mayor tráfico en cada iteración se ha puesto entre paréntesis. En cada iteración se resta al par de mayor tráfico el siguiente en volumen de tráfico, en caso de que se pueda establecer el *lightpath*. Las primeras iteraciones, que se ocupan de los pares de nodos con mayor tráfico suelen establecer los *lightpaths* sin problema, ya que todos los recursos están disponibles (transmisores, receptores y longitudes de onda). Según se agotan éstos, se hace más difícil la concurrencia de las cuatro condiciones que hemos indicado como imprescindibles para el establecimiento de un circuito virtual.

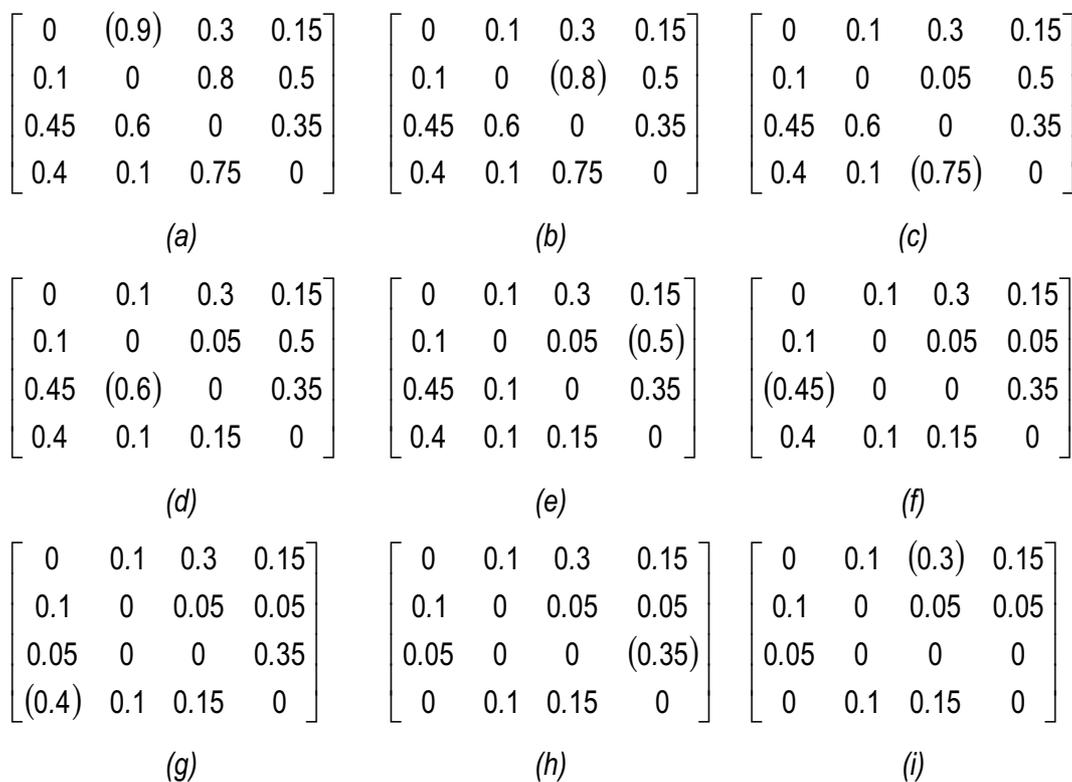


Figura 19. Evolución de la matriz de tráfico en el HLDA



En la tabla de la Fig. 20 se puede ver la evolución del algoritmo: iteración, par cargado, ruta física, longitud de onda usada por el *lightpath* y transmisores y receptores remanentes en los nodos origen y destino respectivamente. A partir de la iteración *f* los *lightpaths* comienzan a no establecerse por agotamiento de recursos. En la iteración *g* el enlace 4→3, parte de la ruta del *lightpath*, ha agotado las dos longitudes disponibles, por lo que bloquea la petición. En las dos últimas iteraciones el nodo 3 tiene agotados transmisores y receptores respectivamente.

Iteración	Par	Ruta	ω	Tx(origen)	Rx(destino)
<i>a</i>	1→2	1→2	ω_0	1	1
<i>b</i>	2→3	2→4→3	ω_0	1	1
<i>c</i>	4→3	4→3	ω_1	1	0
<i>d</i>	3→2	3→4→2	ω_0	1	0
<i>e</i>	2→4	2→4	ω_0	0	1
<i>f</i>	3→1	3→1	ω_0	0	1
<i>g</i>	4→1	4→3→1	No hay ω libre en el enlace 4→3		
<i>h</i>	3→4	3→4	No hay tx libres en el nodo 3		
<i>i</i>	1→3	1→3	No hay rx libres en el nodo 3		

Figura 20. Tabla de iteraciones. Asignación de *lightpaths* para el ejemplo del HLDA

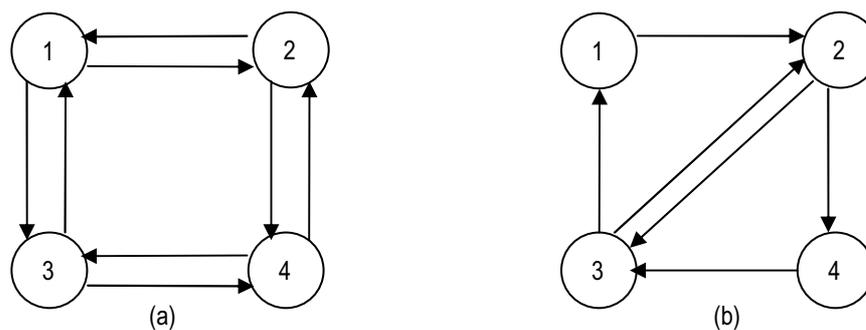


Figura 21. Ejemplo teórico HLDA. (a) Topología física. (b) Topología virtual

3.2.2 MLDA

Consideramos la topología física de la Fig.24(a) y la matriz de tráfico de la Fig. 22.. Se asume que cada nodo tiene tres transmisores y tres receptores y las fibras tienen dos longitudes de onda ω_0 y ω_1 . En primer lugar MLDA crea un *lightpath* por cada enlace físico y después añade enlace basándose en el funcionamiento del HLDA.

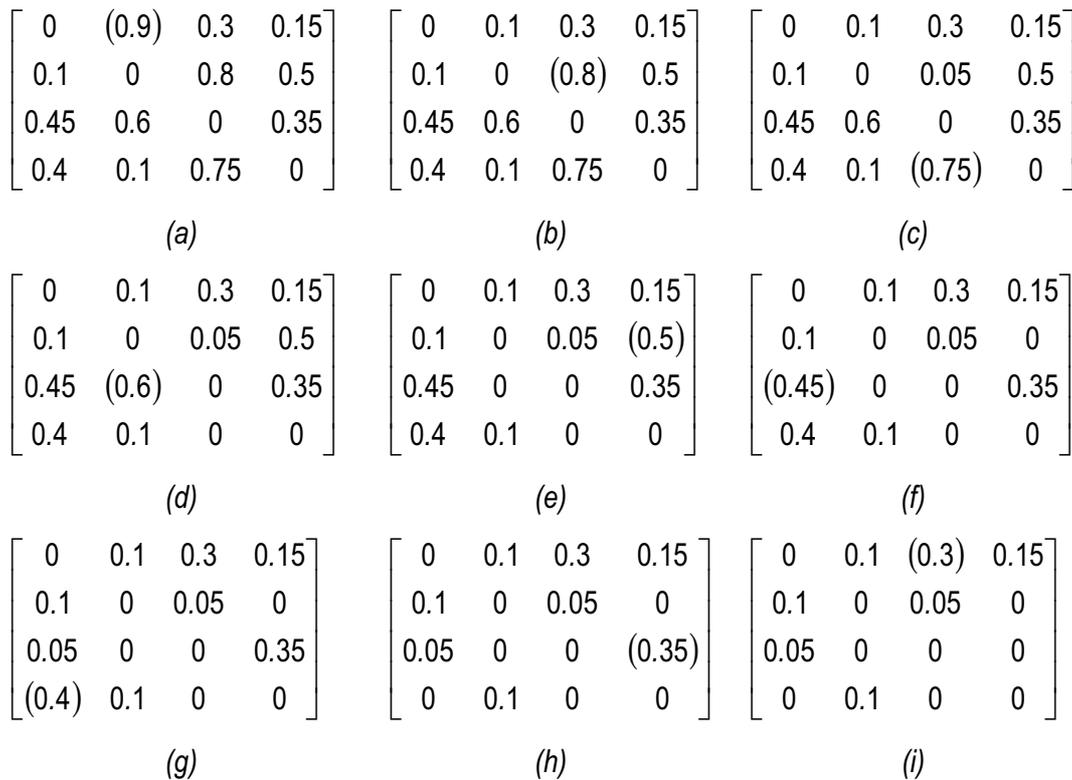


Figura 22. Evolución de la matriz de tráfico en el MLDA

Como consecuencia en todos los nodos se ocupa un número de transmisores y receptores igual al grado físico, de salida y entrada respectivamente; y en todas las fibras se ocupa el primer índice de las longitudes de onda ω . El resultado será una topología virtual que contiene a la topología física, con *lightpath* adicionales añadidos en función de los recursos libres.

Iteración	Par	Ruta	ω	Tx(origen)	Rx(destino)
a	1→2	1→2	ω_1	0	0
b	2→3	2→4→3	ω_1	0	0
c	4→3	4→3	No hay rx libres en el nodo 3		
d	3→2	3→4→2	No hay rx libres en el nodo 2		
e	2→4	2→4	No hay tx libres en el nodo 2		
f	3→1	3→1	ω_1	0	0
g	4→1	4→3→1	No hay rx libres en el nodo 1		
h	3→4	3→4	No hay tx libres en el nodo 3		
i	1→3	1→3	No hay tx libres en el nodo 1 No hay rx libres en el nodo 3		

Figura 23. Tabla de iteraciones. Asignación de *lightpaths* para el ejemplo del MLDA.

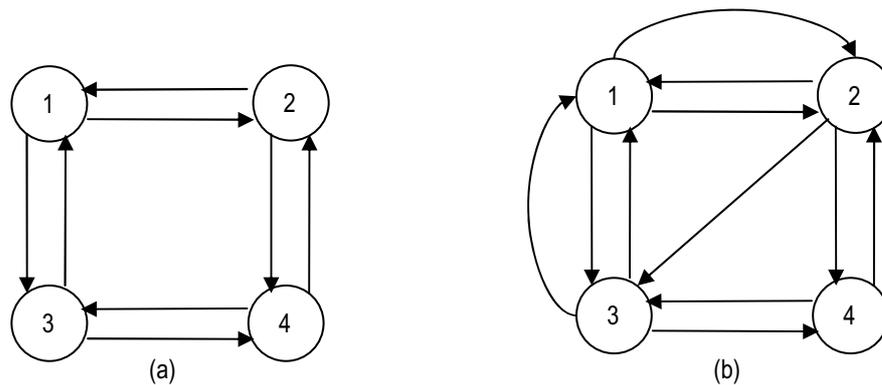


Figura 24. Ejemplo teórico MLDA. (a) Topología física. (b) Topología virtual

3.2.3 LEMA

En este ejemplo se explica el funcionamiento de la primera parte del algoritmo LEMA. Considerar la topología física de la Fig. 25 (a) y la matriz de tráfico de la Fig.25 (b). Cada nodo de la red tiene dos transmisores y dos receptores, la fibra óptica tiene dos longitudes de onda ω y ω_1 .

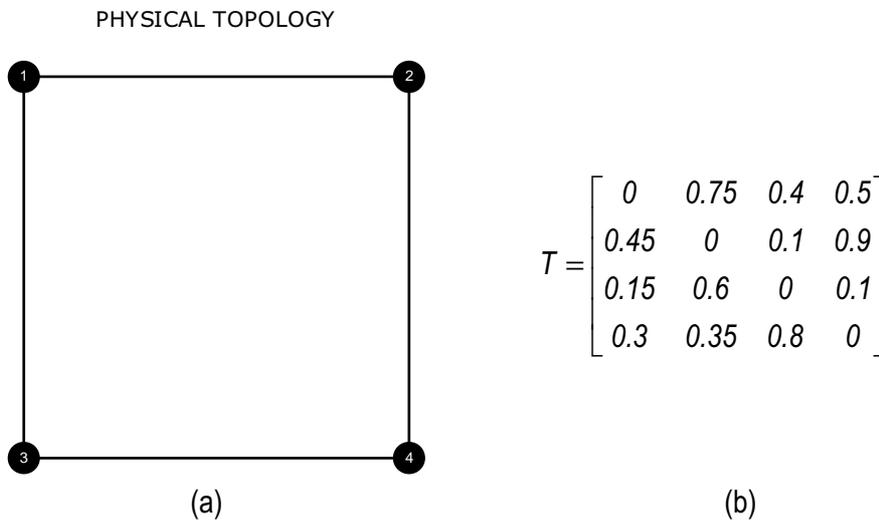
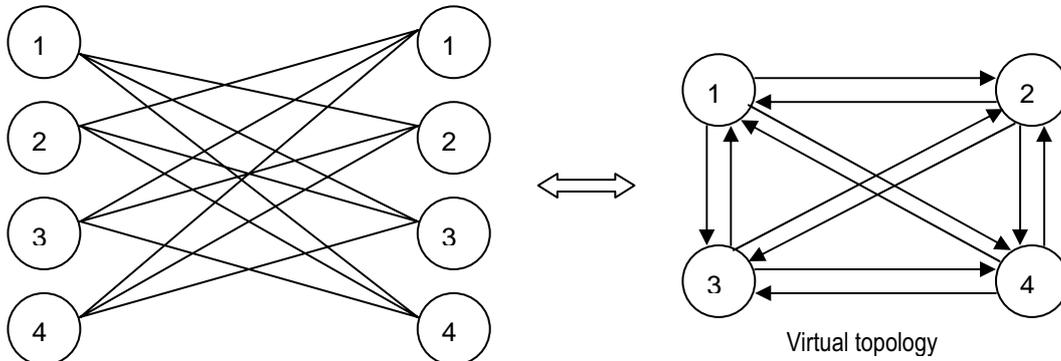


Figura 25. Ejemplo teórico LEMA. (a) Topología física. (b) Matriz de tráfico

En la primera iteración se crea un grafo bipartito completamente conexo, las restricciones de grado no se satisfacen, y es necesario continuar iterando para eliminar algunos enlaces.



- Primera iteración



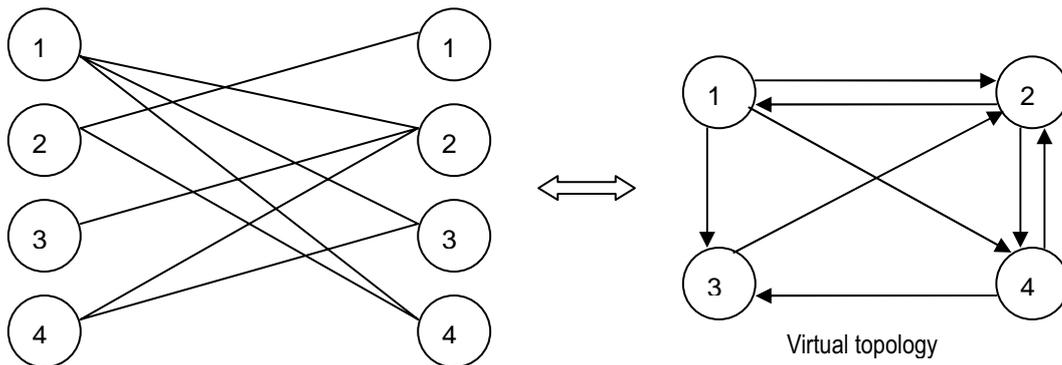
weights =

0	0.75	0.40	0.50
0.45	0	0.10	0.90
0.15	0.60	0	0.10
0.30	0.35	0.80	0

- Segunda iteración

Ahora se seleccionan los cuatro pares menos cargados en la matriz de tráfico, y se eliminan si no desconectan el grafo de la topología virtual. El tráfico de los enlaces eliminados se vuelve a enrutar por las rutas alternativas menos congestionadas.

1. Enlace 2-3 (no desconecta)	2. Enlace 3-4 (no desconecta)
weights =	weights =
0 0.75 0.50 0.50	0 0.75 0.60 0.60
0.55 0 0 0.90	0.65 0 0 0.90
0.15 0.60 0 0.10	0.25 0.70 0 0
0.30 0.35 0.80 0	0.30 0.35 0.80 0
3. Enlace 3-1 (no desconecta)	4. Enlace 4-1 (no desconecta)
weights =	weights =
0 0.75 0.60 0.60	0 0.75 0.60 0.60
0.90 0 0 0.90	1.2 0 0 0.90
0 0.95 0 0	0 0.95 0 0
0.30 0.35 0.80 0	0 0.65 0.80 0

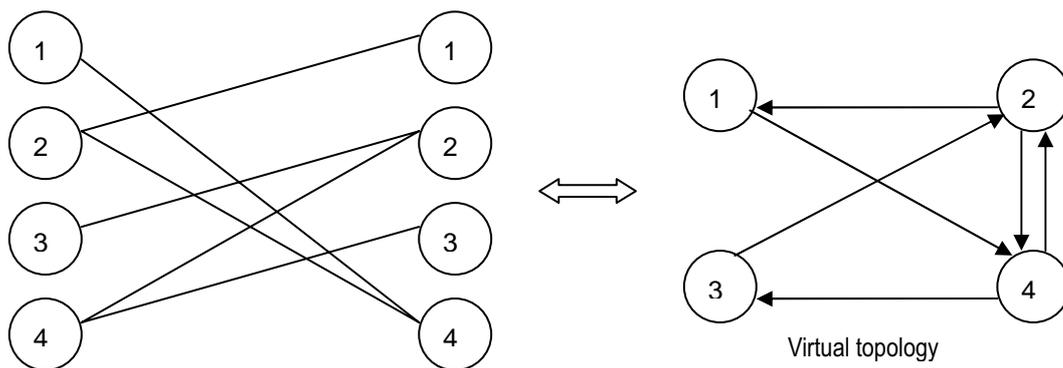


Las restricciones de grado siguen sin satisfacerse, es necesario seguir iterando.

- Tercera iteración

1. Enlace 1-3 (no desconecta)	2. Enlace 1-2 (no desconecta)																																
Weights =	weights =																																
<table border="0"> <tr> <td>0</td> <td>0.75</td> <td>0</td> <td>1.20</td> </tr> <tr> <td>1.20</td> <td>0</td> <td>0</td> <td>1.50</td> </tr> <tr> <td>0</td> <td>0.95</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1.25</td> <td>1.40</td> <td>0</td> </tr> </table>	0	0.75	0	1.20	1.20	0	0	1.50	0	0.95	0	0	0	1.25	1.40	0	<table border="0"> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1.95</td> </tr> <tr> <td>1.20</td> <td>0</td> <td>0</td> <td>1.50</td> </tr> <tr> <td>0</td> <td>0.95</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>2.00</td> <td>1.40</td> <td>0</td> </tr> </table>	0	0	0	1.95	1.20	0	0	1.50	0	0.95	0	0	0	2.00	1.40	0
0	0.75	0	1.20																														
1.20	0	0	1.50																														
0	0.95	0	0																														
0	1.25	1.40	0																														
0	0	0	1.95																														
1.20	0	0	1.50																														
0	0.95	0	0																														
0	2.00	1.40	0																														
3. Enlace 3-2 (desconecta el grafo)																																	
Este enlace no se puede eliminar porque el grafo pierde la conectividad																																	

Es posible eliminar todos los enlaces menos el par 3-2, porque la eliminación de este desconectaría el grafo. Cuando se encuentra un enlace crítico, el resto de los N enlaces menos cogestionados no se evalúan. A continuación se evalúan el grafo bipartito y la topología virtual asociada.



Al finalizar la tercera iteración las restricciones de grado se cumplen en todos los nodos, por lo que se da por terminado el algoritmo.

3.3 ALGORÍTMOS PROGRAMADOS

Cada algoritmo lleva asociado una ficha que contiene: invocación de la función, resumen, argumentos de entrada y salida, función auxiliares y código de errores. Esta información está disponible también en los códigos Matlab de la función correspondiente, accediendo a ella con el comando `help` seguido del nombre la función. También se proporcionan como ficheros web en la aplicación GUI *MatPlanWDM* dentro del menú HELP.



3.3.1 HEURISTIC LOGICAL DESIGN ALGORITHM (HLDA) - v1

FUNCTION

```
[virtualTopology, statisticalInformation, routesMatrix] =  
hldav1(trafficMatrix, physicalTopology, numberTxPerNode, numberRxPerNode,  
numberWavelengthPerFiber, showProgressBar)
```

ABSTRACT

Implementation of the Heuristic Logical Design Algorithm (HLDA) to solve the virtual topology design problem. The traffic matrix between all node pairs source-destination and the physical topology are taken as the input. There is wavelength constraint (without wavelength converters) and the number of transmitter and receivers available at every node is assumed to be given and fixed. Also, the number of available wavelengths per fiber is assumed to be done.

This algorithm attempts to maximize single (virtual)-hop traffic flow. It aims at minimizing congestion in a given network. The network congestion is defined as the maximum offered traffic to any *lightpath*, and the *lightpath* is the sum of all traffic flows between the pair nodes source-destination that use this *lightpath*. The number of established *lightpaths* between any pair node source-destination can be variable. We obtain as output the logic topology or *lightpaths* network.

The HLDA solves three instances of the virtual topology design problem:

- Virtual topology selection (VTS)
- *Lightpath* route selection (LPRS).
- *Lightpath* wavelength selection (LPWS)

The HDLA considers node pairs in nonincreasing order of their traffic. It selects the node pair (say, x) with the most nonzero traffic flow between them. A *lightpath* is established between this node pair, if permissible. A *lightpath* is permissible for node pair x with these requirements:

1. A physical route.
2. A wavelength on the route.
3. A free transmitter at the source node of x .
4. A free receiver at the destination node of x .

When a *lightpath* is established between pair x , the traffic associated with x is updated by subtracting from it the traffic associated with pair y . Here, node pair y has the highest traffic after pair x . If a *lightpath* cannot be established between node pair x , the traffic associated with it is set to zero. Now, the node pair which has the maximum amount of nonzero traffic is chosen and the above procedure is repeated. Note that the chosen node pair could be either x or y . When all the node pairs with nonzero traffic have been considered, the procedure stops.

As it has said above, the HDLA solves the LPRS and LPWS subproblems as follows. Whenever a node pair is considered, a shortest (propagation) delay path (with some free wavelength[s]) is chosen. The wavelength with the lowest index among those which are free on the chosen path is



selected. Note that the VTS, LPRS, and LPWS subproblems are not solved independently.	
INPUT PARAMETERS	
<code>trafficMatrix (NxN)</code>	Traffic between all node pairs, where N is the number of nodes in the network. An entry (i,j) is the traffic from node i to node j expressed in arbitrary units.
<code>physicalTopology (NxN)</code>	Graph of the physical topology of nodes and links, where N is the number of nodes. An entry (i,j) is the weight of the link generated from the node i to the node j . It only can be 1 or 0, in case of the link exists or not respectively. As agreement the values of main diagonal are established to 0.
<code>numberTxPerNode (1x1)</code>	Scalar integer number of optics transmitters per node.
<code>numberRxPerNode (1x1)</code>	Scalar integer number of optics receivers per node.
<code>numberWavelengthPerFiber (1x1)</code>	Scalar integer number of wavelengths per fiber.
<code>showProgressBar (1x1)</code>	Flag which indicates if the progress-bar window must be shown. It only can be 1 or 0, in case of the progress bar is shown or not respectively.
OUTPUT PARAMETERS	
<code>virtualTopology (NxN)</code>	Graph of the virtual topology of established <i>lightpaths</i> , where N is the number of nodes. An entry (i,j) is the weight of the generated <i>lightpaths</i> from node i to the node j . It can be any integer number. Zero indicates that these <i>lightpath</i> doesn't exist.
<code>statisticalInformation (2x1)</code>	1th row: Average percentage of used degree (transmitters and receivers). 2th row: Average percentage of used wavelengths at each fiber edge.
<code>routesMatrix (?xN)</code>	Matrix of physical routes in the established <i>lightpaths</i> . Each row is a new physical route from first node to last non-zero node. There are such columns as number of nodes. Non-used hops (columns) are filled with zeros. Last column indicates the used wavelength index.
ASSISTANTS FUNCTIONS	
<ul style="list-style-type: none"> function [<code>stpTree</code> , <code>fatherNode</code> , <code>inTree</code> , <code>numberOfDescendants</code> , <code>hopsToTheSink</code>] = <code>fiberSTP</code> (<code>sinkNode</code>,<code>neighborhoodMatrix</code>) <p>In order to find a physical route (first condition for the <i>lightpath</i> establishment) between source node and destination node, it's necessary to provide a routing algorithm which works along the topology to find a valid physical path between each pair of nodes. It has been used a modification of the STP (Shortest Spanning Tree), called <code>FiberSTP</code>. It finds a route from the sink node to the rest of nodes in tree distribution. All physical links are bidirectionals and</p>	



the number of hops is the main metric, so distances between all node pairs are identical.

- `function [neighborhoodMatrix] = physicNeighborhood(physicalTopology)`

Change the format of physical topology to the appropriate form of the function `FiberSTP`

ERROR CODES

- 1 – Incorrect number of arguments
- 2 – Incorrect argument dimensions
- 3 – Incorrect argument format



3.3.2 HEURISTIC LOGICAL DESIGN ALGORITHM (HLDA) – v2

FUNCTION	
<pre>[virtualTopology, usedWavelength, statisticalInformation] = hldav2 (trafficMatrix, physicalTopology, txPerNode, rxPerNode, wavelengthPerFiber)</pre>	
ABSTRACT	
<p>Implementation of the Heuristic Logical Design Algorithm (HLDA) to solve the virtual topology design problem. The traffic matrix between all node pairs source-destination and the physical topology are taken as the input. There is wavelength constraint (without wavelength converters). The second version of HLDA stands out because of the number of transmitters and receivers in each node and the number of wavelengths in each fiber are done and not fixed. With this version it is possible to assign a variable value of transmitters, receivers and wavelength to each node/fiber.</p> <p>Also it is provided an output matrix with the number of used wavelengths in all the established <i>lightpaths</i>. Finally some statistical percentage indexes are given, through from it possible to know: the percentage of used transmitters, receivers and wavelengths. This version of HLDA allows us to come closer to a real stage, where different technologies perform together in the same WDM network.</p>	
INPUT PARAMETERS	
trafficMatrix (NxN)	Traffic between all node pairs, where N is the number of nodes in the network. An entry (i,j) is the traffic from node i to node j expressed in arbitrary units.
physicalTopology (NxN)	Graph of the physical topology of nodes and links, where N is the number of nodes. An entry (i,j) is the weight of the link generated from the node i to the node j . It only can be 1 or 0, in case of the link exists or not respectively. As agreement the values of main diagonal are established to 0.
txPerNode (1xN)	Integer number of optics transmitters in each node, where N is the number of nodes.
rxPerNode (1xN)	Integer number of optics receivers in each node, where N is the number of nodes.
wavelengthPerFiber (NxN)	Integer number of wavelengths in each fiber, where N is the number of nodes. An entry (i,j) is the number of wavelengths in the current link ($\text{physicalTopology}(i, j) = 1$) from the node i to the node j .
OUTPUT PARAMETERS	
virtualTopology (NxN)	Graph of the virtual topology of established <i>lightpaths</i> , where N is the number of nodes. An entry (i,j) is the weight



	of the generated <i>lightpaths</i> from node i to the node j . It can be any integer number. Zero indicates that these <i>lightpath</i> doesn't exist.
<code>usedWavelength</code> (NxN)	Index of used wavelengths in all the established <i>lightpaths</i> in the virtual topology. The number of wavelength begins with 1, so this corresponds with ω_0 . If more of one wavelength is used in the same <i>lightpath</i> the index of them are noted as: ex. <code>usedWavelength(1,2)=136</code> (used wavelengths in <i>lightpath</i> from node 1 to node 2 are ω_0 , ω_2 and ω_5).
<code>statisticalInformation</code> (3x1)	Three indexes are done as usage information over all the initial available resources: <ul style="list-style-type: none"> • 1th row: Average percentage of used transmitters at each node. • 2th row: Average percentage of used receivers at each node. • 3th row: Average percentage of used wavelengths at each fiber link.

ASSISTANTS FUNCTIONS

- `function [stpTree , fatherNode , inTree , numberOfDescendants , hopsToTheSink] = fiberSTP (sinkNode,neighborhoodMatrix)`

In order to find a physical route (first condition for the *lightpath* establishment) between source node and destination node, it's necessary to provide a routing algorithm which works along the topology to find a valid physical path between each pair of nodes. It has been used a modification of the STP (Shortest Spanning Tree), called FiberSTP. It finds a route from the sink node to the rest of nodes in tree distribution. All physical links are bidirectional and the number of hops is the main metric, so distances between all node pairs are identical.

- `function [neighborhoodMatrix] = physicNeighborhood (physicalTopology)`

Change the format of physical topology to the appropriate form of the function `FiberSTP`

ERROR CODES

- 1 – Incorrect number of arguments
- 2 – Incorrect arguments dimensions
- 3 – Incorrect arguments format



3.3.3 MINIMUM-DELAY LOGICAL TOPOLOGY DESIGN ALGORITHM (MLDA)

FUNCTION	
<pre>[virtualTopology, statisticalInformation, routesMatrix] = hldav1(trafficMatrix, physicalTopology, numberTxPerNode, numberRxPerNode, numberWavelengthPerFiber, showProgressBar)</pre>	
ABSTRACT	
<p>The minimum-delay logical topology algorithm (MLDA) heuristic ensures that a shortest (propagation delay) path exists for every pair of nodes. While it minimizes the average message delay, its performance may be poor in terms of congestion. The MLDA first creates a <i>lightpath</i> for every edge in the physical topology. Then it uses the HLDA to create new <i>lightpaths</i> between node pairs. Therefore, the virtual topology contains the physical topology. As a result, the degree of a node in the virtual topology must be higher than that of the corresponding node in the physical topology.</p> <p>This algorithm runs with the same code of HLDA. Only some sentences have been introduced to include the physical topology into the logical topology. The traffic matrix between all node pairs source-destination, the physical topology, the number of transmitters, receivers and fibers are taken as the input. There is wavelength continuity constraint (without wavelength converters).</p>	
INPUT PARAMETERS	
trafficMatrix (NxN)	Traffic between all node pairs, where N is the number of nodes in the network. An entry (i,j) is the traffic from node i to node j expressed in arbitrary units.
physicalTopology (NxN)	Graph of the physical topology of nodes and links, where N is the number of nodes. An entry (i,j) is the weight of the link generated from the node i to the node j . It only can be 1 or 0, in case of the link exists or not respectively. As agreement the values of main diagonal are established to 0.
numberTxPerNode (1x1)	Scalar integer number of optics transmitters per node.
numberRxPerNode (1x1)	Scalar integer number of optics receivers per node.
numberWavelengthPerFiber (1x1)	Scalar integer number of wavelengths per fiber.
showProgressBar (1x1)	Flag which indicates if the progress-bar window must be shown. It only can be 1 or 0, in case of the progress bar is shown or not respectively.
OUTPUT PARAMETERS	
virtualTopology (NxN)	Graph of the virtual topology of established <i>lightpaths</i> , where N is the number of nodes. An entry (i,j) is the weight of the generated <i>lightpaths</i>



	from node i to the node j . It can be any integer number. Zero indicates that these <i>lightpath</i> doesn't exist.
statisticalInformation(2x1)	1th row: Average percentage of used degree (transmitters and receivers). 2th row: Average percentage of used wavelengths at each fiber edge.
routesMatrix(?xN)	Matrix of physical routes in the established <i>lightpaths</i> . Each row is a new physical route from first node to last non-zero node. There are such columns as number of nodes. Non-used hops (columns) are filled with zeros. Last column indicates the used wavelength index.
ASSISTANTS FUNCTIONS	
<ul style="list-style-type: none"> function [stpTree , fatherNode , inTree , numberOfDescendants , hopsToTheSink] = fiberSTP (sinkNode, neighborhoodMatrix) <p>In order to find a physical route (first condition for the <i>lightpath</i> establishment) between source node and destination node, it's necessary to provide a routing algorithm which works along the topology to find a valid physical path between each pair of nodes. It has been used a modification of the STP (Shortest Spanning Tree), called <i>FiberSTP</i>. It finds a route from the sink node to the rest of nodes in tree distribution. All physical links are bidirectionals and the number of hops is the main metric, so distances between all node pairs are identical.</p> <ul style="list-style-type: none"> function [neighborhoodMatrix] = physicNeighborhood (physicalTopology) <p>Change the format of physical topology to the appropriate form of the function <i>FiberSTP</i></p>	
ERROR CODES	
<p>1 – Incorrect number of arguments 2 – Incorrect argument dimensions 3 – Incorrect argument format</p>	



3.3.4 TRAFFIC INDEPENDENT LOGICAL DESIGN ALGORITHM (TILDA)

FUNCTION	
<pre>[virtualTopology, statisticalInformation, routesMatrix] = hldav1(trafficMatrix, physicalTopology, numberTxPerNode, numberRxPerNode, numberWavelengthPerFiber, showProgressBar)</pre>	
ABSTRACT	
<p>TILDA designs logical topology regardless of the traffic. It first places logical edges between all one-hop neighbours in the physical topology, then between all two-hop neighbours (provided that there are no logical edges between them already), then between all three-hop neighbours (provided that there are no logical edges between them already), etc., provided the degree constraints are not violated. Since <i>lightpaths</i> which use as few physical topology edges as possible will tend to keep the number of wavelengths required, and may be an appropriate choice if the traffic is unknown or known to be uniform.</p>	
INPUT PARAMETERS	
physicalTopology (NxN)	Graph of the physical topology of nodes and links, where N is the number of nodes. An entry (i,j) is the weight of the link generated from the node i to the node j . It only can be 1 or 0, in case of the link exists or not respectively. As agreement the values of main diagonal are established to 0.
numberTxPerNode (1x1)	Scalar integer number of optics transmitters per node.
numberRxPerNode (1x1)	Scalar integer number of optics receivers per node.
numberWavelengthPerFiber (1x1)	Scalar integer number of wavelengths per fiber.
showProgressBar (1x1)	Flag which indicates if the progress-bar window must be shown. It only can be 1 or 0, in case of the progress bar is shown or not respectively.
OUTPUT PARAMETERS	
virtualTopology (NxN)	Graph of the virtual topology of established <i>lightpaths</i> , where N is the number of nodes. An entry (i,j) is the weight of the generated <i>lightpaths</i> from node i to the node j . It can be any integer number. Zero indicates that these <i>lightpath</i> doesn't exist.
statisticalInformation (2x1)	1th row: Average percentage of used degree (transmitters and receivers). 2th row: Average percentage of used wavelengths at each fiber edge.
routesMatrix (?xN)	Matrix of physical routes in the established <i>lightpaths</i> . Each row is a new physical route from first node to last non-zero node. There are such



	columns as number of nodes. Non-used hops (columns) are filled with zeros. Last column indicates the used wavelength index.
ASSISTANTS FUNCTIONS	
<ul style="list-style-type: none"> function [stpTree , fatherNode , inTree , numberOfDescendants , hopsToTheSink] = fiberSTP (sinkNode,neighborhoodMatrix) <p>In order to find a physical route (first condition for the <i>lightpath</i> establishment) between source node and destination node, it's necessary to provide a routing algorithm which works along the topology to find a valid physical path between each pair of nodes. It has been used a modification of the STP (Shortest Spanning Tree), called <code>FiberSTP</code>. It finds a route from the sink node to the rest of nodes in tree distribution. All physical links are bidirectionals and the number of hops is the main metric, so distances between all node pairs are identical.</p> <ul style="list-style-type: none"> function [neighborhoodMatrix] = physicNeighborhood (physicalTopology) <p>Change the format of physical topology to the appropriate form of the function <code>FiberSTP</code></p>	
ERROR CODES	
<p>1 – Incorrect number of arguments 2 – Incorrect argument dimensions 3 – Incorrect argument format</p>	



3.3.5 RANDOM LOGICAL DESIGN ALGORITHM (RLDA)

FUNCTION	
<pre>[virtualTopology, statisticalInformation, routesMatrix] = hldav1(trafficMatrix, physicalTopology, numberTxPerNode, numberRxPerNode, numberWavelengthPerFiber, showProgressBar)</pre>	
ABSTRACT	
<p>RLDA is the simplest heuristic algorithm, but very useful in terms of comparison proposes. It places logical edges entirely at random, subject to finding a <i>lightpath</i> for each edge and not violating degree constraints, but ignoring traffic matrix altogether.</p>	
INPUT PARAMETERS	
physicalTopology (NxN)	Graph of the physical topology of nodes and links, where N is the number of nodes. An entry (i,j) is the weight of the link generated from the node i to the node j . It only can be 1 or 0, in case of the link exists or not respectively. As agreement the values of main diagonal are established to 0.
numberTxPerNode (1x1)	Scalar integer number of optics transmitters per node.
numberRxPerNode (1x1)	Scalar integer number of optics receivers per node.
numberWavelengthPerFiber (1x1)	Scalar integer number of wavelengths per fiber.
showProgressBar (1x1)	Flag which indicates if the progress-bar window must be shown. It only can be 1 or 0, in case of the progress bar is shown or not respectively.
OUTPUT PARAMETERS	
virtualTopology (NxN)	Graph of the virtual topology of established <i>lightpaths</i> , where N is the number of nodes. An entry (i,j) is the weight of the generated <i>lightpaths</i> from node i to the node j . It can be any integer number. Zero indicates that these <i>lightpath</i> doesn't exist.
statisticalInformation (2x1)	1th row: Average percentage of used degree (transmitters and receivers). 2th row: Average percentage of used wavelengths at each fiber edge.
routesMatrix (?xN)	Matrix of physical routes in the established <i>lightpaths</i> . Each row is a new physical route from first node to last non-zero node. There are such columns as number of nodes. Non-used hops (columns) are filled with zeros. Last column indicates the used wavelength index.



ASSISTANTS FUNCTIONS

- `function [stpTree , fatherNode , inTree ,
numberOfDescendants , hopsToTheSink] = fiberSTP
(sinkNode,neighborhoodMatrix)`

In order to find a physical route (first condition for the *lightpath* establishment) between source node and destination node, it's necessary to provide a routing algorithm which works along the topology to find a valid physical path between each pair of nodes. It has been used a modification of the STP (Shortest Spanning Tree), called `FiberSTP`. It finds a route from the sink node to the rest of nodes in tree distribution. All physical links are bidirectionals and the number of hops is the main metric, so distances between all node pairs are identical.

- `function [neighborhoodMatrix] = physicNeighborhood
(physicalTopology)`

Change the format of physical topology to the appropriate form of the function `FiberSTP`

ERROR CODES

- 1 – Incorrect number of arguments
- 2 – Incorrect argument dimensions
- 3 – Incorrect argument format



CAPÍTULO 4

Descripción de la herramienta

4.1 INTRODUCCIÓN A LA HERRAMIENTA

En la sección 3.3 del capítulo anterior se han descrito las cuatro funciones implementadas en Matlab: `hldav1.m`, `hldav2.m`, `mlda.m`, `tilda.m` y `rlda.m`. Estos cuatro algoritmos forman el núcleo fundamental de la herramienta, que las pondrá en funcionamiento aportando una rica casuística de topologías físicas y patrones de tráfico. Como se ha desarrollado ya teóricamente, los algoritmos están programados en nivel decreciente de complejidad, por lo que sus resultados comparativos *a priori* ya se conocen, aunque será muy interesante constatar el comportamiento en escenarios variables.

Como banco de pruebas de los algoritmos se han diseñado un entrenador de algoritmos y un diseñador de topologías virtuales. Ambas rutinas incluyen funciones auxiliares y medidas que se describirán minuciosamente. Por último, la funcionalidad de estos programas se incluyen en la **herramienta final: MatPlanWDM**. Se trata de una aplicación en GUI de Matlab con un entorno amigable e intuitivo que permite de forma fácil y accesible simular los algoritmos en distintos escenarios, evaluar un algoritmo en condiciones variables o comparar los resultados de los cuatro algoritmos bajo las mismas condiciones. Para el correcto funcionamiento del GUI se presenta un completo manual de usuario, dirigido sobre todo a un entorno docente.

4.2 ESCENARIOS

Entendemos por escenario la topología física y la matriz de tráfico que modelan la topología virtual obtenida de un algoritmo. Son las dos variables de entrada clásicas e imprescindibles en



el funcionamiento de la mayoría de los algoritmos. De un conjunto amplio de topologías y patrones se puede tener acceso a un rico abanico de simulaciones, que en conjunto permiten modelar de forma precisa el comportamiento de un algoritmo heurístico.

Mientras la matriz de tráfico es prescindible, en algunos algoritmos que trabajan sin ella (TILDA, RLDA), la topología física es estrictamente necesaria, a la hora de plantear al problema.

4.2.1 Topologías físicas

Para poder evaluar correctamente un heurístico, los resultados tienen que presentarse en distintas topologías físicas. Pese a trabajar con heurísticos, que teóricamente posibilitan manejar un número de nodos mayor, las topologías físicas que habitualmente se utilizan para simular los diseños de topologías virtuales están por debajo de 20 nodos. Son los tamaños normales de las redes WAN, pero también existe una ligadura respecto a los problemas computacionalmente posibles. Como se verá posteriormente, los tiempos exponenciales en la resolución de la topología virtual, por medio de la programación lineal, no permiten abordar problemas de más de una decena de nodos. Por lo tanto, y aunque un algoritmo heurístico obtenga estas soluciones en un tiempo ínfimo, no se puede obtener una medida de la bondad del resultado, sino disponemos de una referencia, aunque sean los valores de los límites presentados en el segundo capítulo.

Las topologías se dibujan mediante una de las funciones auxiliares programadas en el desarrollo de esta herramienta, y que se presentará en apartados siguientes, son grafos de enlaces bidireccionales que se representa con una línea simple de color naranja.

4.2.1.1 Red de 4 nodos. SIMPLENET

La red de 4 nodos con enlaces perimetrales tiene un valor fundamentalmente didáctico y teórico. Su representación con la función `plotTopology.m` que incluye nuestra herramienta, está en la Fig. 26. Como se ha visto, en los ejemplos teóricos de la sección 3.2, estos se hacen mayoritariamente sobre este tipo de red, que requiere pocas iteraciones. Su resolución y la posterior comprobación de los resultados implementados, orientan de forma eficiente sobre el correcto funcionamiento del heurístico. También se toma de referencia en redes *backbone* de muy pocos nodos, donde se requieren tiempos de resolución bajos.

Physical Topology, 4-node SIMPLENET

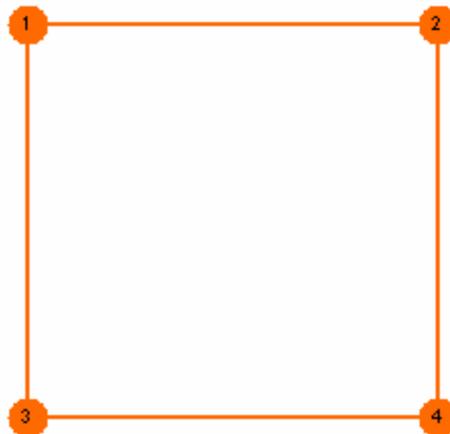


Figura 26. Topología física de 4 nodos. SIMPLENET

4.2.1.2 Red de 9 nodos. POELTENET

Esta topología está basada en la posición geográfica de nueve ciudades centroeuropeas, cuya posición central corresponde a la ciudad austriaca de Sankt Poelten de la que recibe el nombre. La topología física de nueve nodos supone un salto cualitativo en complejidad, tanto para los enfoques de programación lineal, como para los heurísticos. El número de iteraciones en nuestros algoritmos crece sensiblemente y abordarlo mediante un *solver* puede suponer varias horas de simulación.

Physical Topology, 9-node POELTENET

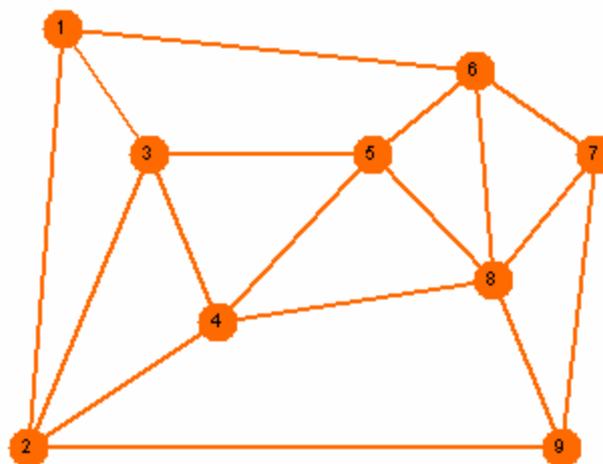


Figura 27. Topología física de nueve nodos. POELTENET

4.2.1.3 Red de 14 nodos. NSFNET

La topología NSFNET está inspirada en la red de comunicaciones creada en EE.UU. en el año 1986 por la *National Science Foundation* (NSF), organismo público que se hizo cargo de administración y gestión de la red durante los primeros tiempos. La NSFNET empezó siendo el modelo de referencia de redes de larga distancia, y aunque su concepción inicial (su primer *backbone* era de 56 Kbps) fue rápidamente rebasada por el crecimiento de la red, sus características la han hecho idónea, para el estudio las diversas tecnologías posteriores. Sus catorce nodos conforman un ejemplo estándar de red *backbone* de larga distancia.

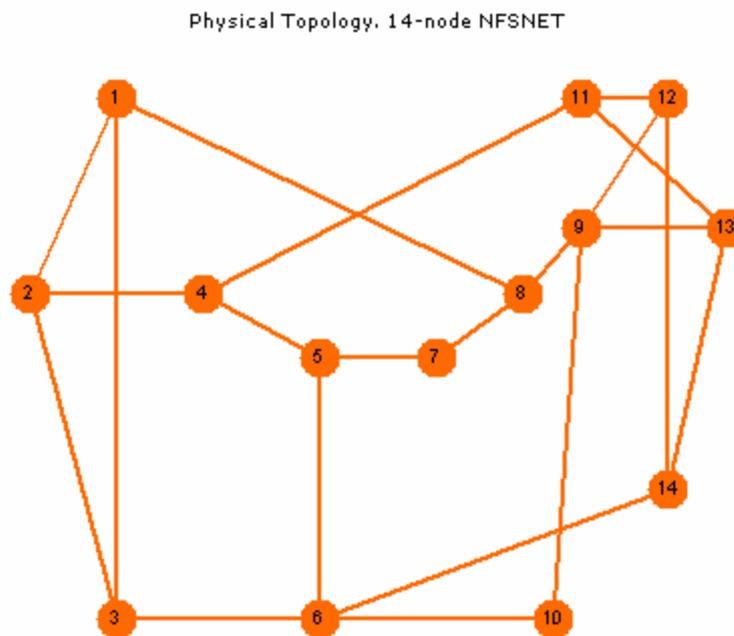


Figura 28. Topología física de 14 nodos. NSFNET

4.2.2 Patrones de tráfico

Existen gran variedad de patrones de tráfico que obedecen a modelos que se aproximan a escenarios reales. Mediante ellos es posible probar un algoritmo para poder prever el comportamiento que puede tener una situación de tráfico real. El hecho de que los algoritmos utilicen, en nuestro caso para minimizar la congestión, criterios sobre la carga de los nodos; hace que los resultados globales se ajusten mejor a unos patrones que a otros. Por ejemplo, un algoritmo que atiende en primer lugar los nodos cargados, e.g. HLDA, responde mejor con un patrón de tráfico descompensado, donde un cierto porcentaje de nodos presenta un volumen superior en varios órdenes que el resto. Sin embargo este mismo algoritmo, evaluado ante un



patrón uniforme no optimiza tanto los *lightpaths* establecidos, ya que la diferencia entre la carga de los pares con más tráfico y el resto, no es sustancial.

Por lo tanto, no es aventurado decir que en función también del patrón de tráfico, un algoritmo será más recomendable que otro, relativizando una vez más el término eficiencia, comprendido como el conjunto de variables (y resultados) a tener en cuenta.

4.2.2.4 Uniformes

Si bien los patrones de tráfico uniformes no son realistas cuando se tienen que modelar demandas en redes pequeñas, cuando se trata de redes de larga distancia WAN los patrones uniformes se ajustan de forma aproximada a los escenarios reales. Por ello se proporcionan dos patrones de tráfico uniforme en dos órdenes de magnitud de diferente.

1. Distribución uniforme U(0,1). Todos los pares de nodos presentan un carga de tráfico extraída aleatoriamente de una distribución uniforme U(0,1).

```
trafficMatrix =
    0      0.88    0.37    0.88    0.38    0.56    0.37    0.74    0.59
0.096    0      0.065   0.51    0.91    0.37    0.76    0.0047  0.95
0.64     0.8     0      0.96    0.67    0.79    0.38    0.6     0.26
0.44     0.44    0.84     0      0.96    0.8     0.9     0.96    0.51
0.066    0.98    0.15    0.048   0      0.38    0.18    0.4     0.64
0.37     0.096   0.17    0.38    0.75     0      0.37    0.73    0.4
0.25     0.53    0.068   0.41    0.37    0.34     0      0.68    0.49
0.92     0.55    0.82     0.4     0.45    0.97    0.52     0      0.75
0.63     0.28    0.13    0.42    0.039   0.48    0.09    0.2     0
```

2. Distribución uniforme U(0,100). Todos los pares de nodos presentan un volumen de tráfico extraído aleatoriamente de una distribución uniforme U(0,100).

```
trafficMatrix =
    0      31      99      93
    19     0      69      13
    6.4    98     0      69
    7.5    56     81     0
```

4.2.2.5 No uniformes

En un escenario de patrones no uniformes, se asume que hay pares de nodos cargados por encima de otros, varios órdenes de magnitud. Esto en arquitecturas WAN, se entiende como nodos jerárquicamente iguales al resto (de la red de *backbone*) pero con una carga de tráfico



sensiblemente mayor, por ejemplo grandes ciudades, donde el tráfico depende frecuentemente del número de habitantes. Igual que en el caso anterior se presentan porcentajes distintos de nodos cargados, así como un último patrón específico para la red NSFNET.

1. Distribución no uniforme. El tráfico entre el 50 % de los pares se toma aleatoriamente de una distribución uniforme $U(0,100)$. El 50 % de pares restantes se toma de una distribución uniforme $U(0,10)$.

```
trafficMatrix =
  0      3.1      8.5      48      9.2      7.8      6.3      82      7.4
9.4     0        64      21      3.3      60      6.6      28      36
16      14       0        25      2.1      62      5.4      40      4.4
9.1     6.4     1.7       0        3.6     4.7      7.5      7.8     77
4.5     50       25      8.2       0        2.3     47      3.3     5.9
8.1     7.7     44      2.3  1e+002    0        9.2     51      23
63     0.54     98      82      7.2     3.2      0        51      28
60     1.1     0.37     17      90      61      94      0       1.1
6.9     45      3.1     5.9     6.4     7.9      68      25      0
```

2. Distribución no uniforme. El tráfico entre el 25 % de los pares de nodos tomados aleatoriamente es un tráfico uniforme de 0 a 100 – $U(0,100)$. El tráfico del resto de los pares se toma de una distribución uniforme $U(0,10)$.

```
trafficMatrix =
  0      46      2.6      6.2
0.48     0      7.1      9.9
4.6     2.9      0        67
91      7      7.3      0
```

3. Distribución no uniforme. Patrón de tráfico sólo válido para la red NSFNET de 14 nodos. 42 pares escogidos aleatoriamente, tienen un tráfico tomado de una distribución uniforme $U(0,100)$. A los pares de nodos restantes, les corresponde un volumen de tráfico tomado de una distribución uniforme $U(0,1)$. El ejemplo de la matriz se omite por el tamaño para su impresión.



4.3 FUNCIONES AUXILIARES

Para la implementación de las dos rutinas de prueba se han creado funciones auxiliares para poder visualizar mejor los resultados de las simulaciones. Tienen funciones de representación, extracción de medidas e indicación de la evolución del algoritmo. Sus fichas se presentan a continuación.

4.3.1 PLOT TOPOLOGY

FUNCTION	
<code>plotTopology(nodesPlaceMatrix, edgesMatrix, label, type)</code>	
ABSTRACT	
<p>This function draws a topology of nodes and edges when it is done, the positions of nodes, the matrix of edges information and the kind of representation. The kind of representation can be physical or virtual (<i>lightpath</i> network). In the physical representation it is assumed bidirectional edges between each node pair (graph theory). The <i>lightpath</i> network or virtual topology is represented by means of blue arrows. In this case a number over the drawn link points the number of established <i>lightpaths</i> in each arrow. Node and edges physical are drawn in orange, since this color is used in optical networks.</p>	
INPUT PARAMETERS	
<code>nodesPlaceMatrix (Nx2)</code>	XY coordinates of nodes, where N is the number of nodes. They are contained in the first and the second column respectively.
<code>edgesMatrix (NxN)</code>	Information about the edges in the topology, where N is the number of nodes. An entry (i,j) is the number of edges (physical or logical) from the node i to the node j . 0 entry indicates absence of link. If the matrix represents physical topology each entry can be only 0 or 1 and it must be symmetrical. As agreement the values of main diagonal are established to 0.
<code>type (1x1)</code>	Indicates the kind of representation. <ul style="list-style-type: none"> • If <code>type=0</code> (physical representation), the edges are drawn according to graph theory. • If <code>type=1</code> (virtual representation), the edges are drawn as arrows.
ASSISTANT FUNCTIONS	



<ul style="list-style-type: none"> • <code>function [h,yy,zz] = arrow(varargin)</code> <p>It draws the arrows of the virtual topology. It allows several uses in the position of start-stop points, dimensions and structure of the arrow and handles properties.</p>
ERROR CODES
<p>1 – Incorrect number of arguments 2 – Argument <code>type</code> must be either 0 or 1 3 – Argument <code>nodePlaceMatrix</code> must have two columns</p>

A continuación se muestran dos ejemplos de la utilización de la función `plotTopology`, en sus dos tipos. La representación corresponde a la red AustriaNET de 8 nodos.

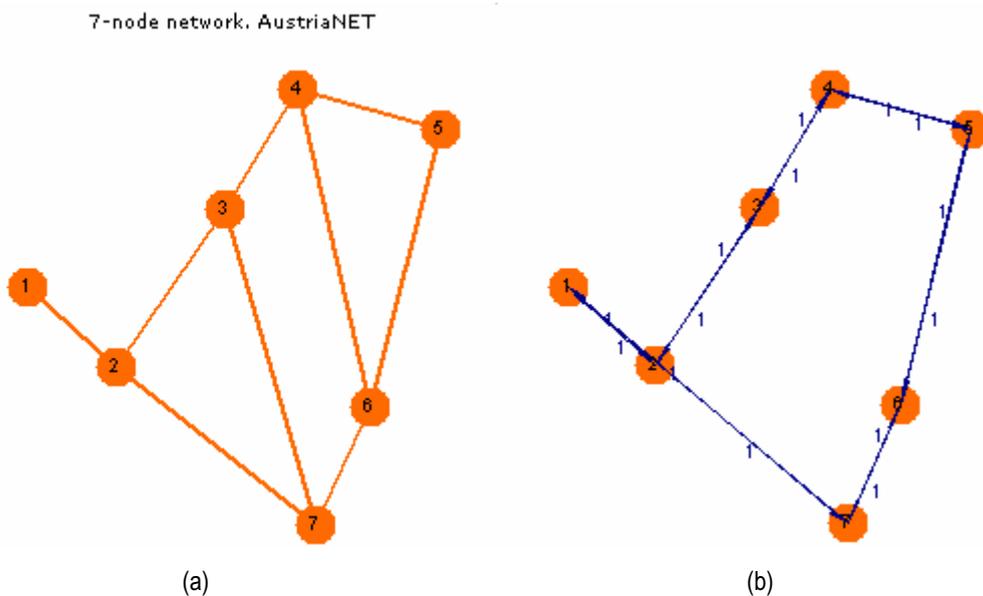


Figura 29. (a) Topología física de la red AustriaNET. (b) Topología virtual

4.3.2 CONGESTION
FUNCTION
<code>congestion = congestion(virtualTopology,trafficMatrix)</code>
ABSTRACT
<p>This function provides an arbitrary congestion value for comparison proposes between algorithms. Virtual topology and traffic matrix are done as input and it determinates global congestion in the network. It is supposed each <i>lightpath</i> carries all the traffic of the node pair, so</p>



when a *lightpath* is established all the traffic in that pair is fixed to zero. The non-zero traffic amounts are sum in both dimensions of the matrix, and that value corresponds with the congestion.

INPUT PARAMETERS

<code>virtualTopology (NxN)</code>	Graph of the virtual topology of established <i>lightpaths</i> , where N is the number of nodes. An entry (i,j) is the weight of the generated <i>lightpaths</i> from node i to the node j . It can be any integer number. Zero indicates that these <i>lightpath</i> doesn't exist.
<code>traffiMatrix (NxN)</code>	Traffic between all node pairs, where N is the number of nodes in the network. An entry (i,j) is the traffic from node i to node j expressed in arbitrary units.

4.3.3 STATUSBAR

FUNCTION

`f=statusbar (p, f)`

ABSTRACT

Display a status/progress bar and inform about the elapsed as well as the remaining time (linear estimation).

`f=statusbar`
 Get all status/progress bar handles.

`f=statusbar(title)`
 Create a new status/progress bar. If title is an empty string, the default 'Progress ...' will be used.

`f=statusbar(title,f)`
 Reset an existing status/progress bar or create a new if the handle became invalid.

`f=statusbar(done,f)`
 For $0 < done < 1$, update the progress bar and the elapsed time. Estimate the remaining time until completion. On user abort, return an empty handle.

`delete(statusbar)`
 Remove all status/progress bars.

`drawnow`
 Refresh all GUI windows.



4.4 MANUAL DE LA HERRAMIENTA. MATPLANWDM.

4.4.1 Programas designer.m y trainer.m

Como paso previo al diseño e implementación de la aplicación en GUI se desarrollaron dos programas que incluían las funciones que debía contener el MatPlanWDM. El primero de ellos diseña topologías virtuales a partir de los datos de la simulación, y el segundo permite comparar la bondad de los algoritmos heurísticos entre sí, o en un escenario cambiante.

4.4.1.6 designer.m

La rutina implementada en designer.m resuelve mediante uno de los cuatro algoritmos el problema del diseño de la topología virtual. Para ello proporciona tres topologías físicas y cinco patrones de tráfico, dejando a elección del usuario el número de transmisores y receptores por nodos, así como el número de longitudes de onda por fibra. A continuación se muestra un ejemplo de la consola de entrada de datos: red de cuatro nodos, distribución uniforme $U(0,1)$, dos transmisores y receptores por nodos, dos longitudes de onda por fibra y resuelto con el HLDA.

```
#####  
VIRTUAL TOPOLOGY DESIGNER FOR WDM NETWORKS v1.0  
#####  
  
Select the desired physical topology:  
  
1. 4-NODE SIMPLNET  
2. 9-NODE POELTENET  
3. 14-NODE NFSNET  
  
    selection: 1  
  
Select the desired traffic pattern:  
  
1. UNIFORM DISTRIBUTION (0,1)  
2. UNIFORM DISTRIBUTION (0,100)  
3. NON UNIFORM DISTRIBUTION. 50 % U(0,100) & 50 % U(0,10)  
4. NON UNIFORM DISTRIBUTION. 25 % U(0,100) & 75 % U(0,10)  
5. ONLY FOR 14-NODE NFSNET. NON UNIFORM DISTRIBUTION. 42 RANDOM PAIRS  
   U(0,100) & REST U(0,1)  
  
    selection: 1
```



```

trafficMatrix =

      0      0.0579      0.1389      0.2722
0.9169      0      0.2028      0.1988
0.4103      0.8132      0      0.0153
0.8936      0.0099      0.6038      0

Node degree: 2
Wavelengths per fiber: 2

Select the heuristic algorithm:

1. HDLA
2. MDLA
3. TILDA
4. RLDA

      selection: 1
    
```

Los resultados de la topología virtual se obtienen en un informe de resultados, así como en la representación gráfica con la función plotTopology.

Informe de resultados:

```

#####
VIRTUAL TOPOLOGY RESULTS
-----

Generated in: 0.109 seconds

Obtained for a 4-node SIMPLINET
Traffic pattern: Uniform distribution U(0,1).
Node degree: 2
Wavelengths per fiber: 2
Used algorithm: HLDA (Heuristic Logical Design Algorithm)

Physical routes and used wavelengths:

-----
2  1  -  -  1
4  3  1  -  1
3  4  2  -  1
4  3  -  -  2
1  3  4  -  2
2  4  -  -  1
1  3  -  -  1

-----

Congestion (arbitrary units): 0.69606
Used transmitters and receivers: 87.5 %
Used wavelengths: 62.5 %

#####
St. Poelten University of Applied Sciences
Gonzalo J. Moreno Muñoz, 2006
#####
    
```

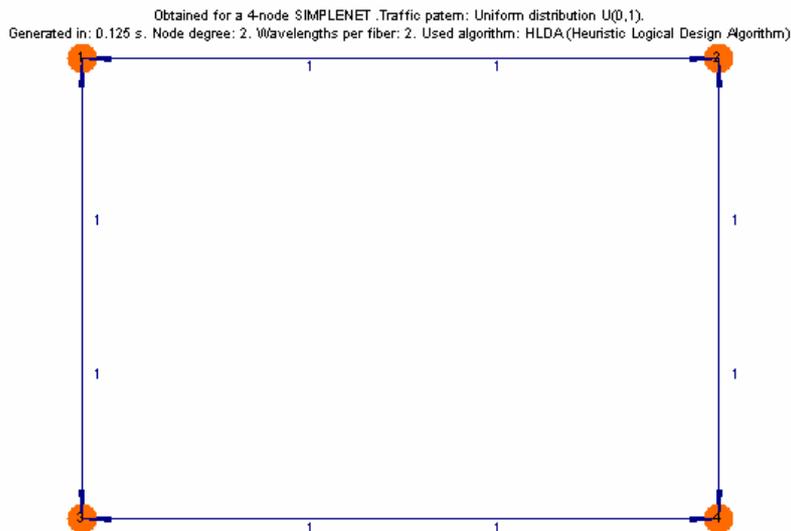


Figura 30. Topología virtual obtenida con el programa designer.m

4.4.1.7 trainer.m

Una vez se ha resuelto un problema concreto de diseño de topología virtual, para caracterizar correctamente un algoritmo heurístico es preciso evaluarlo en un escenario cambiante y compararlos entre ellos. Ese es el objetivo del programa trainer.m, que ofrece dos modalidades: evaluar un algoritmo individualmente o compararlos conjuntamente. Parte de las mismas posibilidades de simulación que el programa designer.m (topologías y patrones) pero incluye la posibilidad de simular un rango de transmisores y receptores por nodos, y en un rango de longitudes de onda por fibra.

1ª Alternativa. Evaluar un algoritmo de forma individual.

Para caracterizar el comportamiento de un algoritmo de forma individual se establecen dos dimensiones de variación: el número de transmisores y receptores por nodo; y el número de longitudes de onda por fibra. Estos dos intervalos, los pide el programa durante la adquisición de datos. A continuación se muestra un ejemplo de la consola de entrada de datos: red de nueve nodos, distribución uniforme U(0,100), un rango de transmisores y receptores por nodo de 2 a 9 (en saltos de 1) y un rango de longitudes de onda de 2 a 9 (en saltos de 1).



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WDM HEURISTIC ALGORITHMS SIMULATOR TOOL v1.0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Select the desired option:

1. Evaluate a specific heuristic algorithm
2. Compare heuristic algorithms

    selection: 1

Select the desired physical topology:

1. 4-NODE SIMPLENET
2. 9-NODE POELTENET
3. 14-NODE NFSNET

    selection: 2

Select the desired traffic pattern:

1. UNIFORM DISTRIBUTION (0,1)
2. UNIFORM DISTRIBUTION (0,100)
3. NON UNIFORM DISTRIBUTION. 50 % U(0,100) & 50 % U(0,10)
4. NON UNIFORM DISTRIBUTION. 25 % U(0,100) & 75 % U(0,10)
5. ONLY FOR 14-NODE NFSNET. NON UNIFORM DISTRIBUTION. 42 RANDOM PAIRS
   U(0,100) & REST U(0,1)

    selection: 2

NODE DEGREE INTERVAL
Low node degree limit: 2
Gap of node degree: 1
High node degree limit 9

Select the heuristic algorithm:

1. HDLA
2. MDLA
3. TILDA
4. RLDA

    selection: 1

WAVELENGTH PER FIBER INTERVAL
Low wavelengths per fiber limit: 2
Gap of wavelengths per fiber: 1
High wavelengths per fiber limit: 9
```

Los resultados obtenidos se componen del informe numérico y de las curvas de congestión. En la gráfica, se representa la congestión en el eje de ordenadas, frente al número de transceptores, en el eje de abscisas. Se dibuja una curva por cada valor del número de longitudes de onda por fibra, simulada.



Informe de resultados:

```
#####
SIMULATION RESULTS 17/9/2006
-----

Used algorithm: HLDA (Heuristic Logical Design Algorithm)
Number of simulations: 64. Run in 11.766 seconds

Obtained for a 9-node POELTENET
Traffic pattern: Uniform distribution U(0,100).
Node degree interval between 2 and 9 (8 measurements)
Numbers of wavelengths per fiber simulated: 2 3 4 5 6 7 8 9

#####
St. Poelten University of Applied Sciences
Gonzalo J. Moreno Muñoz, 2006
#####
```

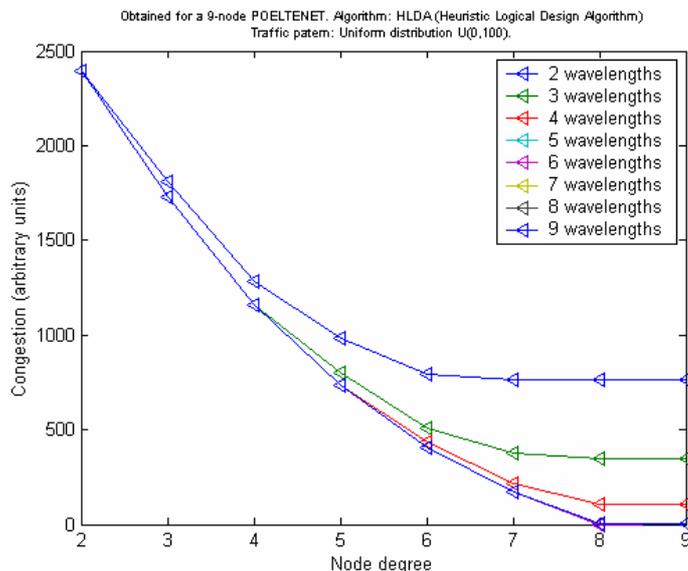


Figura 31. Curvas de congestión obtenidas con el programa trainer.m

2ª Alternativa. Comparar los algoritmos entre sí.

Esta segunda opción que incluye el programa trainer.m, permite comparar en términos de congestión el comportamiento de cada uno de los cuatro algoritmos heurísticos en el mismo escenario (topología física y patrón de tráfico) pero con un número variable de transceptores. De esta forma se puede averiguar la idoneidad de un algoritmo, bajo unas condiciones de funcionamiento determinadas.



A continuación se muestra la ejecución del programa para un ejemplo de topología de 14 nodos, patrón de tráfico NSFNET, número de transceptores variable de 5 a 14 (en saltos de 1) y número de longitudes fijo e igual a 12.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
WDM HEURISTIC ALGORITHMS SIMULATOR TOOL v1.0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Select the desired option:

1. Evaluate a specific heuristic algorithm
2. Compare heuristic algorithms

      selection: 2

Select the desired physical topology:

1. 4-NODE SIMPLENET
2. 9-NODE POELTENET
3. 14-NODE NFSNET

      selection: 3

Select the desired traffic pattern:

1. UNIFORM DISTRIBUTION (0,1)
2. UNIFORM DISTRIBUTION (0,100)
3. NON UNIFORM DISTRIBUTION. 50 % U(0,100) & 50 % U(0,10)
4. NON UNIFORM DISTRIBUTION. 25 % U(0,100) & 75 % U(0,10)
5. ONLY FOR 14-NODE NFSNET. NON UNIFORM DISTRIBUTION. 42 RANDOM PAIRS
   U(0,100) & REST U(0,1)

selection: 5

NODE DEGREE INTERVAL
Low node degree limit: 5
Gap of node degree: 1
High node degree limit 14

Wavelengths per fiber: 12

Working HLDA.....ready
Working MLDA.....ready
Working TILDA.....ready
Working RLDA.....ready

#####
#####
```

El informe de resultados revela los tiempos de simulación para cada algoritmo, y la curvas de congestión muestran los resultados que cada algoritmo proporciona respecto a los demás.



```
#####
SIMULATION RESULTS - 17/9/2006
-----

Algorithms compared: HLDA, MLDA, TILDA and RLDA

Obtained for a 14-node NFSNET
Traffic pattern: Non uniform distribution. 42 random pairs U(0,100) the rest
U(0,1).
Interval of Tx and Rx: 5 and 14 (10 measurements)
Wavelengths per fiber : 12
Number of simulations: 10

-----

HLDA. Run in 7.218 seconds
MLDA. Run in 5.594 seconds
TILDA. Run in 0.75 seconds
RLDA. Run in 0.875 seconds

#####
St. Poelten University of Applied Sciences
Gonzalo J. Moreno Muñoz, 2006
#####
```

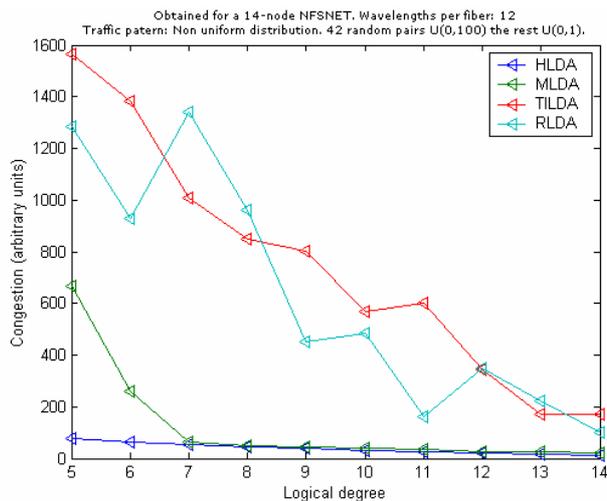


Figura 32. Curvas de congestión para los algoritmos heurísticos obtenidas con el programa trainer.m

4.4.2 Aplicación en GUI de Matlab: MatPlanWDM

El resultado de la combinación de algoritmos, topologías físicas, patrones de tráfico y funciones auxiliares es la aplicación en GUI (*Graphical User Interface*) de Matlab: **MatPlanWDM**. Esta aplicación incluye la funcionalidad de las rutinas programadas anteriormente *designer.m* y *trainer.m* e incluidas también juntos con los códigos del PFC. Por lo tanto, recogiendo la funcionalidad de ambos programas, el MatPlanWDM funciona en tres modos diferentes, que combinados ofrecen una caracterización completa de un algoritmo heurístico



Se ha querido hacer un diseño accesible y escalable. Las tres topologías físicas y los cinco patrones de tráfico son fácilmente ampliables con otros supuestos, y la inclusión de otros algoritmos heurísticos, para evaluarlos o compararlos con los ya programados, no presenta dificultad. Estamos pues, ante una herramienta ideal para funciones docentes que combina la facilidad del manejo, con el rigor en la simulación y obtención de resultados. Para este último objetivo se han provisto en el mismo entorno GUI las descripciones de las fichas de los cuatro algoritmos heurísticos, tanto en sus términos teóricos como en los aspectos prácticos de las funciones programadas.

Los resultados se presentan en un doble formato, mediante gráficas que permiten la comprensión global de la red obteniendo una idea intuitiva, y por medio de informes de texto. Estos últimos aportan la información fundamental de cada simulación. Se han extraído los indicadores más representativos, a la hora de evaluar un algoritmo, o el rendimiento de una topología virtual determinada.

4.4.3 Modos de funcionamiento

El MatPlanWDM incorpora la funcionalidad de los programas *designer.m* y *trainer.m*: diseño de topologías virtuales, evaluación de un algoritmo, comparación de todos los algoritmos entre sí. La combinación de estos tres modos de funcionamiento proporciona una caracterización fidedigna de un algoritmo heurístico. Cada una de las funciones, requiere distinto formato de datos de entrada, con variación en algunos campos, y por supuesto, distintos datos de salida. La selección del modo de funcionamiento se hace en el menú *Mode* de los menús desplegables de la parte superior.

- *Design Lightpath Logical Network*. Este modo incorpora la funcionalidad del programa *designer.m* y consiste en el diseño de la topología lógica, con un algoritmo heurístico, bajo condiciones fijas de topología física, patrón de tráfico, número de transmisores y receptores y longitudes de onda por enlace.
- *Evaluate Heuristic Algorithm*. Muchas características de los algoritmos no terminan por conocerse hasta que se somete a un escenario variable y se contempla su comportamiento. Esto precisamente es lo que hace este modo, con dos dimensiones de variación, respecto a la congestión. Este modo permite la simulación en un rango del

número de transmisores y receptores, y al mismo tiempo, en otro rango de número de longitudes de onda por fibra. Ambos intervalos se introducen en el cuadrante *Simulation Data*.

- *Compare Heuristic Algorithms*. Como sabemos existe variabilidad de los resultados de los algoritmos en función del escenario al que se apliquen. Para poder evaluar qué algoritmo se ajusta mejor a unas determinadas condiciones físicas y de tráfico se usa este modo, que compara en términos de congestión los resultados de los cuatro algoritmos.

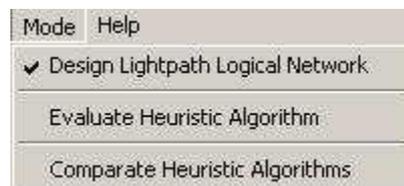


Figura 33. Modos de funcionamiento del MatPlanWDM

4.4.4 Componentes de la interfaz gráfica

En la Fig. 34 se presenta la interfaz gráfica del MatPlanWDM. Está dividida en cuatro cuadrantes, que tratan de dar independencia a la función que tiene asignada cada uno y permiten segregar los datos de entrada de los resultados de salida, ya sean gráficos o informes.



Figura 34. Interfaz gráfica del MatPlanWDM



Encima de los cuadrantes se especifica el título del modo de funcionamiento. El primer cuadrante (esquina superior izquierda) está destinado a la toma de datos para simulación, se presenta sombreado y resalta sobre el resto. Los tres cuadrantes restantes presentan los resultados, estando los dos inferiores destinados a la representación de gráficas y topologías; mientras que el segundo (esquina superior derecha) obtiene los informes y los resultados numéricos.

Los menús desplegables situados en la franja superior permiten conmutar entre los modos de funcionamiento, consultar los ficheros de ayuda o cerrar la aplicación.

4.4.4.8 Cuadrante *Simulation Data*

Esta sección de la interfaz es común a los tres modos de funcionamiento del MatPlanWDM, y permite introducir los datos de la simulación. Los campos utilizados y su formato dependen del modo de funcionamiento, así, mientras la topología física y los patrones de tráfico son comunes a los tres, el número de transmisores y receptores puede ser un rango, o los algoritmos heurísticos no se seleccionan en caso que haya que compararlos entre sí. En la Fig. 35 se pueden consultar los campos del cuadrante *Simulation Data*.

Simulation Data

Physical Topology:

Traffic Pattern:

Tx and Rx per node:

Wavelengths per fiber:

Heuristic Algorithm:

Design Reset

Figura 35. Campos del cuadrante *Simulation Data*

De los dos botones que dispone, el primero de ellos sólo se activa cuando todos los campos requeridos en la simulación están completos y en el formato de datos apropiados. Además, la etiqueta varía en función del modo en el opere la aplicación: *Design*, *Evaluate*, *Compare*. El



segundo botón *Reset*, pone todos los campos a cero y vuelve los menús desplegables a su posición original para introducir los datos de una nueva simulación. También borra las ventanas de resultados: gráficas e informes.

Los campos de la ventana *Simulation Data* son:

- *Physical Topology*. Topología física a elegir entre las tres presentadas en la sección 4.2.1.
 - SIMPLENET. Cuatro nodos.
 - POELTENET. Nueve nodos.
 - NSFNET. Catorce nodos.

- *Traffic Pattern*. Patrón que describe la demanda de tráfico a elegir entre los patrones introducidos en la sección 4.2.2.
 - Distribución uniforme $U(0,1)$.
 - Distribución uniforme $U(0,100)$.
 - Distribución no uniforme. 50 % $U(0,100)$ & 50 % $U(0,10)$
 - Distribución no uniforme. 25 % $U(0,100)$ & 75 % $U(0,10)$
 - Distribución no uniforme. Sólo para la NSFNET.

- *Tx and Rx per node*. Es el número de transmisores y receptores ópticos por nodo, también conocido como grado lógico de un nodo. Este valor limita fuertemente la topología virtual resultante. En el segundo y tercer modo de funcionamiento, el número de transmisores y receptores por nodo, es un rango. Su valor tiene que ser un número entero mayor que cero y en el caso de que se determine un intervalo, el límite superior mayor que el inferior. En caso contrario no se activa el botón de la simulación.

- *Wavelengths per fiber*. Este valor determina el número de longitudes de onda en cada enlace de fibra. Es un número entero mayor que cero.

- *Heuristic Algorithm*. En este menú desplegable se selecciona uno de los cuatro algoritmos heurísticos disponibles.
 - HLDA (*Heuristic Logical Design Algorithm*)



- MLDA (*Minimum-Delay Logical Design Algorithm*)
- TILDA (*Traffic Independent Logical Design Algorithm*)
- RLDA (*Random Logical Design Algorithm*)

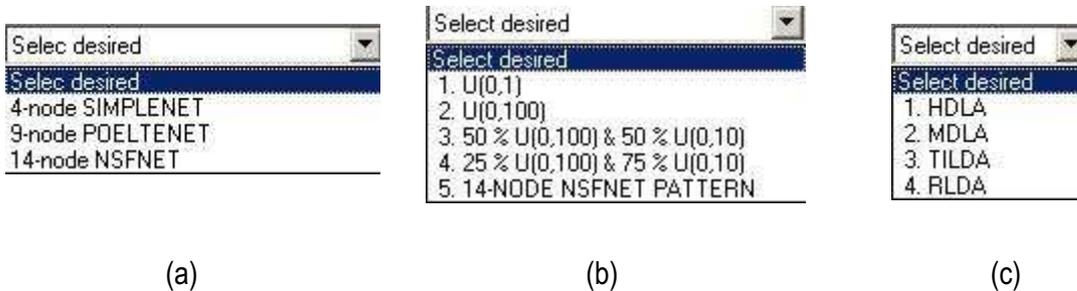


Figura 36. Menús desplegables en el cuadrante *Simulation Data*. (a) Topología física
(b) Patrón de tráfico (c) Algoritmo heurístico

4.4.4.9 Cuadrante *Physical Topology*

El tercer cuadrante (esquina inferior izquierda) de la interfaz gráfica lleva un componente *axis* para la representación de las topologías físicas en simulación mediante la función *plotTopology*. Para los tres de modos de funcionamiento este cuadrante invoca la misma función ya que la topología aparece en los tres.

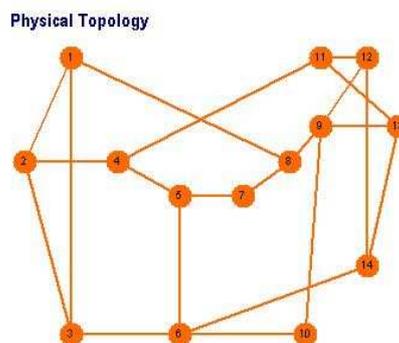


Figura 37. Segundo cuadrante de la interfaz gráfica. Topología física.

4.4.4.10 Cuadrantes *Simulations Results*

Los cuadrantes *Simulations Results* son el segundo (esquina superior derecha) y el cuarto (esquina inferior derecha) de la interfaz y en los dos se representan los resultados de la simulación, con el informe y de forma gráfica respectivamente.



Los contenidos del informe de texto varían según el modo de funcionamiento que opera la aplicación. Por ejemplo, en caso de que sean los resultados relativos al diseño de la topología virtual, el informe recoge las rutas físicas de *lightpaths* establecidos y la longitud de onda usada. Sin embargo, los resultados relativos a las pruebas comparativas entre los algoritmos, extraen el número de simulaciones, los rangos y los tiempos. De forma similar el cuarto cuadrante muestra los resultados gráficos según el modo de funcionamiento. En el primer modo, esta ventana dibuja la topología virtual (red de *lightpaths*) y en el segundo y tercer modo se obtienen las curvas de congestión. En el primer supuesto se proporciona una leyenda relativa a los nodos, los enlaces físicos y los *lightpaths*.

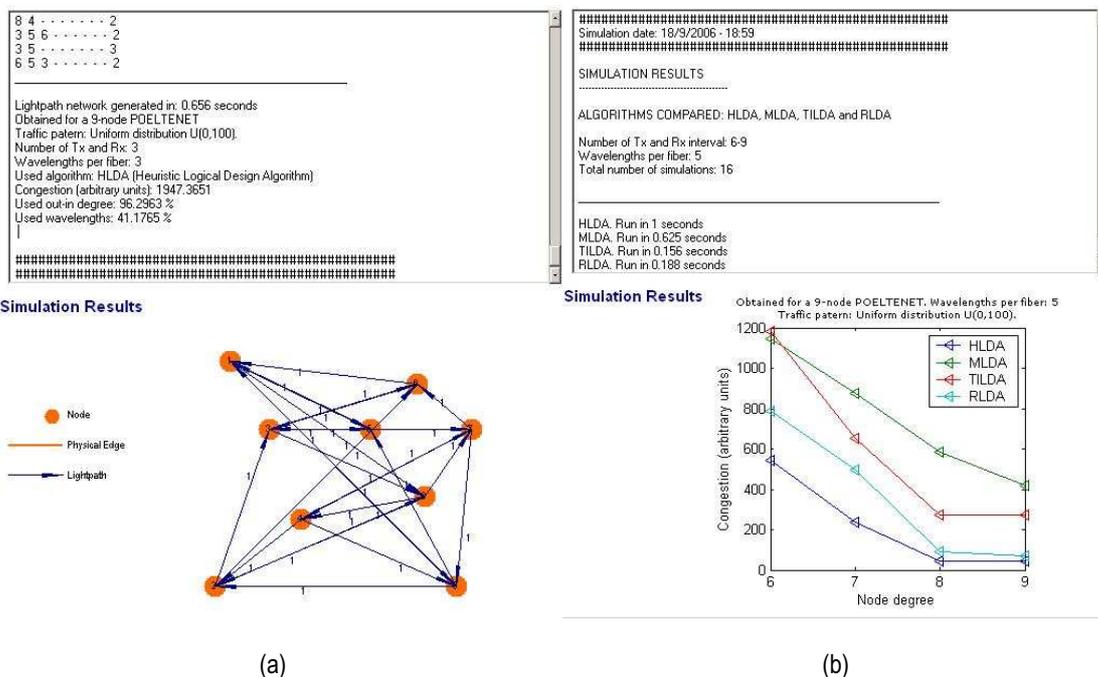


Figura 38. Resultados presentados en el segundo y cuarto cuadrante de la interfaz del MatPlanWDM. (a) Resultado del diseño de una topología virtual. (b) Resultados de la comparación de los algoritmos

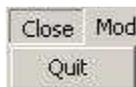
4.4.4.11 Menús desplegables

En la franja superior de la interfaz, se ubican los menús de selección desplegables que posibilitan la conmutación de modos de operación, ofrecen los ficheros de ayuda o cierran la aplicación.

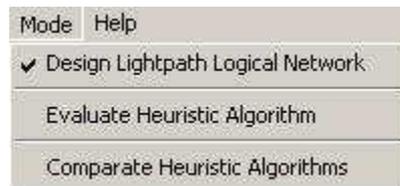
- *Close*. Este menú lleva sólo una opción, *Quit*, para el cierre de ventana de la interfaz y de aplicación la MatPlanWDM.



- *Mode*. Tiene tres opciones, clasificadas por separadores, correspondiendo cada una de ellas y en orden, a los modos de funcionamiento del programa.
 - *Design Lightpath Logical Network*
 - *Evaluate Heuristic Algorithm*
 - *Compare Heuristic Algorithms*
- *Help*. Este menú ofrece los ficheros de ayuda de cada algoritmo heurístico. Los ficheros de ayuda invocan al navegador web predeterminado para mostrar una página en formato htm. Los ficheros de ayuda que se proporcionan son hldav1.htm, mllda.htm, tilda.htm y rlda.htm. La última opción llama a una ventana flotante con los créditos de la aplicación. Un ejemplo de ambas funciones se puede consultar en la Fig. 40.
 - HLDA
 - MLDA
 - TILDA
 - RLDA
 - *About this tool*



(a)



(b)



(c)

Figura 39. Menus desplegables. (a) *Close* (b) *Mode* (c) *Help*

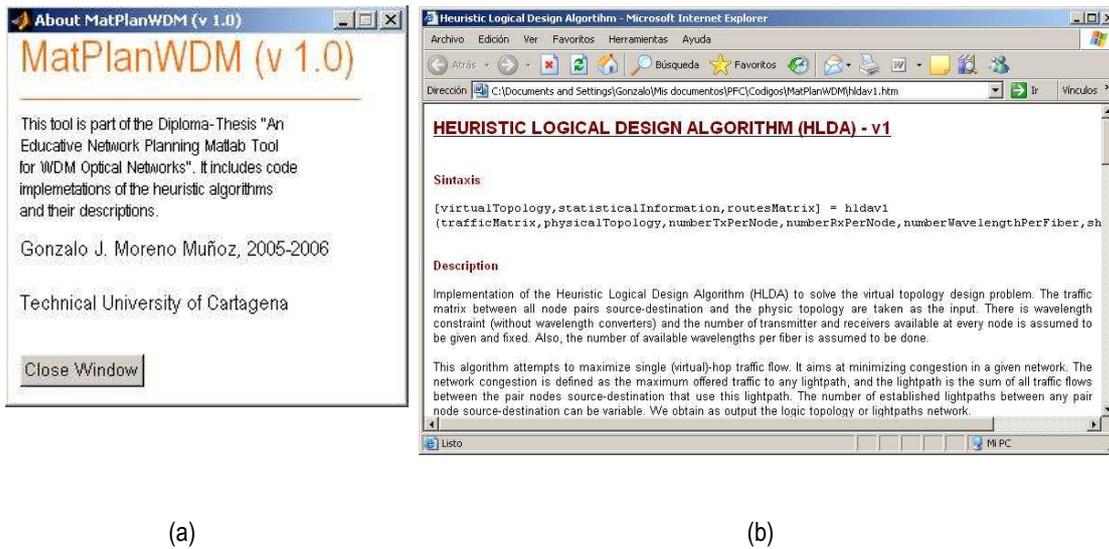


Figura 40. (a) Ventana flotante de créditos del MatPlanWDM. (b) Fichero web de ayuda del HLDA

4.4.5 Resultados del MatPlanWDM. Informes y gráficas.

Como se ha ido comentando, tanto la adquisición de datos como los resultados obtenidos, varían en función del modo de operación. En cualquier caso se proporciona un informe minucioso sobre los resultados y gráficas que lo apoyan de forma intuitiva. En las Fig. 42, 44 y 46 se pueden ver ejemplos de los resultados de la aplicación en todos los modos de funcionamiento.

4.4.5.1 Modo 1. Diseño de topologías virtuales

Cuando se resuelve el problema de diseño de la topología virtual, varios aspectos se deben tener en cuenta a la hora de presentar sus resultados y rendimiento. Para ello es fundamental la obtención gráfica de la topología, o de forma similar, la red de *lightpaths*. Para ello se utiliza la función `plotTopology` que nos permite representar los *lightpath* mediante flechas azules. Cada *lightpath* lleva asociado un pequeño número que indica el número de *lightpaths* establecidos para el mismo par de nodos. Los nodos se representan numerados a partir del 1 y en naranja. En la Fig. 41 se puede ver una red de *lightpaths* obtenida como resultado de resolver el problema de diseño de la topología virtual para una red POELTENET de nueve nodos.

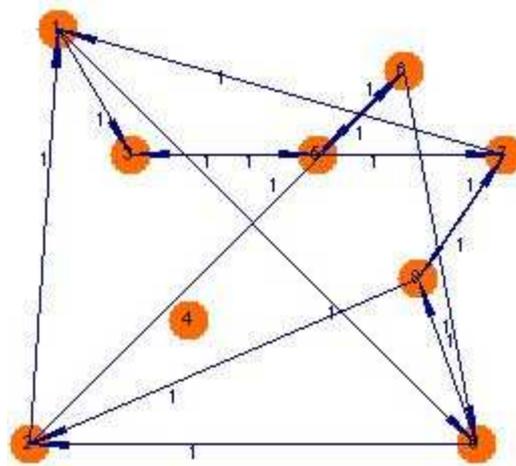


Figura 41. Red de *lightpaths*. Resultado gráfico del primer modo de funcionamiento.

Junto con la red de *lightpath* existen una serie de datos esenciales para caracterizar el algoritmo. Estos datos se proporcionan en el informe de resultados, que contiene:

- Fecha y hora de la simulación.
- Matriz de tráfico generada.
- Rutas físicas para cada *lightpath*.
- Índice de la longitud de onda para cada *lightpath*.
- Tiempo requerido para la obtención de la topología virtual.
- Topología física, patrón de tráfico, número de transceptores por nodo, número de longitudes de onda por fibra y algoritmo heurístico utilizado.
- Congestión de la red.
- Porcentaje de transceptores usados (promediados entre todos los nodos).
- Porcentaje de longitudes de onda usadas (promediadas entre todos los enlaces).

A continuación se muestra el informe de resultados de la topología virtual de la Fig. 41. Los dos formatos de resultados son esenciales para poder comprender la topología virtual, por ejemplo, considerar el *lightpath* establecido desde el nodo 3 al nodo 7. Para conocer la ruta física y la longitud de onda que usa, es necesario consultar el informe. Ese *lightpath* atraviesa los nodos 2 y 9 en su trayecto, y utiliza el primer índice de longitud de onda: ω_0 .



```
#####
Simulation date: 18/9/2006 - 19:46
#####

VIRTUAL TOPOLOGY RESULTS
-----

Generated traffic matrix:

  0   67   73   25   72   24   34   63   71
76   0   22   9.2  3.5  94   31   39   65
90   84   0   60   98   8.9  95   85   45
42   53   0.7   0   50   54   54   53   17
66   26   32   29   0   90   70   81   84
0.13  16   74   47   80   0   88   39   97
98   56   79   4.6  52   63   0   96   13
41   96   40   8.6  50   26   89   0   25
35   95   40   51   75   38   40   95   0

Physical routes and used wavelengths:
-----
7 6 1 - - - - - 1
3 5 - - - - - 1
6 8 9 - - - - - 1
7 8 - - - - - 1
8 9 2 - - - - - 2
9 8 - - - - - 1
9 2 - - - - - 1
3 2 9 7 - - - - - 1
2 1 6 - - - - - 1
5 6 - - - - - 1
8 7 - - - - - 1
6 5 - - - - - 1
2 1 - - - - - 2
1 3 - - - - - 1
1 2 9 - - - - - 2
5 3 - - - - - 1
-----

Lightpath network generated in: 0.406 seconds
Obtained for a 9-node POELTENET
Traffic pattern: Uniform distribution U(0,100).
Number of Tx and Rx: 2
Wavelengths per fiber: 2
Used algorithm: HLDA (Heuristic Logical Design Algorithm)
Congestion (arbitrary units): 2507.2962
Used out-in degree: 88.8889 %
Used wavelengths: 33.8235 %

#####
#####
```

En la Fig. 42 se puede contemplar el aspecto de la interfaz del MatPlanWDM cuando presenta los resultados del problema de la topología virtual.

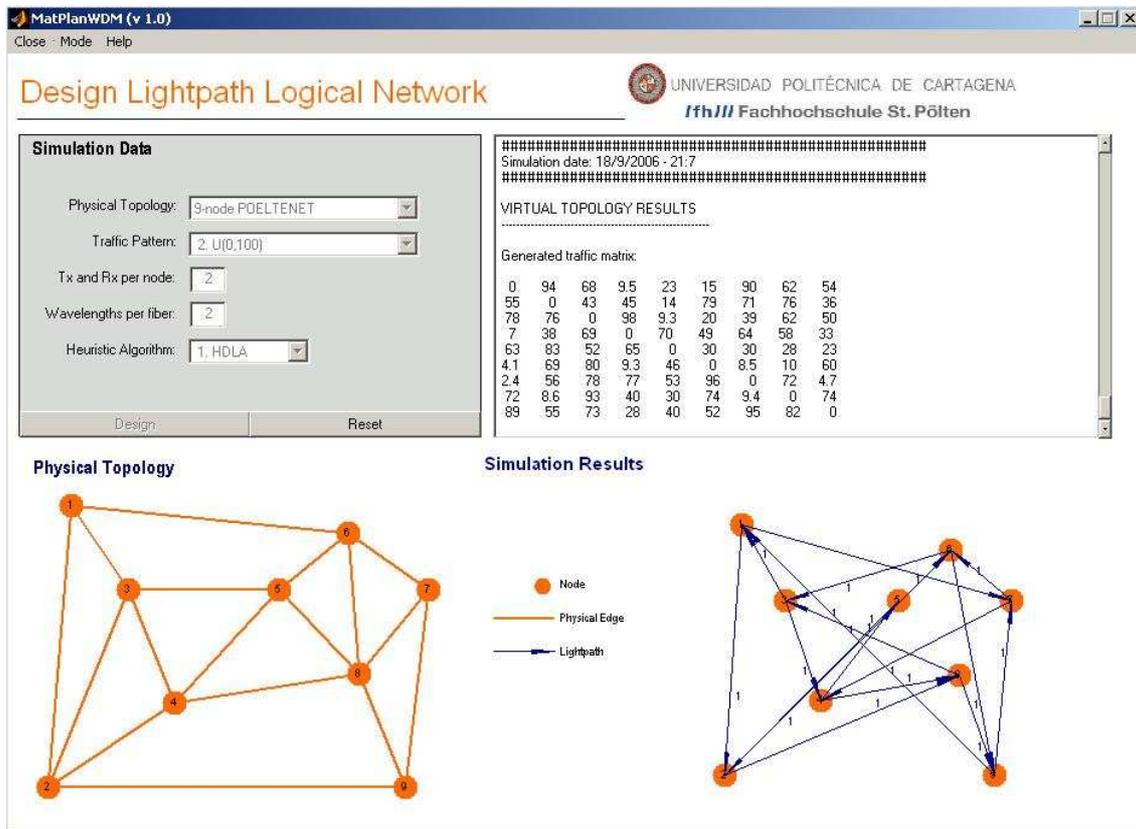


Figura 42. Interfaz gráfica al diseñar una topología virtual

4.4.5.2 Modo 2. Evaluación de un algoritmo heurístico

Si en el primer modo la importancia de los dos formatos de los resultados, se complementaba, en este segundo modo, la información esencial recae en la representación gráfica, sin menoscabo de datos aportados por el informe de resultados.

La representación gráfica corresponde a las curvas de congestión. Se trata de unos ejes de dos dimensiones cuyo eje de ordenadas, toma los valores de la congestión de la red en unidades arbitrarias, y el eje de abscisas es creciente según el número de transceptores (transmisores y receptores por nodos). Adicionalmente, se dibuja, en distinto color, una curva por cada número de longitudes de onda por fibra. En la Fig. 43 se puede ver la representación gráfica de las curvas de congestión para distintos valores de número de longitudes de onda por fibra.

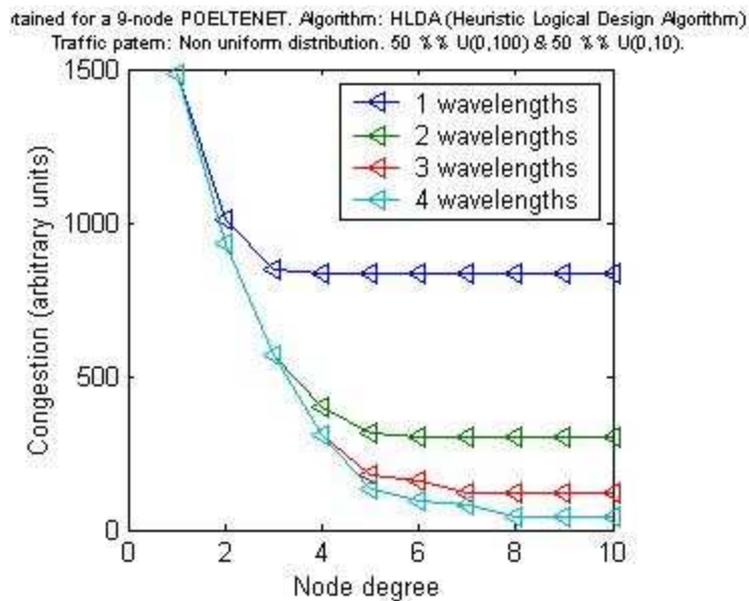


Figura 43. Curvas de congestión según longitudes de onda

El informe de resultados incorpora una serie de datos útiles desde un punto de vista práctico. A continuación se enumeran sus campos:

- Fecha y hora de la simulación.
- Matriz de tráfico generada.
- Algoritmo heurístico evaluado, patrón de tráfico.
- Matriz de tráfico generada.
- Intervalo del número de transceptores.
- Intervalo del número de longitudes de onda por fibra.
- Número de simulaciones. Resultado del producto de los dos datos anteriores.
- Tiempo total de la simulación.

La simulación realizada en la Fig. 43 correspondía a una topología POELTENET de nueve nodos con un patrón no uniforme de tráfico. Se simula un rango de transceptores desde 1 hasta 10, y se presentan cuatro curvas de congestión (longitudes de onda de 1 a 4). El informe de resultado obtenido es el siguiente:



```
#####
Simulation date: 18/9/2006 - 20:47
#####

SIMULATION RESULTS
-----

EVALUATED ALGORITHM: HLDA (Heuristic Logical Design Algorithm)

Obtained for a 9-node POELTENET
Traffic pattern: Non uniform distribution. 50 %% U(0,100) & 50 %% U(0,10).

Generated traffic matrix:
  0  24  55  98  6.4  6.6  5.8  98  77
 40  0  0.68  6.8  1.7  62  83  69  40
0.56  9.1  0  7.4  85  5.1  9.7  2.3  7.1
 3.8  3.1  14  0  78  2.9  69  35  99
 2.6  95  2.6  44  0  67  5.1  1e+002  34
 6.6  7.2  23  9.3  25  0  27  2.4  9.8
 7  3.8  85  6.1  2.7  4.2  0  60  75
0.91  9  66  7.4  8.1  9.6  42  0  4.6
 9.5  15  6.2  42  4  12  8.6  9  0

Number of Tx and Rx interval: 1-10
Number of wavelengths per fiber interval: 1-4
Total number of simulations: 40
Run in: 7.765 seconds
#####
```

En la Fig. 44 se puede ver la ventana de la interfaz gráfica cuando se presentan los resultados del segundo modo de funcionamiento: evaluación de un algoritmo heurístico.

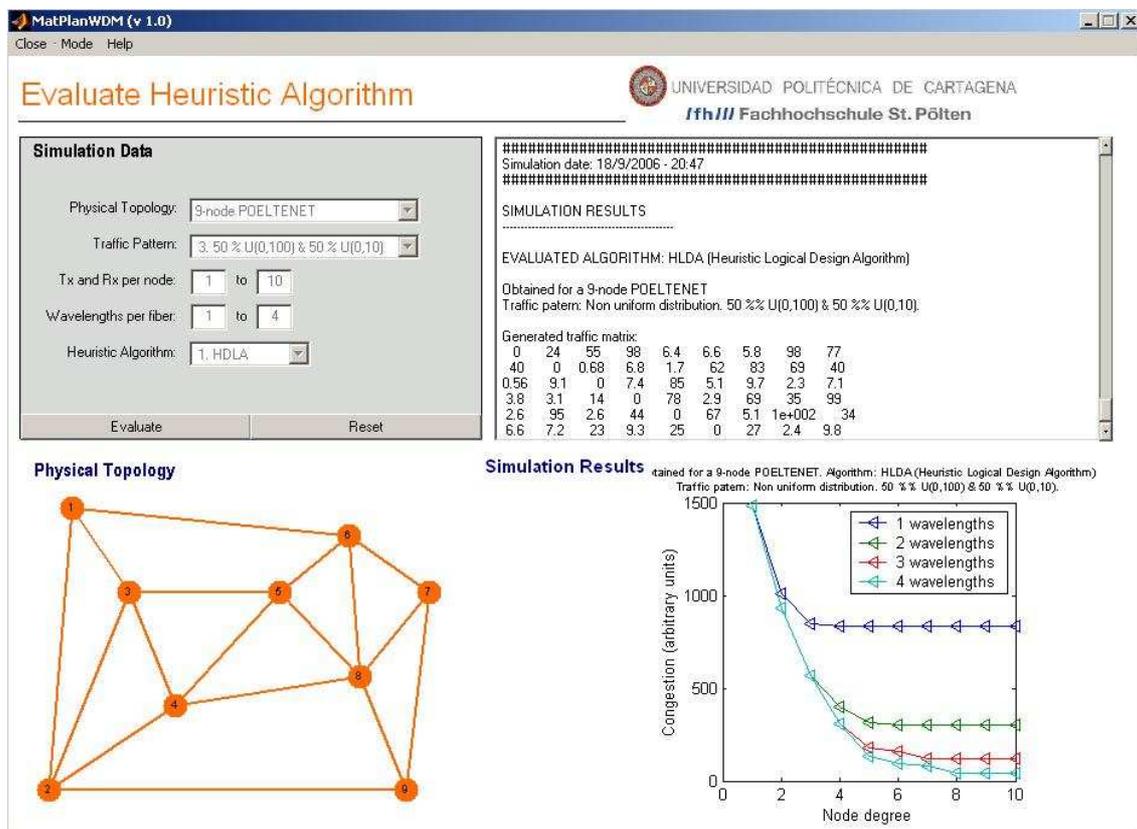


Figura 44. Interfaz gráfica al evaluar un algoritmo heurístico

4.4.5.3 Modo 3. Comparación de algoritmos heurísticos

De forma similar al modo anterior, la importancia en la evaluación de los resultados, radica en la representación gráfica. Se trata igualmente de curvas de congestión, con la salvedad, de que ahora las curvas son en función de los cuatro algoritmos.

La representación gráfica tiene la congestión como variable de ordenadas y el número de transceptores en abscisas. Se dibujan cuatro curvas de colores: azul, verde, rojo y cian para representar respectivamente los algoritmos HLDA, MLDA, TILDA y RLDA. En la Fig. 45 se proporciona un ejemplo de los ejes descritos.

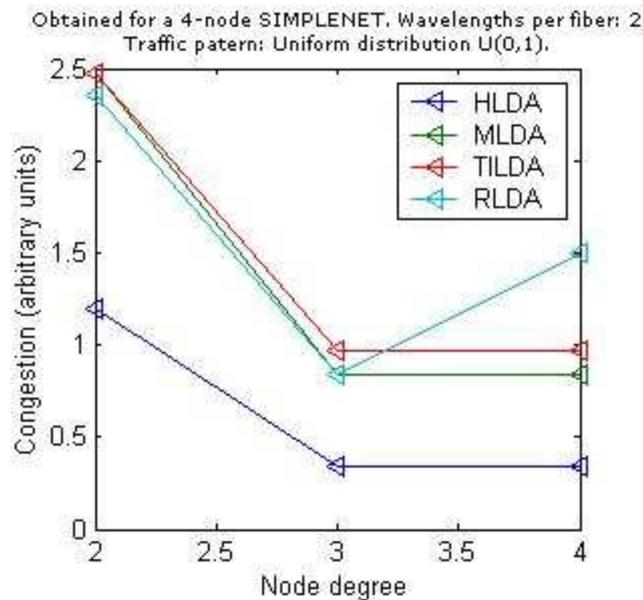


Figura 45. Representación gráfica de las curvas de congestión

El informe de resultados aporta información comparativa para conocer la velocidad de los algoritmos, aparte de los datos de entrada de la simulación. Sus campos son:

- Fecha y hora de la simulación.
- Algoritmos heurísticos comparados.
- Topología física, patrón de tráfico.
- Matriz de tráfico generada.
- Intervalo del número de transceptores.
- Número de longitudes de onda por fibra.
- Número total de simulaciones.



➤ Tiempos de ejecución para cada algoritmo.

Siguiendo la representación gráfica de la Fig. 45 a continuación se presenta el informe de resultados. Se trata de un topología SIMPLENET de cuatro nodos, tráfico uniforme $U(0,1)$, con un intervalo de transceptores de 2 a 4 y con número de longitud de onda fijo e igual a 2.

```
#####  
Simulation date: 18/9/2006 - 21:10  
#####  
  
SIMULATION RESULTS  
-----  
  
ALGORITHMS COMPARED: HLDA, MLDA, TILDA and RLDA  
  
Obtained for a 4-node SIMPLENET  
Traffic pattern: Uniform distribution U(0,1).  
  
Generated traffic matrix:  
  0 0.075 0.71 0.54  
0.69 0 0.96 0.33  
0.44 0.68 0 0.046  
0.3 0.47 0.45 0  
  
Number of Tx and Rx interval: 2-4  
Wavelengths per fiber: 2  
Total number of simulations: 12  
  
-----  
  
HLDA. Run in 0.109 seconds  
MLDA. Run in 0.078 seconds  
TILDA. Run in 0.063 seconds  
RLDA. Run in 0.062 seconds  
  
#####  
#####
```

Para obtener una visión general de los resultados obtenidos en el tercer modo de funcionamiento, dentro del conjunto de la interfaz, se proporciona la Fig. 46.

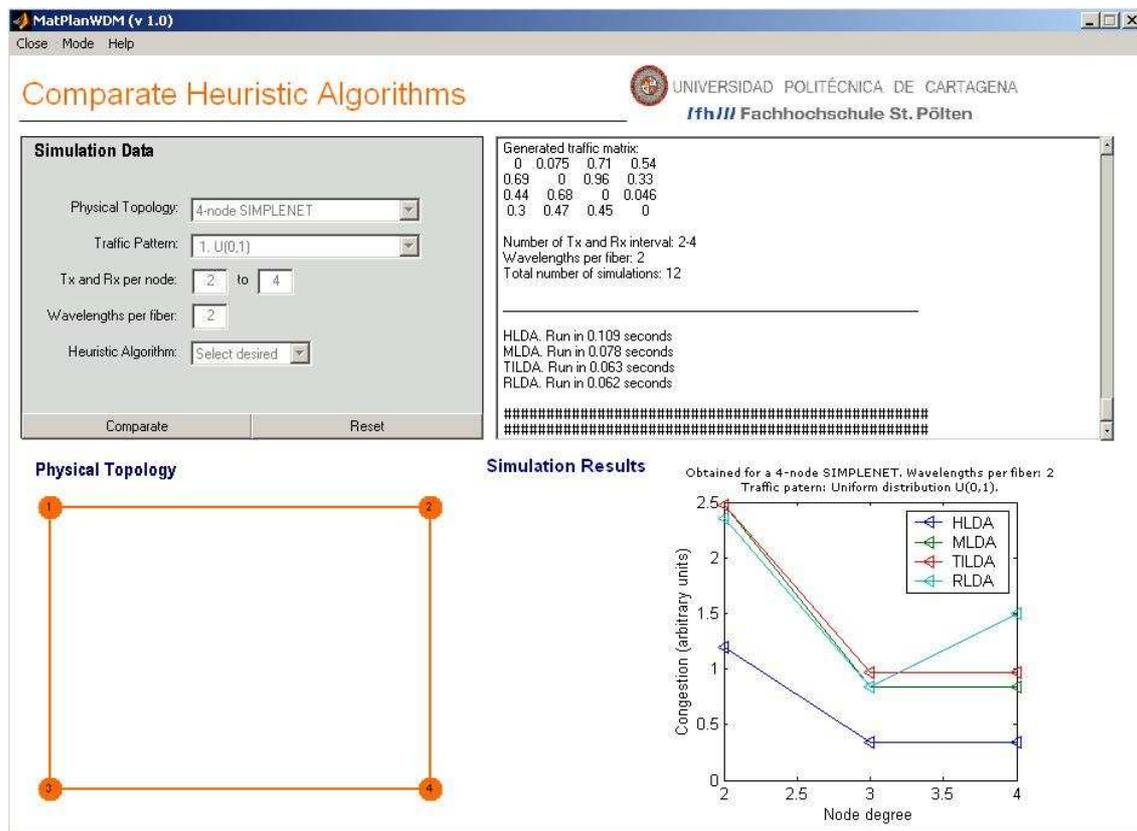


Figura 46. Interfaz gráfica al comparar los algoritmos heurísticos



CAPÍTULO 5

Resultados

5.1 TOPOLOGÍAS VIRTUALES

5.1.1 Ejemplo 1. SIMPLENET 4 nodos. Patrón uniforme. HLDA y MLDA.

Como primer ejemplo de diseño de topologías virtuales, y comprobar el correcto funcionamiento de los algoritmos HLDA y MLDA, ejecutamos el desarrollo teórico para estos algoritmos propuesto en las secciones 3.2.1 y 3.2.2. La comparativa de los resultados desarrollados a mano y los obtenidos por nuestros programas es una manera introductoria de comprender el mecanismo del diseño de las topologías virtuales y el funcionamiento de los algoritmos.

Para este ejemplo no se usa ninguna de nuestras aplicaciones sino que se programa una rutina específica con los datos de entrada necesarios. El nombre de este programa es ejemplo1.m y se encuentra en el directorio de trabajo. A continuación se muestra la invocación.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%EJEMPLO I (a) (HLDAv1)  
  
%Input variables  
trafficMatrix=[0 0.9 0.3 0.15;0.1 0 0.8 0.5;0.45 0.6 0 0.35;0.4 0.1 0.75 0];  
physicTopology = [0 1 1 0;1 0 0 1;1 0 0 1;0 1 1 0];  
txPerNode =2;  
rxPerNode =2;  
wavelengthPerFiber =2;  
nodesPlaceMatrix=[-1 1;1 1;-1 -1;1 -1];  
  
%Function calling  
[virtualTopology] =  
hldav1(trafficMatrix,physicTopology,txPerNode,rxPerNode,wavelengthPerFiber,0)  
  
%Graphical representation  
figure('Color',[1 1 1])  
hold on
```



```

subplot(2,1,1), plotTopology(nodesPlaceMatrix,physicTopology,0,'Physical Topology 4-node
SIMPLENET') %Physical topology is plotted
subplot(2,1,2), plotTopology(nodesPlaceMatrix,virtualTopology,1,'Topologia virtual.
Ejemplo 1(a). HLDA') %Virtual topology is plotted
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%EJEMPLO I (b) (MLDA)

%Input variables
trafficMatrix=[0 0.9 0.3 0.15;0.1 0 0.8 0.5;0.45 0.6 0 0.35;0.4 0.1 0.75 0];
physicTopology = [0 1 1 0;1 0 0 1;1 0 0 1;0 1 1 0];
txPerNode =3;
rxPerNode =3;
wavelengthPerFiber =2;
nodesPlaceMatrix=[-1 1;1 1;-1 -1;1 -1];
%Function calling
[virtualTopology] =
mlda(trafficMatrix,physicTopology,txPerNode,rxPerNode,wavelengthPerFiber,0)
%Graphical representation
figure('Color',[1 1 1])
hold on
subplot(2,1,1), plotTopology(nodesPlaceMatrix,physicTopology,0,'Physical Topology 4-node
SIMPLENET') %Physical topology is plotted
subplot(2,1,2), plotTopology(nodesPlaceMatrix,virtualTopology,1,'Topologia virtual.
Ejemplo 1(b). MLDA') %Virtual topology is plotted
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    
```

En las Fig. 47 (b) y (c) se puede ver la analogía con las topologías virtuales obtenidas de forma teórica en las Fig. 21 (b) y 24 (b). Para profundizar en el comportamiento de ambos algoritmos, a la hora de diseñar la topología virtual, consultar el ejemplo teórico al que hacemos referencia.

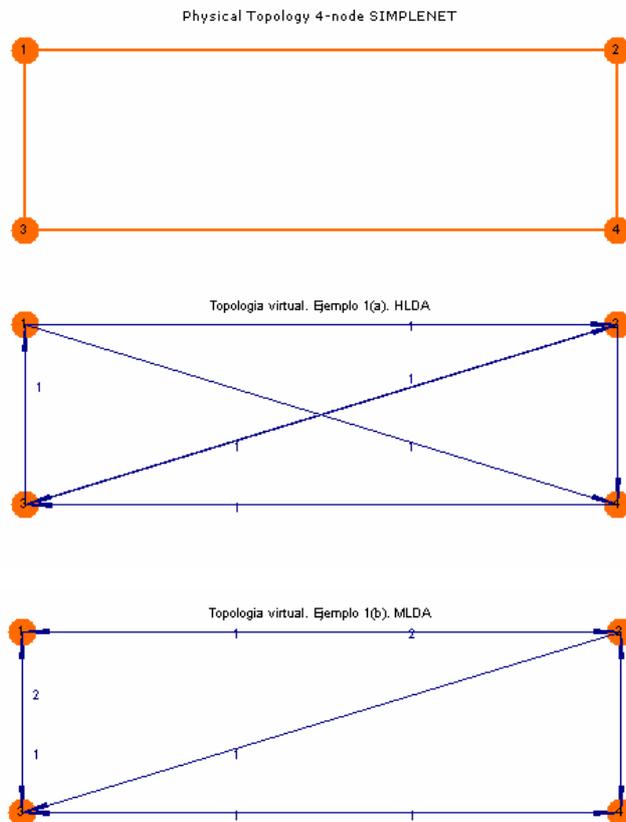


Figura 47. Topologías físicas y lógicas del Ejemplo 1. SIMPLENET de 4 nodos. Patrón uniforme. HLDA y MLDA



Con respecto al MLDA conviene notar que es el único algoritmo que establece más de un *lightpath* por par de nodos. El resto de algoritmos pueden hacerlo de forma teórica pero la disponibilidad de recursos no lo permite, porque establecer un *lightpath* por segunda vez supone haber procesado previamente todos los pares existentes en la red. El MLDA no exige esta condición.

A continuación se muestran las tablas de enrutamiento para *lightpaths* obtenidas de la variable `routesMatrix` para ambos casos. De forma similar a las topologías lógicas, se pueden comparar con los ejemplos de las secciones 3.2.1 y 3.2.2.

EJEMPLO I (a) .HLDA				
Ruta física			Longitud de onda	
1	2	-	-	1
2	4	3	-	1
4	3	-	-	2
3	4	2	-	1
2	4	-	-	2
3	1	-	-	1
1	3	4	-	2

EJEMPLO I (b) .MLDA				
Ruta física			Longitud de onda	
2	1	-	-	1
3	1	-	-	1
1	2	-	-	1
4	2	-	-	1
1	3	-	-	1
4	3	-	-	1
2	4	-	-	1
3	4	-	-	1
1	2	-	-	2
2	4	3	-	2
3	1	-	-	2

5.1.2 Ejemplo 2. POELTENET 9 nodos. TILDA.

A continuación se presenta un ejemplo más complejo, donde el número de nodos se aproxima a los escenarios reales de redes WAN y se aplica para su resolución el algoritmo TILDA. Recordamos que este algoritmo trabaja de forma independiente a la matriz de tráfico, por lo que no se proporciona el patrón.

Igual que en el ejemplo anterior se programa la rutina `ejemplo2.m` para el diseño de la topología virtual.



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%EJMEPLO II (TILDA)

%Input variables
physicalTopology = [0 1 1 0 0 1 0 0 0;1 0 1 1 0 0 0 0 1;1 1 0 1 1 0 0 0 0;0 1 1 0 1 0 0
1 0;0 0 1 1 0 1 0 1 0;1 0 0 0 1 0 1 1 0;0 0 0 0 0 1 0 1 1;0 0 0 1 1 1 1 0 1;0 1 0 0 0 0
1 1 0];
txPerNode =4
rxPerNode =4
wavelengthPerFiber= 5
nodesPlaceMatrix=[-12 7;-14 -3;-7 4;-3 0;6 4;12 6;19 4;13 1;17 -3];

%Function calling
[virtualTopology,statisticalInformation,routesMatrix] = tilda
(physicalTopology,txPerNode,rxPerNode,wavelengthPerFiber,0)

%Graphical representation
figure('Color',[1 1 1])
hold on
subplot(2,1,1), plotTopology(nodesPlaceMatrix,physicTopology,0,'Physical Topology 4-node
SIMPLENET') %Physical topology is plotted
subplot(2,1,2), plotTopology(nodesPlaceMatrix,virtualTopology,1,'Topologia virtual.
Ejemplo 2. TILDA') %Virtual topology is plotted
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    
```

La reducción de la información *a priori* del problema, como es la ausencia de la matriz de tráfico, acarrea, como es natural, un menor rendimiento de la red. Por otra parte, como ventaja, se puede abordar el problema sin un dato fundamental en otros algoritmos heurísticos.

La tendencia en todos los diseños es a usar un porcentaje bajo de canales de longitudes de onda, siempre por debajo de la utilización de los transceptores. Pero en este caso esa diferencia se acusa aún más, porque el diseño de topología virtual obedece exclusivamente a criterios de red física. Los porcentajes de utilización son:

- Uso de transceptores: 94.4444 %
- Uso de longitudes de onda: 21.1765 %

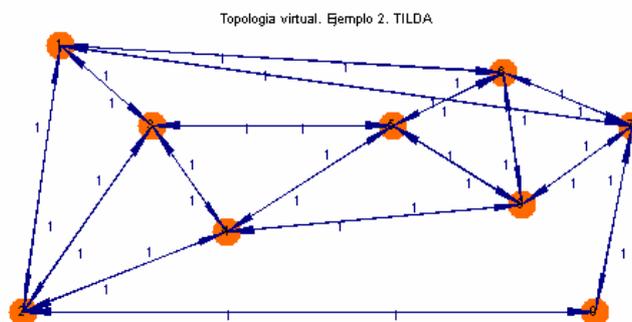


Figura 48. Topología virtual Ejemplo2. POELTENET de 9 nodos. TILDA

5.1.3 Ejemplo 3. NSFNET 14 nodos. Patrón propio. RLDA.

Los datos seleccionados para este ejemplo resultan contradictorios, porque se escoge la topología más compleja (14 nodos) con el patrón propio de la red NSFNET, descrito en el apartado 4.2.2. y para obtener la topología virtual, se aplica el algoritmo que, en teoría, presenta resultados más pobres. El patrón propio de la NSFNET tiene 42 pares cargados dos órdenes de magnitud por encima que el resto, por lo que una asignación aleatoria de *lightpaths* entre los $14 \times 14 - 14 = 182$ pares de nodos, no parece la solución óptima. Más bien sería necesario un algoritmo que evaluara los pares, en orden descendente de tráfico (HLDA y MLDA). Aun así es posible hacer otra lectura de este ejemplo, porque la simplicidad del RLDA lo hace óptimo en términos de velocidad, variable a tener muy en cuenta en diseños de cierto tamaño. Por lo tanto en una red compleja y con requerimientos *on-line*, es decir reconfiguración instantánea de la red de *lightpath*, un ejemplo así sería viable.

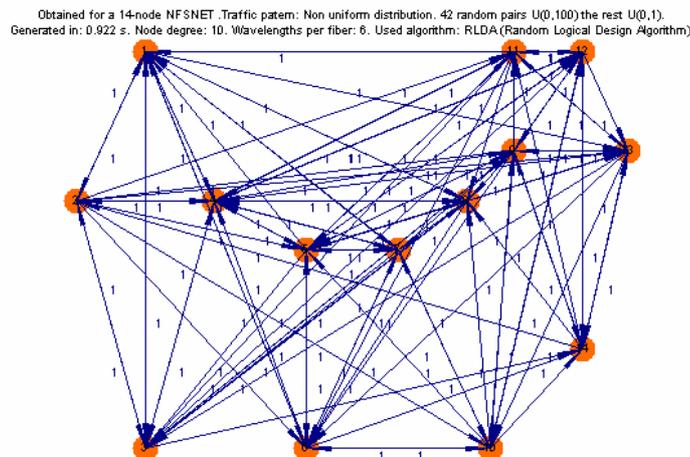


Figura 49. Topología virtual del Ejemplo 3. NSFNET de 14 nodos. RLDA.

5.2 EVALUACIÓN DE ALGORITMOS

5.2.1 HLDA

El algoritmo HLDA es el que presenta un comportamiento más lineal de los cuatro algoritmos estudiados. Y no porque exista una proporcionalidad directa en todo el espacio de simulación,



sino porque su comportamiento es conocido y lógico en determinados segmentos del espacio y presenta características muy interesantes. Quizá su mejor calificativo, privativo respecto al resto, es estabilidad. A continuación se presenta el informe de resultados y las curvas de congestión para tres longitudes de onda diferentes. La topología simulada es la POELTENET de 9 nodos con un patrón de tráfico uniforme.

```
#####
SIMULATION RESULTS 19/9/2006
-----

Used algorithm: HLDA (Heuristic Logical Design Algorithm)
Number of simulations: 40. Run in 8.625 seconds

Obtained for a 9-node POELTENET
Traffic pattern: Uniform distribution U(0,100).
Node degree interval between 1 and 10 (10 measurements)
Numbers of wavelengths per fiber simulated: 3 4 5 6

#####
St. Poelten University of Applied Sciences
Gonzalo J. Moreno Muñoz, 2006
#####
```

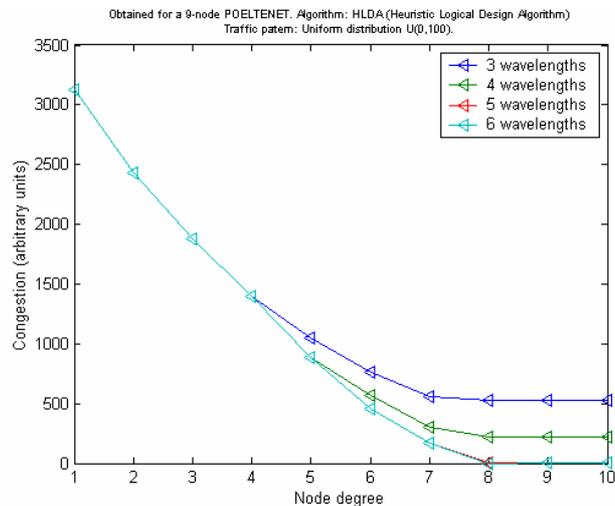


Figura 50. Curvas de congestión en la caracterización del HLDA. Escenario 1.

En la gráfica anterior se pueden contemplar dos etapas: una lineal y otra amortiguada. La etapa lineal se corresponde con un número bajo de transceptores, donde estos limitan el rendimiento de la red independientemente del número de longitudes de onda que se usen. Aproximadamente cuando se alcanza el valor máximo del grado físico de la red (número de enlaces físicos conectados a cada nodo), las curvas se separan para entrar en la segunda etapa. Esta etapa invierte los conceptos de la primera, porque ahora es el número de longitudes de



onda por fibra la variable que limita los resultados de la congestión, independientemente del número de transceptores presentes por nodo. En esta dinámica, y claramente a partir de ocho transceptores por nodo, las curvas se estabilizan en un nivel de congestión constante dependiente de las longitudes de onda, y que ya no variará con sucesivos incrementos de los transceptores.

Para remarcar este carácter de estabilidad, donde el mismo patrón de congestión se repite con independencia de la topología y de la matriz de tráfico, dato muy importante, se ofrecen en la siguiente figura, las curvas de congestión de una simulación completamente diferente: NSFNET de 14 nodos con su patrón propio.

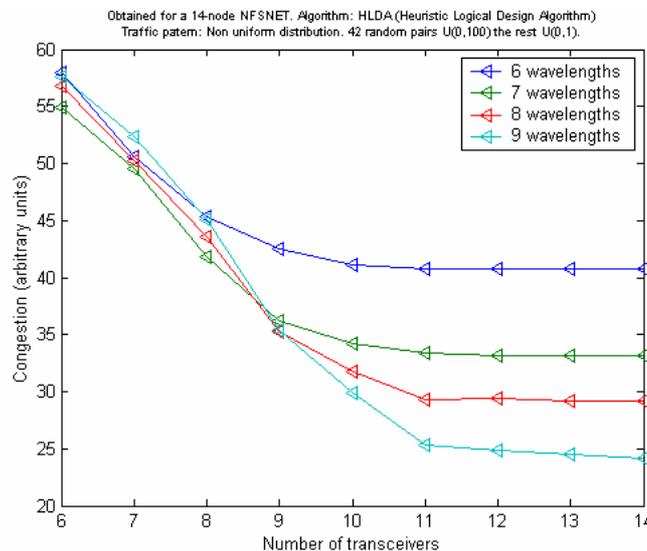


Figura 51. Curvas de congestión en la caracterización del HLDA. Escenario 2.

De forma similar al caso anterior se observa la caída lineal de las curvas en la primera etapa, pasando a la estabilidad respecto al número de transceptores, en la segunda.

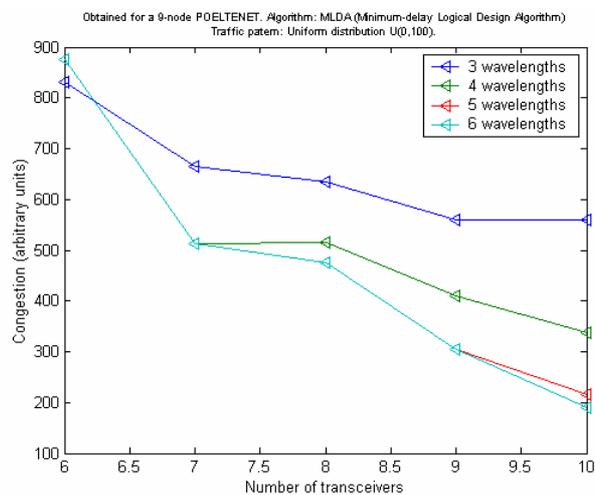
5.2.2 MLDA

Lo dicho en el apartado anterior para el HLDA adquiere ahora su verdadera importancia, porque como veremos, los resultados del MLDA no se ajustan tanto a un patrón determinado, dado sus características de establecer segundos canales lógicos, entre el mismo par de nodos sin haber considerado el resto de pares. El funcionamiento del MLDA está indicado para arquitecturas, donde exista una carga evidente con respecto al resto, en determinados pares de nodos. Sólo en ese caso estarán justificados los sucesivos *lightpaths*. Esto se evidencia

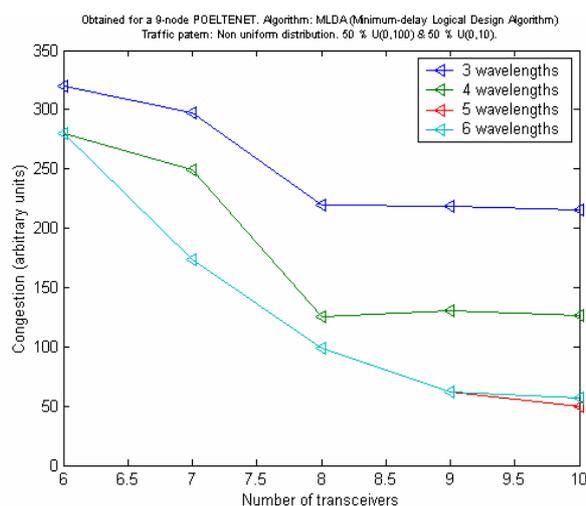


simulando el MLDA en una misma topología física (POENTENET de nueve nodos) pero para patrones crecientes en no-uniformidad. Se simulan estos tres:

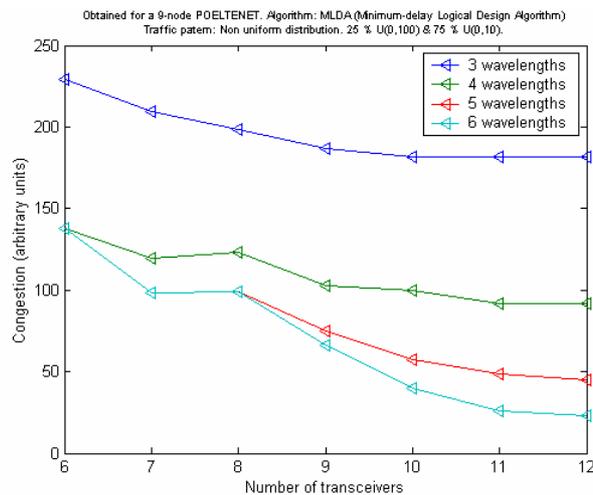
- Distribución uniforme $U(0,100)$
- Distribución no uniforme. 50 % $U(0,100)$ & 50 % $U(0,10)$
- Distribución no uniforme. 25 % $U(0,100)$ & 75 % $U(0,10)$



(a)



(b)



(c)

Figura 52. Curvas de congestión para la caracterización del MLDA. (a) Patrón uniforme U(0,100) (b) Patrón no-uniforme 50 % (c) Patrón no-uniforme 25 %

En la primera simulación no se observa un comportamiento claro, dentro de la tendencia decreciente, ya que un patrón uniforme no requiere sobrecarga de *lightpaths*, y por lo tanto se establecerán muchos de ellos de forma innecesaria y con la consabida ocupación de recursos de red. La estructura de las curvas de congestión en la segunda gráfica ya parece apuntar a la que presentaba el HLDA, si bien no es posible marcar una frontera en el comportamiento de las curvas y todavía sus variaciones presentan cierta brusquedad. Cuando existe un conjunto reducido de pares cargados respecto al resto, es cuando el MLDA asemeja más sus resultados al comportamiento del HLDA. Como se puede observar la tendencia apuntada en la segunda gráfica se confirma, presentando en la tercera figura -Fig 52 (c)-, las curvas ese contorno más suavizado y amortiguado, al tiempo que se alcanza la estabilidad para cierto valor del número de transceptores.

5.2.3 TILDA y RLDA

Los dos últimos algoritmos se caracterizan de forma conjunta, ya que comparten el diseño de la topología virtual sin tener en cuenta la matriz de tráfico. Esto resulta útil en determinados problemas de optimización, cuando no se dispone de toda la información de entrada para invocar un algoritmo más elaborado. Sin embargo, y pesa a presentar resultados similares globales, el comportamiento de uno y otro marca una diferencia importante en cuanto a estabilidad y



fiabilidad. Se simula para ambos el mismo escenario, topología POELTENET de nueve nodos, con matriz de tráfico uniforme $U(0,100)$

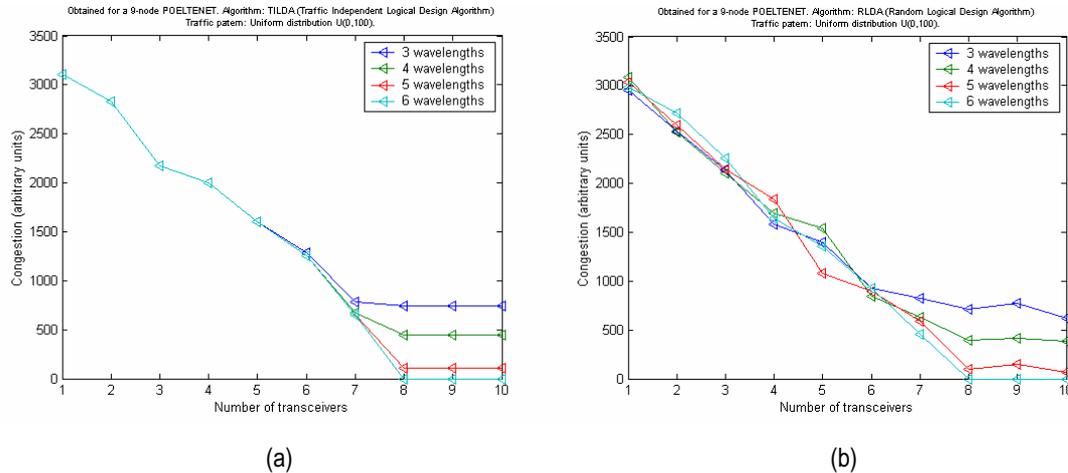


Figura 53. Curvas de congestión para la caracterización del TILDA (a) y RLDA (b)

Como se ve en el rango representado, el comportamiento global de ambos algoritmos es similar, también en niveles de congestión. Una primera etapa donde la diferencia de congestión viene dada por el número de transceptores y una segunda, donde las longitudes de onda determinan la congestión. Sin embargo, es evidente la inestabilidad de las curvas del RLDA, dado su comportamiento aleatorio; característica que lo hace un algoritmo imprevisible en una resolución individual, por lo que es dudosa su fiabilidad. Además, el tiempo de simulación de ambos algoritmos es idéntico.

5.3 COMPARATIVA DE HEURÍSTICOS

Después de la minuciosa caracterización del aparatado anterior, solamente queda corroborar las conclusiones expuestas. Para ello se comparan los cuatro algoritmos heurísticos en dos topologías físicas y con dos patrones de tráfico. En cada simulación se obtienen las curvas de congestión, ahora una por cada algoritmo. También se proporcionan los informes de resultados.

Con objeto de proporcionar una visión comparativa más ágil, se presentan juntos los informes de ambas simulaciones



```
ESCENARIO 1
#####
#####

SIMULATION RESULTS - 19/9/2006
-----

Algorithms compared: HLDA, MLDA, TILDA and RLDA

Obtained for a 9-node POELTENET
Traffic pattern: Uniform distribution U(0,100).
Interval of Tx and Rx: 6 and 12 (7 measurements)
Wavelengths per fiber : 5
Number of simulations: 7

-----

HLDA. Run in 1.594 seconds
MLDA. Run in 1.094 seconds
TILDA. Run in 0.219 seconds
RLDA. Run in 0.234 seconds

#####
St. Poelten University of Applied Sciences
Gonzalo J. Moreno Muñoz, 2006
#####

ESCENARIO 2
#####
#####

SIMULATION RESULTS - 19/9/2006
-----

Algorithms compared: HLDA, MLDA, TILDA and RLDA

Obtained for a 14-node NFSNET
Traffic pattern: Non uniform distribution. 25 % U(0,100) & 75 % U(0,10).
Interval of Tx and Rx: 5 and 12 (8 measurements)
Wavelengths per fiber : 6
Number of simulations: 8

-----

HLDA. Run in 6.813 seconds
MLDA. Run in 4.672 seconds
TILDA. Run in 0.578 seconds
RLDA. Run in 0.593 seconds

#####
St. Poelten University of Applied Sciences
Gonzalo J. Moreno Muñoz, 2006
#####
```

Lo más representativo, a la hora de evaluar los informes de resultados en el modo de comparación de heurísticos, son los tiempos de ejecución. Obviamente son proporcionales a la complejidad del algoritmo usado, por un lado, y al número de nodos de la topología por otro.

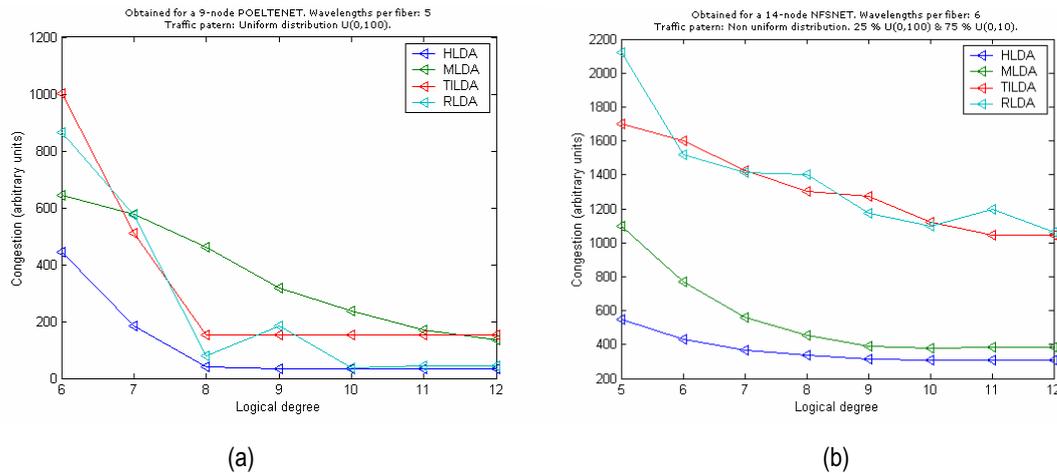


Figura 54. Curvas de congestión para algoritmos. (a) POELTENET 9 nodos. Patrón uniforme. (b) NSFNET 14 nodos. Patrón no uniforme.

En las gráficas anteriores se ponen de manifiesto las afirmaciones realizadas a lo largo de este capítulo. En un escenario con patrón uniforme las diferencias entre los cuatro algoritmos se reducen sustancialmente, hasta el punto de que el RLDA, que comprobamos muy pobre en sus resultados, puede llegar a alcanzar niveles de congestión similares a los del MLDA, pero insisto que se trata de un resultado de muy alta variabilidad. Otro algoritmo independiente de la matriz de tráfico, el TILDA, presenta un comportamiento muy lineal y abrupto en sus cambios. Así esa transición que hablamos entre la etapa lineal y la etapa estacionaria (respecto al número de transceptores) se percibe de forma brusca y radical. Por contra, la misma transición en MLDA se hace de forma amortiguada y suave, presentado más flexibilidad para elegir el número de transceptores por nodo, sin tener un cambio tan radical en los niveles de congestión.

En la Fig. 54 (b) se muestran las curvas de congestión de los algoritmos en un escenario con un patrón de tráfico no-uniforme. Es sensible la diferencia entre aquellos algoritmos que procesan en primer lugar los pares cargados y aquellos que diseñan la topología virtual con independencia de la matriz de tráfico. Como se puede observa el comportamiento del TILDA y RLDA vuelve a ser similar en términos globales pero con una alta variabilidad de este último en los casos individuales. También se muestra una pequeña diferencia entre los dos algoritmos con prima de pares de cargados, estando los niveles de congestión del MLDA por encima del HLDA; esta diferencia se acusa más en el transitorio para un número de transceptores más pequeño.

En definitiva y a modo de conclusión se puede establecer una clasificación general de los cuatro algoritmos en términos globales, no solamente observando los niveles de congestión sino



también los diversos indicadores y consideraciones expuestas en este capítulo. Esta sería y en orden decreciente: HLDA, MLDA, TILDA Y RLDA.

5.4 HEURÍSTICOS VS. PROGRAMACIÓN LINEAL. EJEMPLO

Hasta aquí se han presentado los resultados en diversos formatos que permiten caracterizar de forma relativa a los algoritmos heurísticos. Y es una caracterización relativa, porque todos los resultados, se tienen que poner en referencia a la solución óptima ofrecida por el problema de optimización en programación lineal. Ya se ha explicado a lo largo del Proyecto qué razones nos han llevado a desarrollar los heurísticos para evitar las limitaciones computacionales del problema del diseño de la topología virtual, pero la solución óptima tiene que ser una referencia necesaria a la hora de calificar el rendimiento de un algoritmo propuesto. En esta sección se van a presentar la solución óptima, y para el mismo problema, la alternativa heurística. Para ello se dispone de un conjunto de datos aportados por un el Proyecto Fin de Carrera de Ramón Aparicio Pardo [3] que se ha desarrollado de forma paralela a este.

La herramienta desarrollada en el citado Proyecto incluye la formulación en Matlab del problema de optimización, con sus respectivas restricciones, y la posterior solución mediante un *solver*. Las limitaciones en este caso están en el número de nodos de la topología física, y prácticamente son inviables las simulaciones para topologías con más de ocho nodos en ordenador personal. Para comprobar la correspondencia entre los resultados obtenidos por ambos métodos, nos centraremos en la solución de la topología virtual, pero también en indicadores tangenciales, pero sumamente importantes, desde el punto de vista práctico, como el tiempo de simulación.

5.4.1 Ejemplo AustriaNET de 7 nodos

La primera parte de este Proyecto y del que arriba se ha referencia se desarrolló en la Fachhochschule St. Pölten en Austria, por lo que la topología física que inspira este ejemplo, toma de referencia la distribución geográfica de siete capitales federales austriacas. Se trata de



una topología de 7 nodos a la que se aplica una matriz de tráfico basada en criterios de población y distancias

La implementación de la topología virtual mediante programación lineal se ha realizado partiendo de los datos del citado PFC para el mismo escenario. Por lo tanto el mismo problema se resuelve en primer término con el *solver*, obteniendo la solución óptima y posteriormente con los distintos heurísticos tratados: HLDA, TILDA, RLDA; en grado descendiente de complejidad. El heurístico MLDA no se puede simular ya que las condiciones de entrada de este problema no satisfacen sus requerimientos iniciales.

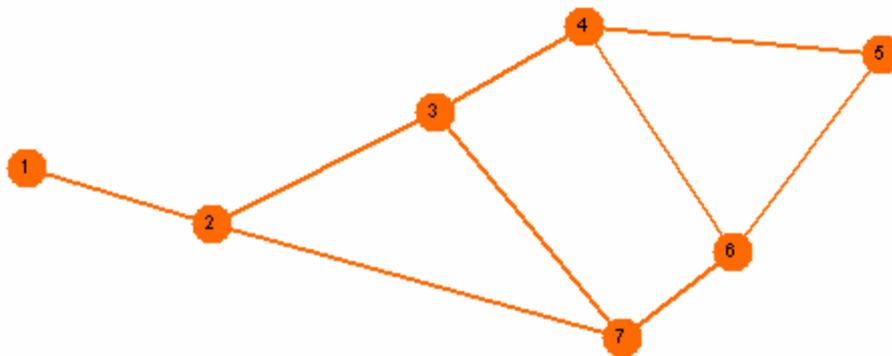


Figura 55. Topología física de la red AustriaNET de 7 nodos

Los datos de entrada al problema en formato Matlab son:

```

trafficMatrix =
    0    1.2135    0.8520    0.6598    0.4897    0.5487    0.6456
    1.2135    0    1.1951    0.8750    0.7124    0.7477    0.8918
    0.8520    1.1951    0    1.3197    1.0333    1.0048    1.0936
    0.6598    0.8750    1.3197    0    1.4519    1.1320    1.0213
    0.4897    0.7124    1.0333    1.4519    0    1.5400    0.9698
    0.5487    0.7477    1.0048    1.1320    1.5400    0    1.3385
    0.6456    0.8918    1.0936    1.0213    0.9698    1.3385    0

physicalTopology =
    0    1    0    0    0    0    0
    1    0    1    0    0    0    1
    0    1    0    1    0    0    1
    0    0    1    0    1    1    0
    0    0    0    1    0    1    0
    0    0    0    1    1    0    1
    0    1    1    0    0    1    0
    
```



5.4.1.1 SOLVER. Programación lineal

A continuación se muestra la topología virtual obtenida por el *solver* junto con la información de la simulación y algunos índices estadísticos. La función objetivo optimizada es la maximización del tráfico en un solo salto.

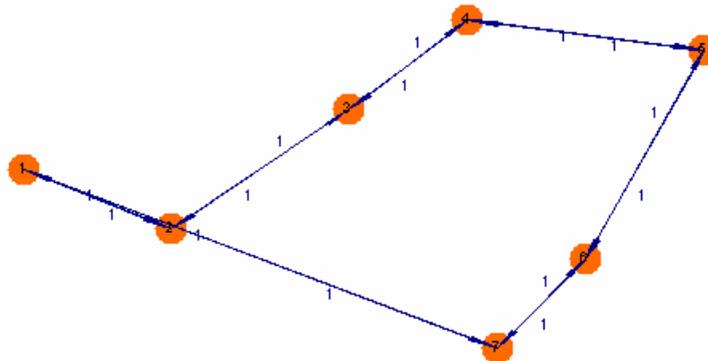


Figura 56. Topología virtual obtenida con el *solver* para AustriaNET

```
#####
#####
GENERAL INFORMATION
-----
Optimal integer solution found
VTD Problem solved in: 0 hours 12 minutes 2.89 seconds
Definition of the Problem solved in: 11 minutes 50.687 seconds
Optimization of the Problem solved in: 12.203 seconds
VTD Problem solved for a: 7-nodes AUSTRIANET
Traffic pattern: Realistic Distribution which depends on population and distance.
Maximum Number of Physical Hops a Lightpath: 4
Node degree: 2
Wavelengths per fiber: 3
Optimized Function: MaxSingleHopTraffic without Wavelength Conversion

STATISTICAL INFORMATION
-----
Percentage of Used Transmitters:100 %
Percentage of Used Receivers: 100 %
Percentage of Used Wavelength Channels: 29.6296 %
#####
#####
```

5.4.1.2 HLDA (HEURISTIC LOGICAL DESIGN ALGORITHM)

El diseño de la topología virtual con el algoritmo HLDA obtiene la solución óptima. Este algoritmo sigue el criterio de maximización el flujo de tráfico de salto único.

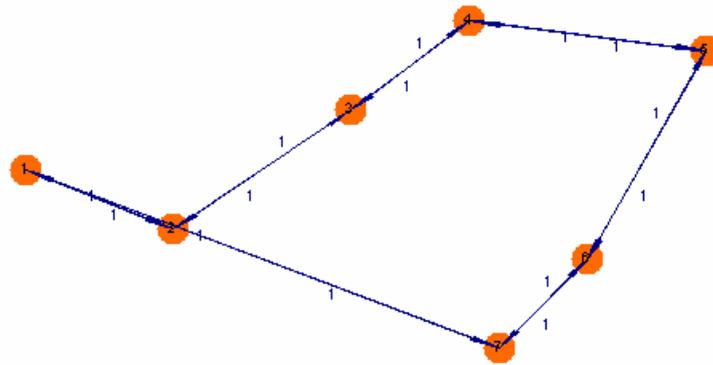


Figura 57. Topología virtual obtenida con el HLDA para AustriaNET

```
#####
#####
GENERAL INFORMATION
-----
VTD Problem solved in : 0.11 seconds
VTD Problem solved for a : 7-nodes AUSTRIANET
Traffic pattern: Realistic Distribution which depends on population and distance.
Node degree: 2
Wavelengths per fiber: 3

STATISTICAL INFORMATION
-----
Percentage of Used Transmitters: 100 %
Percentage of Used Receivers: 100 %
Percentage of Used Wavelength Channels: 29.6296 %
#####
#####
```

5.4.1.3 TILDA (TRAFFIC INDEPENDENT LOGICAL DESIGN ALGORITHM)

El algoritmo heurístico TILDA no tiene en cuenta la matriz de tráfico para el diseño de la topología virtual. Igual que en el caso anterior el TILDA obtiene la solución óptima.

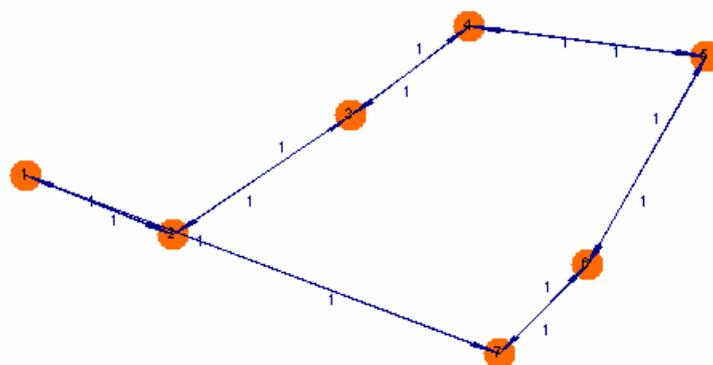


Figura 58. Topología virtual obtenida con el TILDA para AustriaNET



```
#####
#####
GENERAL INFORMATION
-----
VTD Problem solved in : 0.0620 seconds
VTD Problem solved for a : 7-nodes AUSTRIANET
Traffic pattern: Realistic Distribution which depends on population and distance.
Node degree: 2
Wavelengths per fiber: 3

STATISTICAL INFORMATION
-----
Percentage of Used Transmitters: 100 %
Percentage of Used Receivers: 100 %
Percentage of Used Wavelength Channels: 29.6296 %
#####
#####
```

5.4.1.4 RLDA (RANDOM LOGICAL DESIGN ALGORITHM)

El algoritmo RLDA propone una solución aleatoria que se aleja bastante de la solución óptima obtenida por el *solver*.

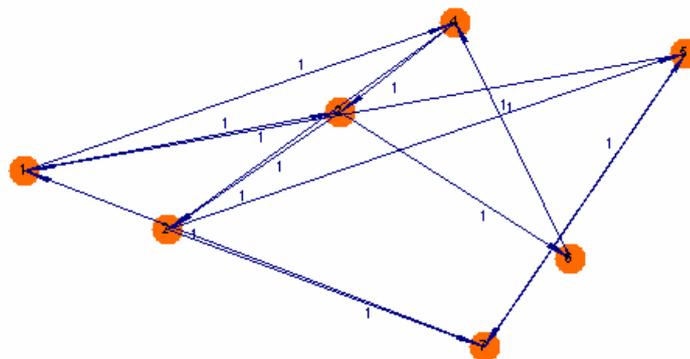


Figura 59. Topología virtual obtenida con el RLDA para AustriaNET

```
#####
#####
GENERAL INFORMATION
-----
VTD Problem solved in : 0.0620 seconds
VTD Problem solved for a : 7-nodes AUSTRIANET
Traffic pattern: Realistic Distribution which depends on population and distance.
Node degree: 2
Wavelengths per fiber: 3

STATISTICAL INFORMATION
-----
Percentage of Used Transmitters: 92.8571 %
Percentage of Used Receivers: 92.8571 %
Percentage of Used Wavelength Channels: 48.1481 %
#####
#####
```



5.4.1.5 Conclusiones

1. Tanto el HLDA como el TILDA obtienen la misma topología virtual que el *solver* (solución óptima). El TILDA la obtiene independientemente de la matriz de tráfico de forma más rápida, por lo que puede ser apropiado en requerimientos de diseño en tiempo real.
2. El salto cuantitativo se produce en el tiempo de diseño de la topología virtual. El *solver* supera los 12 minutos de tiempo mientras el algoritmo heurístico más lento obtiene la solución óptima en 0.11 segundos. Entre los algoritmos heurísticos, el TILDA y RLDA se revelan como los más rápidos con 0.0620 segundos, el primero de ellos obteniendo la solución óptima.
3. El algoritmo RLDA no obtiene la solución óptima en un tiempo idéntico al TILDA, por lo que se demuestra la eficiencia de este sobre aquél. Sin embargo la asignación aleatoria de *lightpaths* se ha revelado más eficiente que los otros dos heurísticos en la utilización de los canales lógicos de longitudes de onda.
4. La siguiente tabla resume los resultados de todas las simulaciones.

	Topología virtual	Tiempo	Tx/Rx usados	Canales usados
SOLVER	Óptima	>12 min	100 %	29.62 %
HLDA	Óptima	0.11 seg	100 %	29.62 %
TILDA	Óptima	0.0620 seg	100 %	29.62 %
RLDA	Subóptima	0.0620 seg	92.82 %	48.14 %



CAPÍTULO 6

Conclusiones

En la reciente evolución de las redes de comunicación, las redes ópticas se han ido abriendo paso de una forma progresiva. Estas se están consolidando como la solución óptima para las redes de alta capacidad por sus ventajas en distintos aspectos. Son conocidos desde hace años las características físicas de la fibra, sobre todo comparándolas con los medios eléctricos. A estas razones para el desarrollo actual de las redes ópticas se le añade la tecnología WDM. No es posible hablar de términos absolutos, sobre todo en el ámbito tecnológico, pero el salto cualitativo en términos de capacidad de transmisión, que introduce WDM ha originado un cambio radical de las expectativas de futuro de las redes ópticas. Las nuevas técnicas de multiplexación experimental llegan a introducir centenares de canales en una sola fibra, que se traduce en velocidades del orden de terabits (Tbps). Combinadas varias fibras en un mazo las velocidades en las redes de transporte son desconocidas hasta ahora. Parece evidente, por tanto, la importancia a corto y medio plazo de las redes de fibra óptica y concretamente de la tecnología WDM sobre la que se desarrolla el presente Proyecto.

Pero esta enorme capacidad de transmisión, ha creado un “gap” entre la velocidad a la que es posible transmitir por el medio físico (óptico), y la velocidad a la que los nodos de conmutación son capaces de procesar la información (electrónico). Como se ejemplificó en los primeros capítulos, la tecnología electrónica llevada al límite en SDH con una OC-768, nos ofrecía unas velocidades en torno a los 40 Gb/s. Una fibra óptica puede soportar más de 100 canales WDM, con 40 Gbps en cada canal. Realizar la conmutación electrónica de todo este tráfico, necesitaría de una interfaz SDH para cada uno de los canales, y un conmutador SDH (DCS, *Digital Crossconnect System*) para conmutar el tráfico desde los canales entrantes hacia los salientes. Los costes de conmutación se disparan hasta varios órdenes de magnitud del coste del medio de transmisión.



Con el objetivo de disminuir la cantidad de equipamiento electrónico necesario para hacer la función de conmutación en redes WDM, han aparecido una serie de alternativas llamadas O-O-O, ya que buscan que aquel tráfico que atraviesa un nodo (es decir, que no está originado ni destinado a un determinado nodo) pueda ser conmutado sin ser convertido a electrónico.

De entre las distintas alternativas existentes de conmutación O-O-O, este Proyecto Fin de Carrera se centra en la conmutación de *lightpaths*, consistente en la opción más avanzada hoy en día que se encuentra en estado comercial.

Los objetivos iniciales del Proyecto se centraron en la resolución de la instancia más importante, aunque no la única, asociada a las redes WDM de conmutación de *lightpaths*: la topología virtual, es decir, el conjunto de *lightpaths* de una red WDM.

Está demostrado que la obtención de la topología virtual óptima en una red WDM, es un problema NP-completo, y en términos de optimización se ha formalizado con el nombre RWA (*Routing and Wavelength Assignment*). Como todo problema de optimización, abordarlo mediante programación lineal, evidencia la intratabilidad a partir de un determinado tamaño de escenario. Es por esta razón que la aparición de algoritmos heurísticos se convierte, no sólo en alternativa, sino en la única posibilidad viable de resolución del problema, teniendo en cuenta los tamaños crecientes de la redes WAN

Cuando se abandona la solución óptima, de alguna forma, se relajan las exigencias de los resultados, y como consecuencia, todos los algoritmos heurísticos son subjetivos en sus planteamientos. Eso no está reñido con unos resultado subóptimos, que se puedan poner en comparación con los óptimos, si es computacionalmente posible, o con ciertos indicadores que nos hablen de la bondad del algoritmo diseñado. Son muchos los algoritmos propuestos en la literatura sobre este particular, pero algunos de ellos han confirmado su efectividad en diversos estudios. De estos cuatro algoritmos nos ocupamos, a un tiempo desarrollándolos de forma teórica y recogiendo sus resultados en la práctica.

Los algoritmos se han programado en Matlab siguiendo los desarrollos teóricos obtenidos de la literatura, aunque al ser no ser métodos formalizados, existe un cierto margen para la interpretación de programador. Quizá en términos de computación no sea el entorno adecuado,



pero teniendo en cuenta que es precisamente la complejidad computacional el inconveniente que ataja un algoritmo heurístico, este aspecto es poco relevante. Los tiempos de ejecución de los algoritmos son despreciables.

El trabajo de investigación del Proyecto ha terminado con la programación y evaluación de los algoritmos. Sus resultados se ajustan según lo observado en la literatura, a lo esperado y en mayor o menor medida a la solución óptima. Se ha puesto interés en que todos los códigos de los algoritmos estén debidamente comentados, así como en proporcionar ficheros de ayuda para facilitar su manejo y ejecución.

En una segunda etapa se desarrollan una serie de herramientas capaces de obtener la solución al problema de la topología virtual de forma rápida e intuitiva por un lado, y evaluar la eficiencia de los algoritmos por otro. No se trata sólo de dulcificar el manejo de los algoritmos y posibilitar la recogida e interpretación de los resultados, sino también de aportar información sobre el funcionamiento de los heurísticos, sobre todo en su comportamiento comparado. Esta perspectiva se relaciona con la parte docente del Proyecto. La herramienta que se presenta es una aplicación inédita y que puede de ser de gran utilidad en el estudio de los aspectos prácticos de las redes WDM.

La versatilidad de Matlab se pone de manifiesto también en su consola de resultados y el entorno gráfico GUI. En la primera se implementan los programas interactivos para la ejecución de los algoritmos y el segundo se recoge toda la funcionalidad de dichos programas para presentar de forma integrada la aplicación.

Como tareas tangenciales a la implementación de los algoritmos, cabe destacar el proceso de búsqueda y caracterización de los escenarios. Cuando se habla de escenario, nos referimos a una topología física (nodos y enlaces), la matriz de tráfico, el número de canales (longitudes de onda) en la fibra, y el número de transceptores (transmisores y receptos) por nodo. La combinación de estos elementos conforma el escenario, que determinará la topología virtual final. Por eso una cuidada selección de los escenarios, permite evaluar de forma fidedigna a los algoritmos. Se seleccionaron tres topologías físicas, en orden creciente de nodos (complejidad) y patrones de tráfico uniformes y no uniformes. Si bien la aplicación no es completamente personalizada, ya que al usuario se le restringen las topologías y los patrones de entrada, si es



criterio del planificador seleccionar las otras dos variables que determinan la simulación. No obstante, y respondiendo a criterios perfectivos de la aplicación, tanto en la estructura de las rutinas de ejecución como en código del GUI se permite la ampliación a otros supuesto, y la introducción de topologías y patrones adicionales es un paso inmediato. Otra de las funciones adicionales y que más ha contribuido a una presentación intuitiva y directa de los resultados, es plotTopology, que se desarrolló para dibujar por igual topologías físicas y virtuales con sólo cambio de un parámetro.

La aplicación GUI resultante proporciona una herramienta atractiva y accesible para poder caracterizar en diversos modos, y con varias opciones de simulación un algoritmo. Por lo tanto, y siguiendo la línea cronológica del Proyecto, con la finalización del MatPlanWDM se llega a la obtención de resultados que contrastan los planteamientos iniciales de los algoritmos HLDA, MLDA, TILDA y RLDA. De ellos se deriva una serie de conclusiones que confirman los planteamientos *a priori* y otras nuevas que han sido posibles por la caracterización que con nuestra herramienta han sido sometidos. Entre las primeras cabe destacar a modo de ejemplo, los resultados del HLDA como mejor algoritmo heurístico, más estable y eficiente en todos los escenarios. Entre las segundas se puede apuntar la idoneidad de algunos algoritmos, calificados de segunda categoría (TILDA), para ciertos escenarios; presentado unos resultados similares a las mejores soluciones heurísticas pero mejorando algunas variables tangenciales, por ejemplo el tiempo de ejecución.

Quizá uno de los aspectos más importantes, por la globalidad que aporta, es la comparación entre los resultados heurísticos y los obtenidos por programación lineal. La calidad de resultados obtenidos, su comparación, etc. no puede desconocer la relatividad de todos ellos. Y es la relatividad propia de la solución subóptima respecto al resultado óptimo. Es por ello que la coincidencia en tiempo y desarrollo de dos Proyectos Fin de Carrera con el mismo substrato teórico y distintas alternativas prácticas, ha contribuido sólidamente a mantener una visión de conjunto del problema general.

Para terminar y volviendo otra vez al principio, el problema del diseño de la topología virtual se incardina de forma muy íntima con la planificación de redes WDM y si es verdad que existen una cantidad de instancias y líneas de investigación que pueden contribuir a ese desarrollo



espectacular que al principio se ha hecho referencia, nuestro problema y los heurísticos que lo abordan forman una parte de gran interés didáctico en este conjunto de tecnologías.



Bibliografía

- [1] Optical WDM Networks: principles and practice / Edited by Krishna M. Sivalingam, Sresh Subramaniam.
- [2] WDM optical networks: concepts, design, and algorithms / C. Siva Ram Murthy, Mohan Gurusamy.
- [3] Aparicio Pardo R. (2006). PFC “Optimización en redes WDM de conmutación de *lightpaths* mediante modelos de programación lineal entera-mixta”. Universidad Politécnica de Cartagena.
- [4] Ramaswami, R. and Sivarajan, K. (1996). Design of logical topologies for wavelength-routed optical networks. IEEE JSAC, 14(5):840–851.
- [5] J. Katou, S. Arakawa and M. Murata. “A Design Method of Logical Topology with Stable Packet Routing in IP over WDM Network”.
- [6] Banerjee, D. and Mukherjee, B. (1996). A practical approach for routing and wavelength assignment in large wavelength-routed optical networks. IEEE JSAC, 14(5):903–908.
- [7] Banerjee, D. and Mukherjee, B. (1997). Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study. Proc. IEEE INFOCOM, pages 269–276.
- [8] Banerjee, S., Yoo, J., and Chen, C. (1997). Design of wavelength-routed optical networks for packet switched traffic. Journal of Lightwave Technology, 15(9): 1636–1646.
- [9] Chen, C. and Banerjee, S. (1995). Optical switch configuration and *lightpath* assignment in wavelength routing multihop lightwave networks. Proc. IEEE INFOCOM, pages 1300–1307.
- [10] Chlamtac, I., Ganz, A., and Karmi, G. (1992). *Lightpath* communications: An approach to high bandwidth optical wans. IEEE Trans. Communications, 40(7):1171–1182.



- [11] Chlamtac, I., Ganz, A., and Karmi, G. (1993). Lightnets: Topologies for high-speed optical networks. *Journal of Lightwave Technology*, 11(5/6):951–961.
- [12] Gerstel, O., Ramaswami, R., and Sasaki, G. (1998). Cost effective traffic grooming in WDM rings. *Proc. IEEE INFOCOM*, pages 69-77.
- [13] Green, P. (1992). *Fiber optic network*. Englewood Cliffs, NJ Prentice Hall.
- [14] Green, P. (1996). Optical networking update. *IEEE JSAC*, 14(5):764–779.
- [15] Krishnaswamy, R. and Sivarajan, K. (1998). Design of logical topologies: a linear formulation for wavelength routed optical networks with no wavelength changers. *Proc. IEEE INFOCOM*, pages 919–927.
- [16] Labourdette, J.-F. P. (1998). Traffic optimization and reconfiguration management of multiwavelength multihop broadcast lightwave networks. *Computer Networks and ISDN Systems*, 30(9-10):981–998.
- [17] Marsan, M, Bianco, A., Leonardi, E., and Neri, F. (1993). Topologies for wavelength-routing all-optical networks. *IEEE/ACM Trans. Networking*, 1(5):534–546.
- [18] Mukherjee, B. (1997). *Optical communication networks*. McGraw-Hill.
- [19] Mukherjee, B., Ramamurthy, S., Banerjee, D., and Mukherjee, A. (1994). Some principles for designing a wide-area optical network. *Proc. IEEE INFOCOM*, pages 110–119.
- [20] Mukherjee, B., Ramamurthy, S., Banerjee, D., and Mukherjee, A. (1996). Some principles for designing a wide-area optical network. *IEEE/ACM Trans. Networking*, 4(5):684–696.
- [21] Ramaswami, R. and Sasaki, G. (1997). Multiwavelength optical networks with limited wavelength conversion. *Proc. IEEE INFOCOM*, pages 489–498.
- [22] Ramaswami, R. and Sivarajan, K. (1998). *Optical networks: a practical perspective*. Morgan Kaufmann Publishers.
- [23] Sahasrabudde, L. and Mukherjee, B. (1999). Light-trees: Optical multicasting for improved performance in wavelength-routed networks. *IEEE Communications Magazine*, pages 67–73.
- [24] Zhang, Z. and Acampora, A. (1995). A heuristic wavelength assignment algorithm for multihop WDM networks with wavelength routing and wavelength reuse. *IEEE/ACM Trans. Networking*, 3(3):281–288.
- [25] Murthy, C. and Gurusamy, M. “WDM Optical Networks: concepts, design, and algorithms”. Chapter 5, pp. 209-210.



- [26] Banerjee, S. Yoo, J. and Chen, C. "Design of Wavelength Routed Optical Network for Packet Switched Traffic," IEEE/OSA Journal of Lightwave Technology, vol. 15, no. 9, pp 1636-1646, September, 1997.
- [27] J. Katou, S. Arakawa and M. Murata. "A Design Method of Logical Topology with Stable Packet Routing in IP over WDM Network".
- [28] Chlamtac I., Farango A., Zhang T. "*Lightpath* Communications: An Approach to High Bandwidth Optical WANs", IEEE Transactions on Communications, vol. 40, no. 7, pp. 1171-1182, July 1992.