

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN  
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



PROYECTO FIN DE CARRERA

## Gestión de Fallos de Redes Ópticas Wavelength-routing con Demanda Estática de Tráfico



Autor: Pedro Francisco Páez Martínez  
Director: Pablo Pavón Mariño  
Codirectora: María Belén García Manrubia

Cartagena, Enero 2013





<b>Autor</b>	Pedro Francisco Páez Martínez
<b>E-mail del Autor</b>	pedro.paez@si.upct.es
<b>Director</b>	Pablo Pavón Mariño
<b>E-mail del Director</b>	pablo.pavon@upct.es
<b>Codirector</b>	María Belén García Manrubia
<b>Título del PFC</b>	Gestión De Fallos De Redes Ópticas Wavelength-routing Con Demanda Estática De Tráfico
<b>Descriptores</b>	Planificación de redes WDM
<b>Resumen</b>	<p>La tecnología óptica WDM articula 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. El presente proyecto aborda el desarrollo de una extensión de la herramienta MatPlanWDM orientada a la planificación de redes WDM wavelength-routing considerando demanda estática de tráfico y proveyendo un algoritmo de protección que dinámicamente decide como encaminar el flujo teniendo en cuenta el estado y la topología física de la red.</p>
<b>Titulación</b>	Ingeniero de Telecomunicación
<b>Intensificación</b>	Planificación y Gestión de Telecomunicaciones
<b>Departamento</b>	Tecnologías de la Información y las Comunicaciones
<b>Fecha de Presentación</b>	Enero de 2013



# Índice General

---

<b>1. Introducción</b>	<b>1</b>
1.1. Redes troncales WDM	1
1.1.1. Introducción	1
1.1.2. Wavelength Division Multiplexing	2
1.2. Redes OCS	4
1.2.1. Descripción de las redes OCS	4
1.2.2. Paradigma de las redes OCS	6
1.2.3. Estructura de un nodo	7
1.3. Survivable WDM networks	9
1.3.1. Introducción	9
1.3.2. Terminología	10
1.3.3. Protección y restauración	10
1.4. Herramienta MatPlanWDM	12
1.5. Objetivos del proyecto	14
<b>2. Descripción de la extensión desarrollada</b>	<b>17</b>
2.1. Introducción	17
2.2. Herramienta	17
2.2.1. Estructura del toolbox	17
2.2.2. Parámetros de entrada	17
2.2.3. Parámetros de salida	18
2.2.4. Generador de fallos	18
2.2.5. Algoritmos de protección	19
2.2.6. eventLoop	20
2.2.7. Estadísticos calculados	20
2.2.8. Extensión de la herramienta	22
2.3. Manual de usuario	22
2.3.1. Instalación y arranque de la aplicación	22

## ÍNDICE GENERAL

---

2.3.2.	Ventana principal . . . . .	23
2.3.3.	Estructura del fichero PR1-netState . . . . .	27
2.3.4.	Estructura del generador de fallos . . . . .	28
2.3.5.	Estructura del módulo de protección (PR2) . . . . .	30
2.3.6.	Estructura de la variable netState . . . . .	30
2.4.	Algoritmos implementados . . . . .	32
2.4.1.	Generador de fallos . . . . .	32
2.4.2.	Algoritmo PR1 . . . . .	32
2.4.3.	Algoritmo PR2 . . . . .	33
<b>3.</b>	<b>Resultados obtenidos</b>	<b>35</b>
3.1.	Introducción . . . . .	35
3.2.	Escenarios . . . . .	35
3.3.	Resultados obtenidos . . . . .	37
<b>4.</b>	<b>Conclusiones y líneas futuras</b>	<b>39</b>
4.1.	Conclusiones y líneas futuras . . . . .	39
	<b>Bibliografía</b>	<b>41</b>

# Índice de Figuras

---

1.1. Implementación TDM . . . . .	1
1.2. Implementación CDM . . . . .	2
1.3. Ventana óptica de baja atenuación . . . . .	2
1.4. Propagación de canales WDM por una única fibra . . . . .	3
1.5. Sistema WDM simple . . . . .	3
1.6. Red OCS . . . . .	5
1.7. Red de lightpaths sobre una red OCS . . . . .	6
1.8. Estructura de un nodo de conmutación . . . . .	8
1.9. Diferentes tipos de fallos en una red óptica WDM . . . . .	9
1.10. Link protection vs Path protection . . . . .	11
1.11. Modelos de protección . . . . .	12
1.12. Diseño de topología virtual y encaminamiento de flujos . . . . .	13
1.13. What-If Analysis . . . . .	14
1.14. Análisis Multihora . . . . .	15
1.15. Análisis dinámico . . . . .	16
2.1. Estructura del toolbox . . . . .	18
2.2. Ventana principal . . . . .	22
2.3. Detalle Input Data . . . . .	23
2.4. Results Information . . . . .	24
2.5. Physical Topology . . . . .	25
2.6. Virtual Topology . . . . .	25
2.7. Generador de fallos . . . . .	26
2.8. Ejemplo de fichero pr1-netState (trafficDemand) . . . . .	27
2.9. Ejemplo de fichero pr1-netState (virtualTopology) . . . . .	28
2.10. Ejemplo de fichero pr1-netState (physicalTopology) . . . . .	29
3.1. Topología de Internet2 . . . . .	35
3.2. Topología de NSFNET . . . . .	36

## ÍNDICE DE FIGURAS

---

3.3. Bloqueo en Internet2 . . . . .	37
3.4. Bloqueo en NSFNET . . . . .	38



# Índice de Tablas

---

2.1. Propiedades y tipos de fallos . . . . .	30
--	----



# Capítulo 1

## Introducción

---

### 1.1. Redes troncales WDM

#### 1.1.1. Introducción

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, mientras que las actuales tasas de tecnología electrónica se sitúan entorno a decenas de gigabits por segundo (Gbps). Por esta razón, es necesario invertir esfuerzo en explotar esta diferencia abismal entre el ancho de banda ofrecido por el dominio eléctrico y el óptico. En una red de comunicaciones ópticas la concurrencia puede implementarse mediante distintas técnicas como multiplexación de longitud de onda-frecuencia (WDM), ranuras temporales (TDM), forma de onda (espectro ensanchado, multiplexación por división de código (CDM)) o una combinación de las mismas.

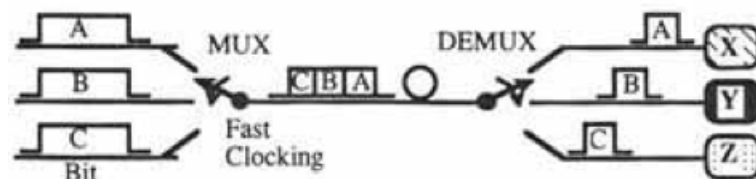


Figura 1.1: Implementación TDM

Si se opta por la tecnología TDM, tendremos que 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

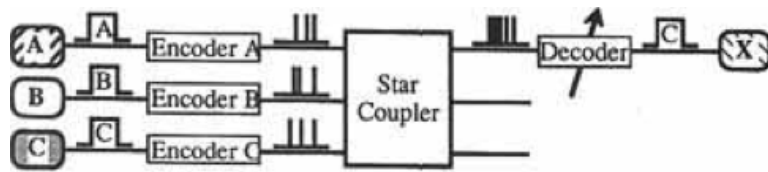


Figura 1.2: Implementación CDM

operar a una velocidad más alta que la velocidad eléctrica. Esta restricción hace que las tecnologías TDM y CDM sean relativamente menos atractivas que WDM.

Actualmente, la tecnología óptica de multiplexación mejor posicionada en las redes de comunicaciones ópticas es WDM 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 este hecho, los esfuerzos más importantes se están haciendo en el desarrollo y aplicación de WDM y en su comercialización.

### 1.1.2. 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 tal y como vemos en la figura 1.3. Sin embargo, no hay que olvidar la limitación impuesta en la tecnología electrónica de los usuarios finales que sólo permite que las velocidades alcanzadas (tasas de datos) del orden de gigabits por segundo (Gbps). Este  *cuello de botella*  electrónico dificulta el aprovechamiento del ancho de banda potencial que ofrece una fibra mediante un solo canal (longitud de onda).

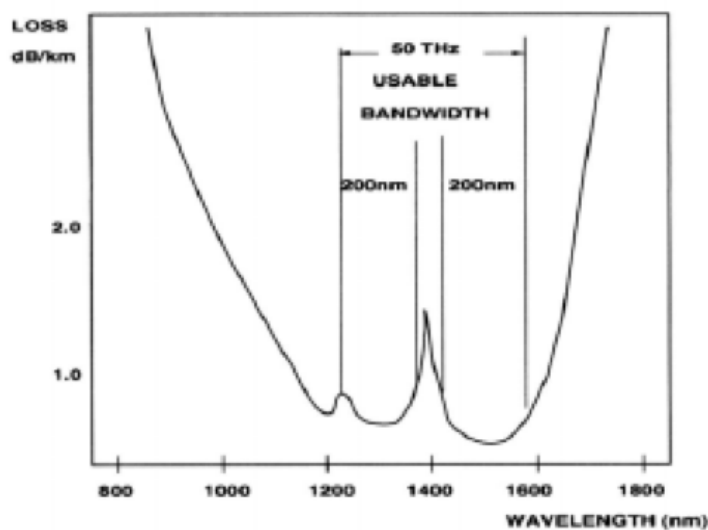


Figura 1.3: Ventana óptica de baja atenuación

Los avances recientes que han permitido alcanzar velocidades de terabits por segundo (Tbps) son el resultado de dos desarrollos diferentes. Por un lado, la tecnología wavelength division multiplexing (WDM) 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) amplifica simultáneamente un conjunto de señales a diferentes longitudes de onda, sin tener en cuenta su velocidad o tipo de modulación.

Conceptualmente se puede trazar un paralelismo entre WDM y FDM (frequency division multiplexing), pues 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 1.4).

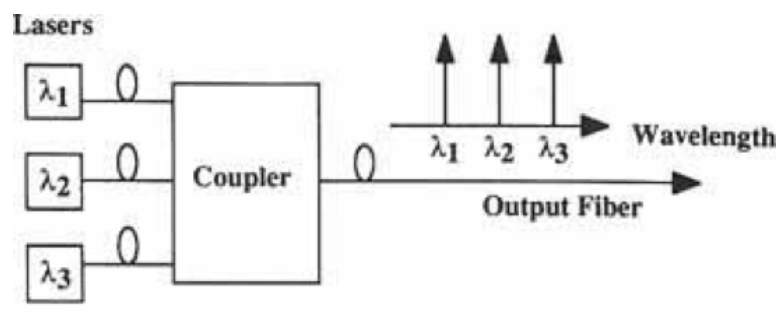


Figura 1.4: Propagación de canales WDM por una única fibra

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.

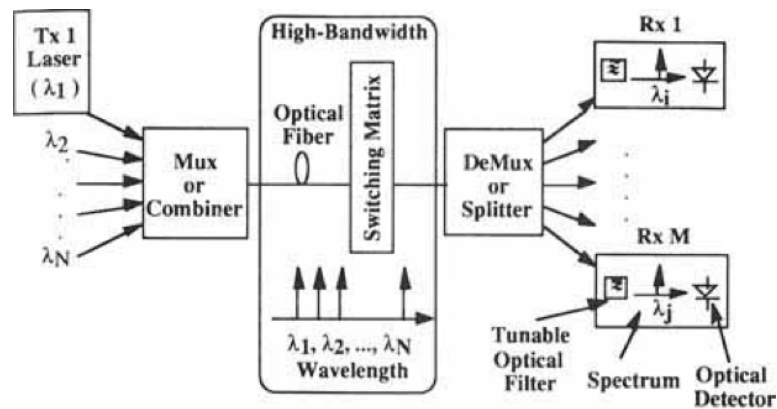


Figura 1.5: Sistema WDM simple

Una única longitud de onda también puede ser dividida entre varios nodos de la red mediante TDM (time division multiplexing). 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.

## 1.2. Redes OCS

### 1.2.1. Descripción de las redes OCS

Las redes OCS (Optical Circuit Switching) presentan la peculiaridad de evitar los tres problemas principales de las redes broadcast: reutilización de longitudes de onda, pérdidas de potencia por división de señal y escalabilidad a redes de área extensas (WANs). En esencia una red OCS 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.

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 de onda 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 OCS presentan ventajas considerables. La propiedad de reutilizar las longitudes de onda en conjuntos disjuntos de enlaces asegura una mayor escalabilidad de la red. Otra característica importante, que permite a las redes wavelength-routed extenderse hasta grandes dis-

tancias, es que la potencia de los lightpaths no se divide en destinos irrelevantes.

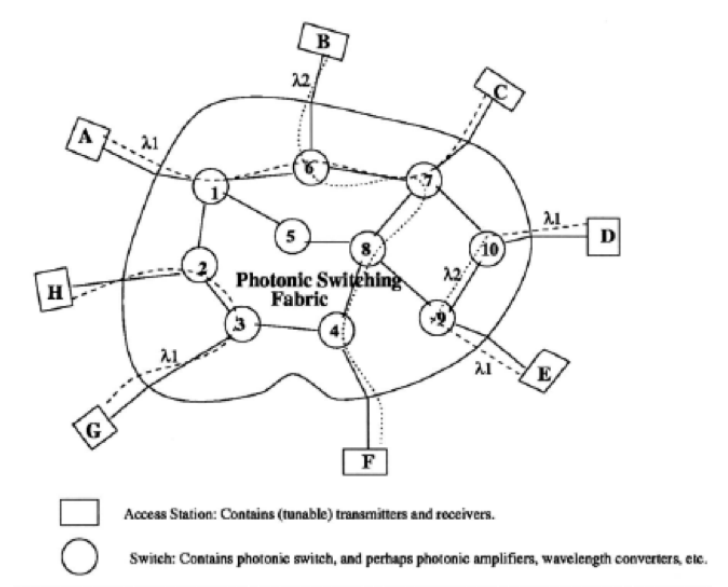


Figura 1.6: Red OCS

El problema fundamental en este tipo de redes, 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 OCS se puede modelar mediante sistemas mono-salto o multi-salto. En la aproximación multi-salto, una topología virtual<sup>1</sup> se superpone sobre la topología física mediante los WXC en los nodos proveyendo a los clientes de la capa superior de transparencia de protocolos. Sobre esta topología virtual los paquetes se encaminan 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 procesamiento, siguiendo la operación inversa y su retransmisión en otra longitud de onda.

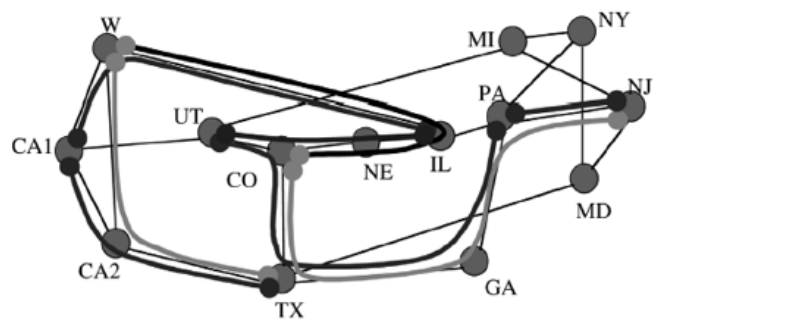
Las características mencionadas -reutilización de longitud de onda, transparencia de protocolos y alta fiabilidad- hacen de las redes OCS idóneas para las redes de área extensa. Estas redes exigen el tratamiento de tres cuestiones fundamentales: diseño, reconfiguración, corrección de fallos. El diseño 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

<sup>1</sup>Conjuntos de lightpaths establecidos de forma lógica. También se denomina red de lightpaths o topología lógica

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.

### 1.2.2. Paradigma de las redes OCS

Los usuarios finales en una red troncal basada en fibra se comunican entre ellos por medio de canales en el dominio óptico que hemos denominado 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 Figura 1.7 se muestra un ejemplo de red de lightpaths.



**Figura 1.7:** Red de lightpaths sobre una red OCS

En una red de  $N$  nodos, si cada nodo está equipado con  $N-1$  transceptores<sup>2</sup> 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 físicas que atravesarán los lightpaths y qué longitud de onda ocupará en cada una; todo con el objetivo de que se establezcan el máximo número de 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 variar para posibilitar que se establezcan un número mayor de lightpaths. Los lightpaths que no se puedan establecer debido a

---

<sup>2</sup>El término transceptor se refiere a los transmisores (láseres) y los receptores (filtros)



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, 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 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. Estos nodos deben contener dispositivos llamados conversores de longitud de onda sintonizable (*Tunable Wavelength Converter*, TWC), que son capaces de modificar la longitud de onda de un canal WDM, sin conversión a electrónico.

Este problema particular se denomina RWA por sus siglas en inglés: *Routing and Wavelength Assignment*. Hemos visto que este planteamiento supone partir de la topología física de la red y su demanda de tráfico para obtener la topología virtual que constituida por un conjunto de lightpaths, los cuales pueden permitir o no conversión, y el encaminamiento del tráfico sobre dicha topología virtual.

### 1.2.3. Estructura de un nodo

Como hemos visto en apartados anteriores las redes OCS están diseñadas para aliviar la diferencia existente entre las capacidades de transmisión disponibles en los enlaces WDM y la capacidad de conmutación electrónica de los nodos que es muy inferior. También sabemos que en este tipo de redes el tráfico es transportado en lightpaths transparentes que pueden atravesar un número variable de nodos. Esta idea se caracteriza por multiplexores add/drop ópticos reconfigurables (R-OADM) o nodos de enrutamiento reconfigurables (RWXC), que permiten que unas longitudes de onda sean procesadas electrónicamente ya que el tráfico o se queda en el nodo, o se añade al enlace, mientras otras son conmutadas de forma óptica sin necesidad de conversión electrónica[1]. De esta forma se consigue reducir el efecto del cuello de botella electrónico.

En la figura 1.8, podemos identificar los principales componentes de un nodo de conmutación de forma visual. Estos elementos determinarán el coste global de la red y son los que se enumeran seguidamente: a) el número de transmisores (T) electro-ópticos (E/O) y el número de receptores (R) optoelectrónicos (O/E) en el nodo, b) el número de conversores de longitud de onda sintonizables (TWC) que necesita el nodo, c) el tamaño del componente WSF, y d) la capacidad de conmutación electrónica requerida, dada por la suma de los tráficos ingress, egress y grooming. La parte óptica esencial del nodo es el componente Wavelength Switch Fabric (WSF) que permite realizar una conmutación transparente de los canales WDM desde los puertos de entrada a los puertos de salida. Esta parte del nodo, se puede construir empleando una configuración fija o reconfigurable. En el primer caso, las conexiones entre los puertos de entrada y salida del WSF quedan establecidas previamente con lo que no pueden ser cambiadas con el fin de adaptarse a la demanda. En el caso reconfigurable, el WSF se implementa empleando multiplexores ópticos add/drop reconfigurables (R-OADM) o bien nodos

de enrutamiento reconfigurables (RWXC), con lo que las conexiones entre los puertos de entrada y salida pueden ser reconfiguradas de forma dinámica en decenas de milisegundos permitiendo así que el nodo se adapte a la demanda.

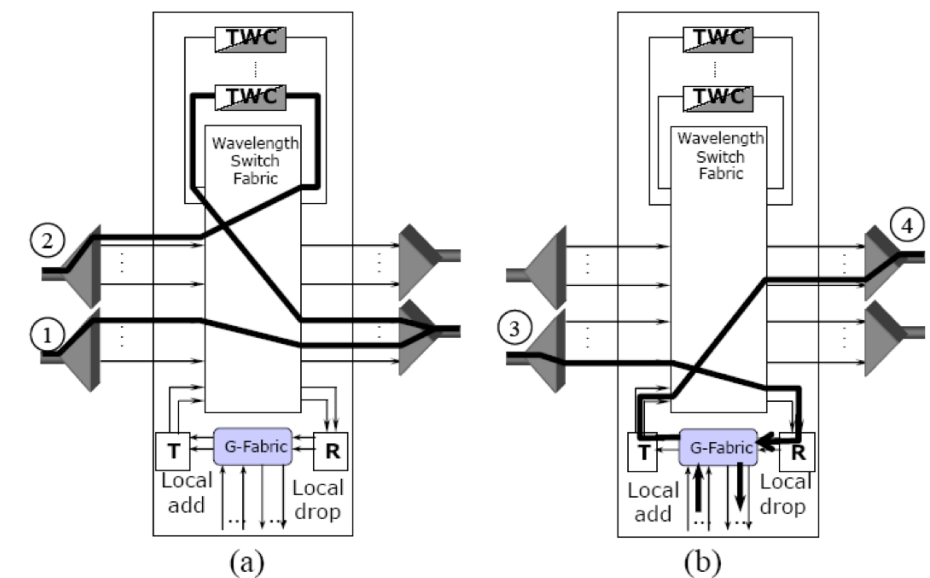


Figura 1.8: Estructura de un nodo de conmutación

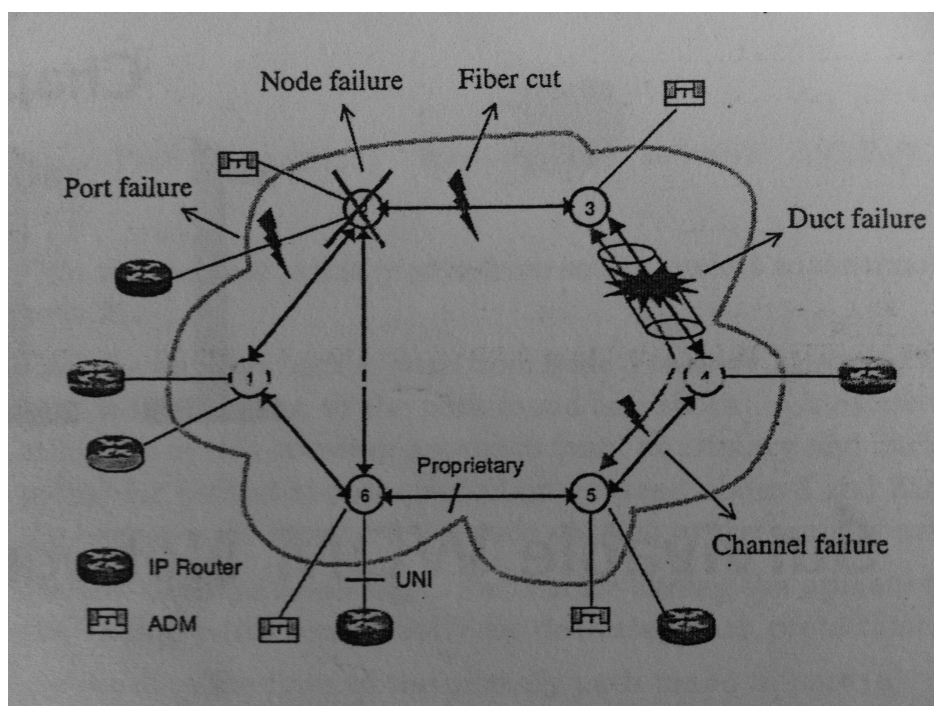
En la parte (a) de la figura 1.8 podemos identificar con claridad los dos tipos de lightpaths: con conversión y sin conversión. El lightpath(1) entra al nodo con una longitud de onda y continúa hacia el siguiente nodo manteniendo dicha longitud de onda sin necesidad de atravesar ningún conversor. En el otro caso, el lightpath(2) atraviesa un conversor de longitud de onda sintonizable (TWC), que se encarga de realizar una conversión de longitud de onda del canal WDM en el dominio óptico. Por tanto, el lightpath llega al nodo con una longitud de onda y continúa hacia el siguiente con una longitud de onda diferente pero sin necesidad de procesamiento electrónico, consiguiendo así aliviar el cuello de botella electrónico.

Los lightpaths (3) y (4) que vemos en la parte (b) de la figura muestran el concepto de *Traffic grooming* que se detalla a continuación. El lightpath(3) llega al nodo y atraviesa la G-Fabric, que se encarga del procesamiento electrónico, ya que parte del tráfico tiene como destino este nodo con lo cual se queda en él y parte del tráfico será acomodado en otros lightpaths como el (4) que tienen como nodo origen este nodo, compartiendo el lightpath con tráfico que tiene como origen este nodo y/o con otro tráfico grooming. El equipamiento que requiere cada nodo en una red depende de los lightpaths establecidos y sus rutas establecidas sobre la topología física que forman la topología virtual.

## 1.3. Survivable WDM networks

### 1.3.1. Introducción

En la figura 1.9 podemos ver los diferentes tipos de fallos que pueden ocurrir en una red óptica WDM. Un “ducto” es una tubería bidireccional que une dos nodos y que contiene un conjunto de fibras en su interior. Un corte de fibra sucede normalmente cuando se producen cortes en el ducto debidos a obras, desastres naturales, etc. Todos los lightpaths que atravesaban la fibra dañada se ven afectados produciéndose una enorme pérdida de tráfico.



**Figura 1.9:** Diferentes tipos de fallos en una red óptica WDM

Los fallos en el equipamiento óptico de los nodos (OXC) normalmente son debidos a catástrofes naturales como incendios o inundaciones. Estos fallos se recogen como fallos de nodo. Los fallos de nodo ocurren ocasionalmente pero el daño es considerable. Además de los fallos de nodos y de enlace, existen otro tipo de fallos llamados “fallos de canal” que también pueden aparecer en las redes ópticas WDM. Un fallo de canal ocurre normalmente debido a un error en el equipamiento de transmisión/recepción que opera en ese canal.

Debida a la alta frecuencia con la que se producen cortes de fibra y la gran pérdida de tráfico que producen, se hace necesario diseñar métodos efectivos para recuperar los enlaces y los nodos afectados. La mayoría de las publicaciones de este campo se centran en la recuperación de un enlace simple o un único nodo, donde un fallo se repara antes de que ocurra el siguiente. Esto es debido a que la mayor parte del tiempo se cumple

este escenario. No obstante, los fallos múltiples (o muy cercanos en el tiempo) ocurren en las redes reales y se deben diseñar métodos capaces de manejarlos.

Los grupos de riesgo compartidos (Shared Risk Groups - SRGs) [10] expresan las relaciones de riesgo que asocian todos los canales ópticos con un único fallo. Por ejemplo, un SRG puede consistir en todos los canales ópticos de una fibra, todos los canales de todas las fibras de un ducto, etc. Como una fibra puede atravesar varios conductos, un canal puede pertenecer a varios grupos de riesgo. Los algoritmos de protección deben explotar los mapas de riesgo para descubrir rutas alternativas para los SRGs para que, en caso de corte, haya siempre al menos una ruta disponible. Si ampliamos el concepto SRG para incluir grupos de nodos y enlaces próximos entre sí, dispondremos de una cobertura mayor.

### 1.3.2. Terminología

En esta sección, vamos a definir algunos términos de interés que se usarán en el documento.

- *Tiempo de restauración (Restoration time (RT))* mide el momento desde que una conexión se “cae” hasta el instante en el que se restaura el tráfico a través de un camino predefinido (protección) o uno obtenido dinámicamente (restauración).
- *Ratio de éxito de restauración (Restoration success rate (RSR))* denota el ratio entre el número de flujos de tráfico recuperados frente al número de flujos de tráfico afectados después de un fallo en la red.
- *Capacidad de restauración (Service restorability)* es un parámetro que representa la capacidad de una red de sobrevivir frente a un fallo concreto.
- *Disponibilidad (Availability)* se define como la probabilidad de que una conexión esté activa en un momento aleatorio.
- *Fiabilidad (Reliability)* es la probabilidad de que un sistema funcione sin interrupciones en un determinado período de tiempo.

### 1.3.3. Protección y restauración

Existen dos tipos de mecanismos de recuperación ante fallos: *protección* y *restauración*. Si disponemos de recursos de respaldo (rutas alternativas y longitudes de onda) previamente reservados, estamos ante un modelo de *protección*. En otro caso, si ante cada fallo en la red, es necesario descubrir dinámicamente una ruta alternativa y una longitud de onda libre, estamos ante un modelo de *restauración*.

Generalmente, los modelos de restauración dinámica son más eficientes utilizando la capacidad de la red ya que no asignan recursos por adelantado y proveen de capacidad de recuperación frente a diferentes tipos de fallos (incluidos fallos múltiples). Por otra parte, los modelos de protección tienen un tiempo de recuperación menor y garantizan

la recuperación de los servicios protegidos (garantía que no podemos asegurar con modelos de restauración).

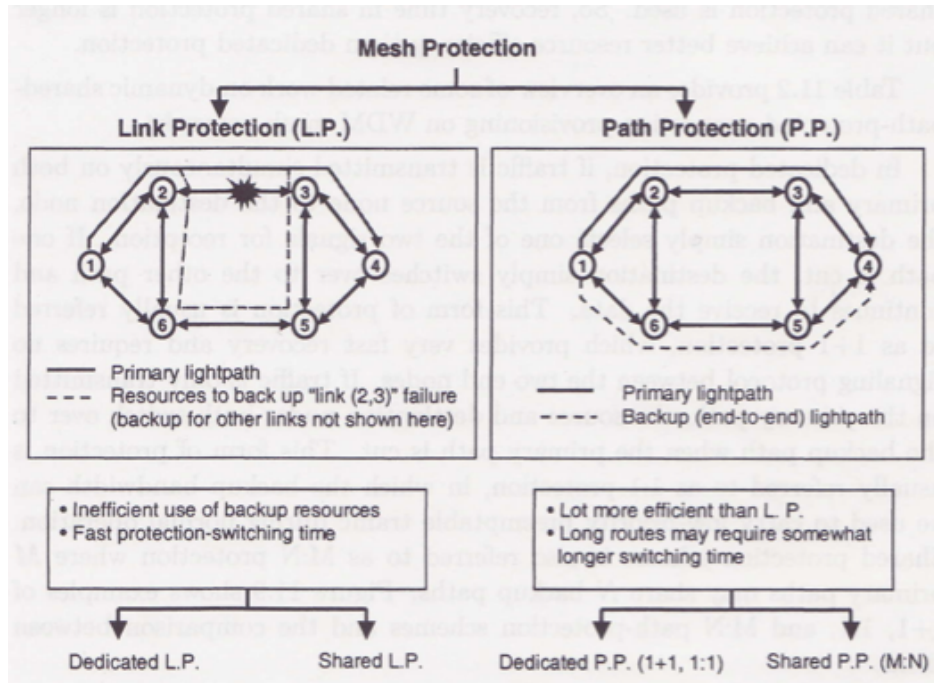


Figura 1.10: Link protection vs Path protection

Los modelos de protección se pueden dividir en dos grupos: *path protection* y *link protection*. En *path protection*, el tráfico se reenruta a través de un camino de respaldo (*backup path*) cuando se produce un fallo en un enlace primario (*primary path*). Ambos caminos deben ser disjuntos de manera que un único fallo en un enlace no pueda afectar a ambos simultáneamente. En *link protection* el tráfico se reenruta sólo a través del enlace caído. En la figura 1.10 se recoge un esquema resumen con las diferencias.

También podemos agrupar los modelos de protección en compartidos o dedicados, en función de si ocupan o no el camino de respaldo con tráfico. En protección dedicada, si el tráfico se transmite simultáneamente a través del enlace primario y el de respaldo, el destino simplemente selecciona una de las dos señales en recepción. Si uno de los enlaces cae, el destino selecciona el otro enlace y continúa recibiendo la información. Esta forma de protección, que proporciona una recuperación muy rápida, se conoce como protección 1+1. Si el tráfico sólo se transmite por el enlace primario y se utiliza el de backup con tráfico no-prioritario mientras que el primario está activo, hablamos de un esquema de protección 1:1. Los modelos de protección compartida se conocen como M:N donde  $M$  enlaces primarios pueden compartir  $N$  caminos de backup.

Los modelos de restauración dinámica también se pueden clasificar en *link restoration* y *path restoration* dependiendo del tipo de reentramiento. En *link restoration*, los nodos finales del enlace caído descubren dinámicamente una ruta a través del enlace, para cada conexión (longitud de onda) que atravesaba dicho enlace. En *path restaura-*

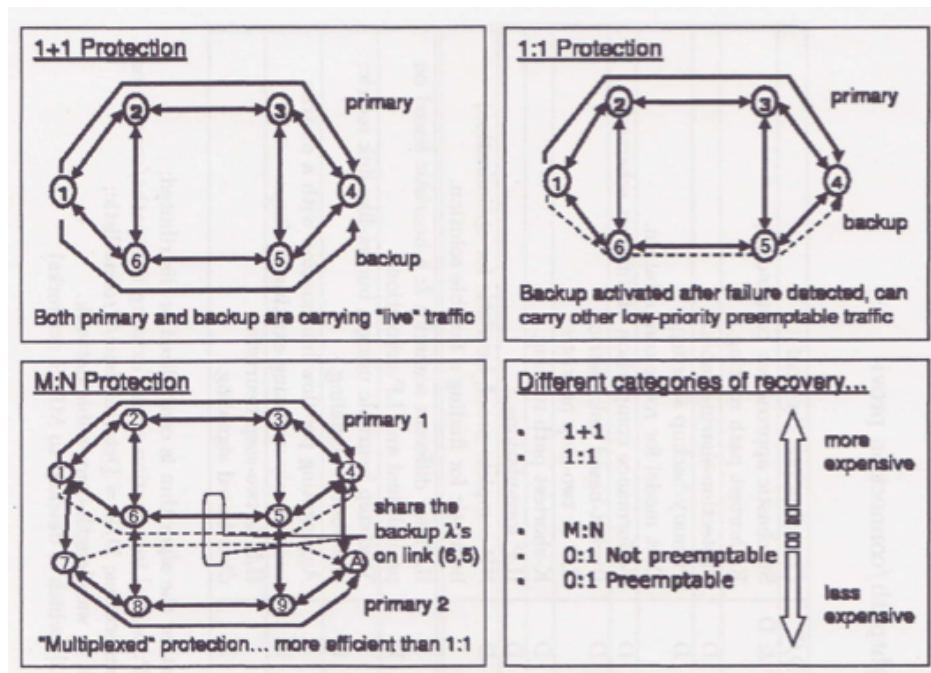


Figura 1.11: Modelos de protección

tion, cuando un enlace cae, los nodos origen y destino de cada conexión son informados del fallo (mediante mensajes de los nodos adyacentes al enlace caído) e "independientemente" descubren una ruta de respaldo. En la figura 1.11 se muestra un esquema resumen de lo comentado en este apartado.

## 1.4. Herramienta MatPlanWDM

MatPlanWDM es una herramienta de planificación de redes WDM wavelength-routing. Hace uso de una topología virtual, una matriz de tráfico y una serie de constantes como el número de transmisores, receptores, etc. como parámetros de entrada. Proporciona varios heurísticos para resolver el diseño de la topología virtual, la matriz de encaminamiento y la matriz de flujos. Además, dispone de varios modos de funcionamiento que se describen a continuación.

### Diseño de topología virtual y encaminamiento de flujos

A través de este modo de funcionamiento, el usuario puede calcular la topología virtual y las matrices de encaminamiento y flujos. Para ello, basta con definir unos parámetros de entrada, cargar un diseño de red (o calcularlo<sup>3</sup>) dando como resultado la topología virtual sobre la que se dibuja el encaminamiento de los distintos flujos.

<sup>3</sup>Para calcular un diseño es necesario especificar la topología física, la matriz de tráfico y el algoritmo de planificación

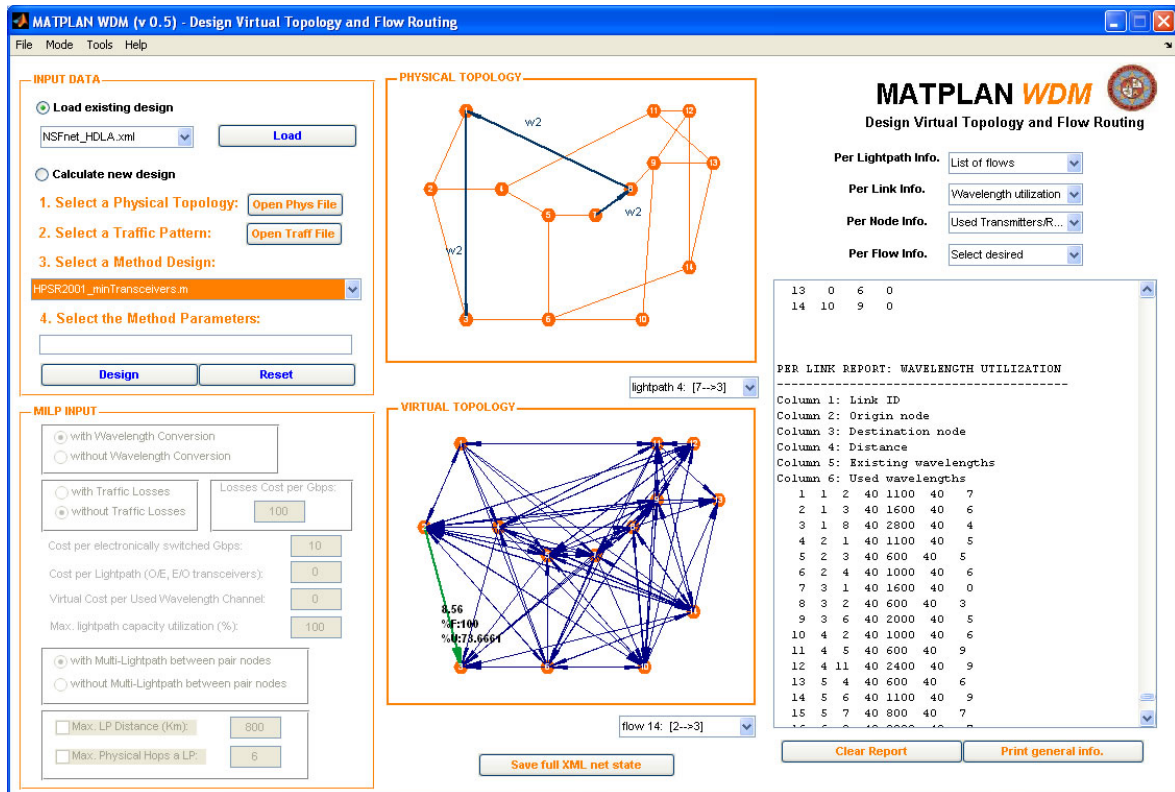


Figura 1.12: Diseño de topología virtual y encaminamiento de flujos

Además, proporciona una serie de parámetros e indicadores de los resultados obtenidos facilitando la evaluación y comparación entre diferentes soluciones. La solución final y los parámetros calculados pueden guardarse en formato XML para posteriormente hacer uso de ellos en este u otro modo.

### Análisis What-If

El análisis What-If permite al usuario lanzar una serie de simulaciones variando los parámetros de entrada durante un rango dado. Para ello, introducimos el parámetro de entrada (número de transmisores, receptores, convertidores, longitudes de onda por fibra, capacidad de los lightpath, factor de multiplicidad y algoritmos de planificación) a variar, el incremento aplicado y el número de muestras a tomar.

Al finalizar la simulación, obtenemos un report con los resultados de cada simulación y una gráfica comparativa.

### Análisis Multihora

El análisis multi-hora permite la evaluación de algoritmos de planificación dinámica que interactúan con matrices de tráfico que varían con las horas del día. Ésto es útil para planificar topologías internacionales que extienden múltiples zonas horarias donde

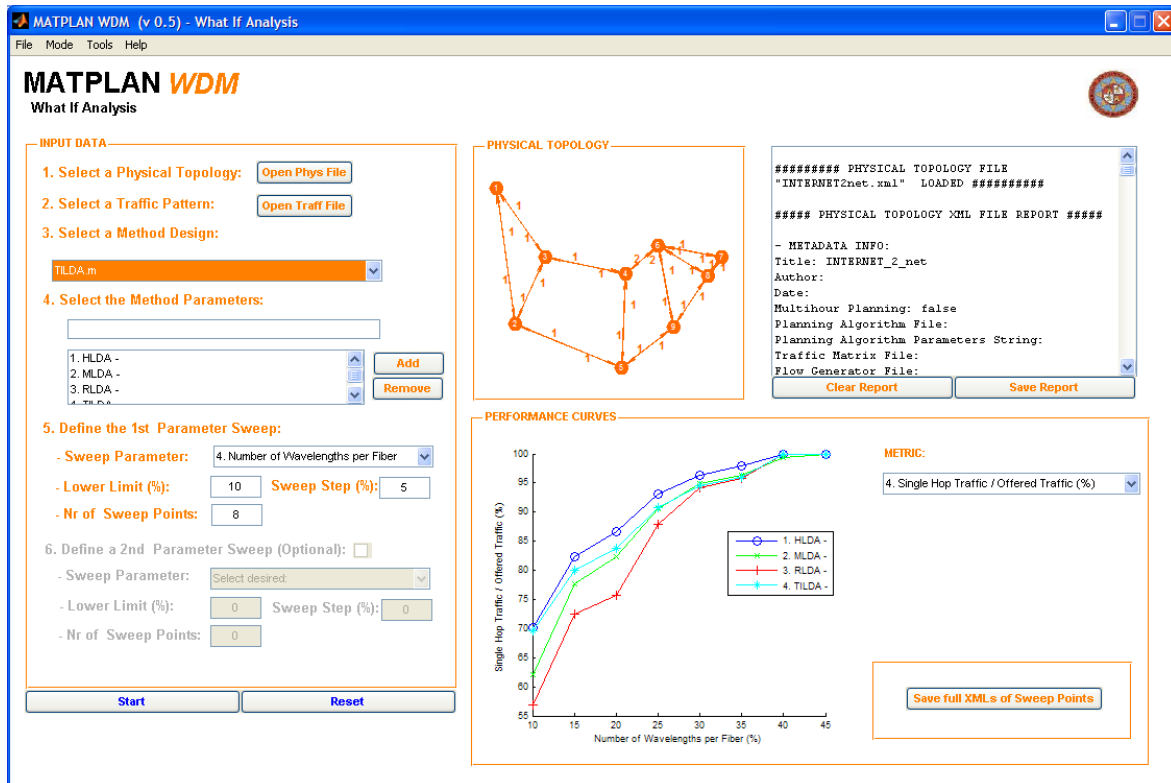


Figura 1.13: What-If Analysis

la actividad de los nodos se ve afectada por la hora su local.

## Análisis dinámico

El modo dinámico nos permite hacer pruebas con algoritmos dinámicos que reaccionan frente a entradas y salidas de flujos. El problema global implica decidir la evolución de la topología virtual y el enrutamiento de flujos.

De manera similar a los otros modos, al finalizar la simulación obtenemos un report con los resultados, y tenemos la posibilidad de guardar el estado de la red en formato XML.

## 1.5. Objetivos del proyecto

Como se ha visto en el presente capítulo de introducción, la tecnología óptica WDM articula 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.

En este marco, el presente proyecto aborda el desarrollo de una extensión de la herramienta MatPlanWDM con fines educativos y orientada a la planificación de



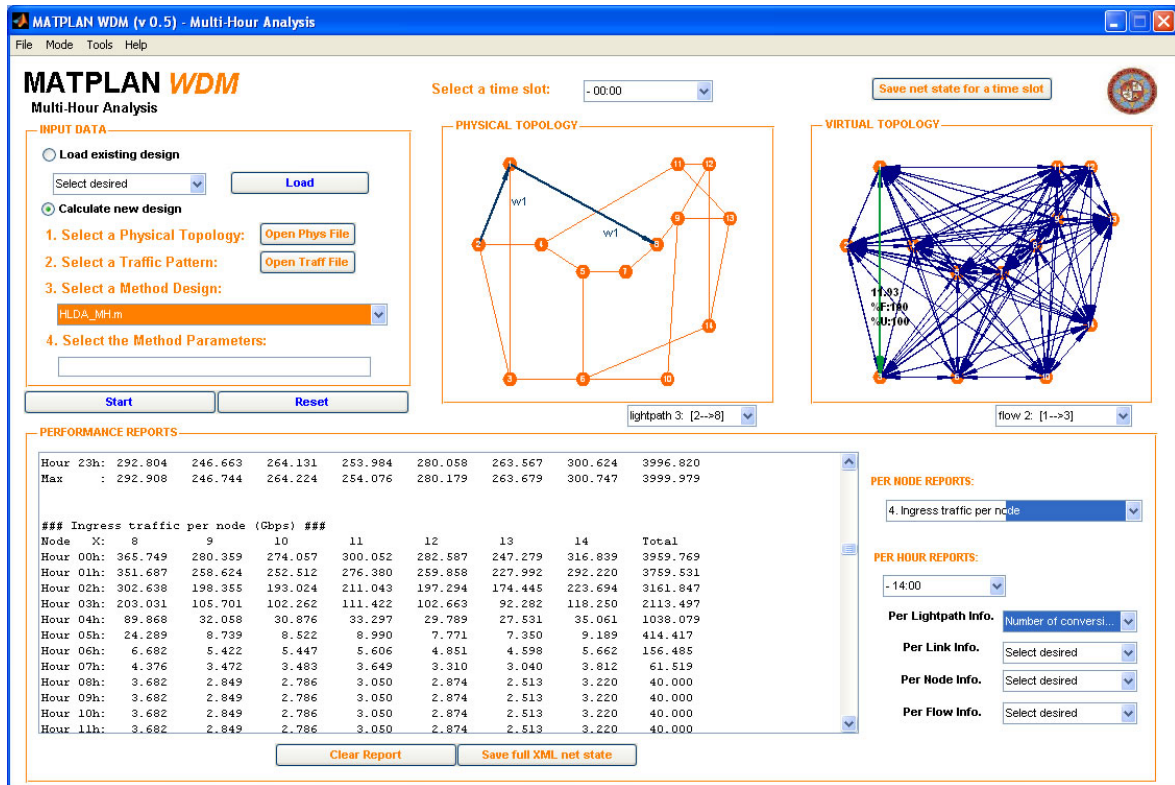


Figura 1.14: Análisis Multihora

redes WDM wavelength-routing. Esta extensión considera demanda estática de tráfico y provee un algoritmo de protección que dinámicamente decide como encaminar el flujo teniendo en cuenta el estado de la red en ese momento, así como la topología física de dicha red. Además también incluye un generador para simular la generación dinámica de fallos. La extensión ha sido implementada sobre MATLAB de forma que pueda ser extendida sin gran esfuerzo, con sólo incluir los algoritmos y generadores en sus correspondientes librerías.

El segundo capítulo se centra en la descripción de la extensión desarrollada. Primero vemos la estructura del toolbox con todas sus funciones documentadas y después se incluye un manual de usuario de la extensión de protección de MatPlanWDM para acercarnos al manejo de la herramienta.

En el tercer capítulo se presentan los resultados obtenidos sobre distintos escenarios, donde se ha querido incluir un amplio abanico de casuísticas que nos permita conocer la herramienta. 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 1. Introducción

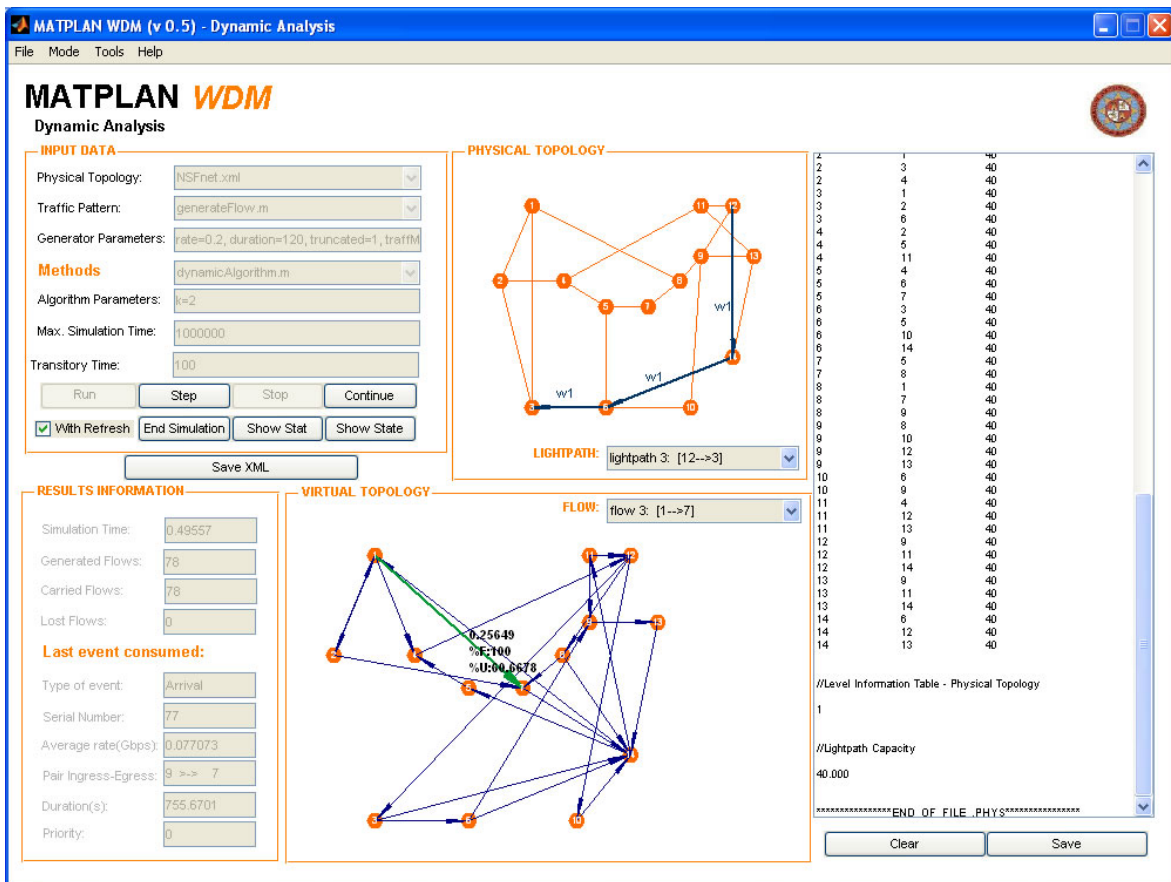


Figura 1.15: Análisis dinámico

# Capítulo 2

## Descripción de la extensión desarrollada

---

### 2.1. Introducción

En el capítulo anterior detallábamos como, hasta ahora, la herramienta MatPlanWDM no consideraba escenarios de protección-restauración y el objetivo principal del proyecto es implementar una extensión que lo permitiera. En el presente capítulo vamos a detallar la extensión desarrollada, algoritmos implementados y un manual de usuario mediante el que se explica el uso (o ampliación) de la aplicación.

### 2.2. Herramienta

#### 2.2.1. Estructura del toolbox

La herramienta Protection-MatPlanWDM ha sido implementada como un toolbox de MATLAB. Todas las funciones del toolbox están documentadas y abiertas al uso. A continuación, procederemos a describir mas detalladamente cada uno de los elementos y procesos de la estructura del toolbox representados en la figura 2.1.

#### 2.2.2. Parámetros de entrada

Los parámetros de entrada necesarios para la simulación son los siguientes:

- Un archivo de Matlab que implementa el módulo generador de fallos.
- Un archivo de Matlab que contiene el módulo de protección al que nos referiremos en adelante como algoritmo PR2.
- Un archivo XML que recoge el diseño de la red obtenido (encaminamiento del tráfico y topología virtual) por un módulo de planificación (Algoritmo PR1) así

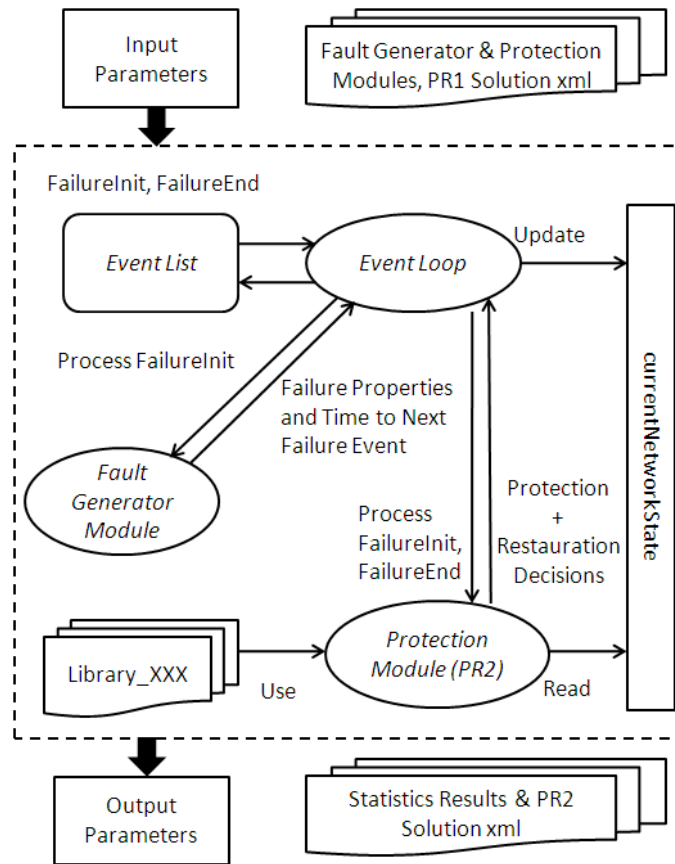


Figura 2.1: Estructura del toolbox

como la topología física de la red (número de Tx/Rx/TWC...) que denominaremos *PR1-netState*.

### 2.2.3. Parámetros de salida

En cualquier momento durante la simulación podemos obtener los siguientes parámetros de salida:

- Un archivo XML con la información del estado de la red.
- Un conjunto de estadísticos que se detallan en la sección 2.2.7.

### 2.2.4. Generador de fallos

La función del módulo generador de fallos es alimentar a la lista de eventos futuros (*Event List*) del módulo central de nuestra aplicación, al que nos referimos en adelante como *eventLoop*. Cuando se produce un evento fallo (*FailureInit*), el módulo generador de fallos recibe los parámetros específicos de dicho generador, la descripción del estado

de la red y la topología física, y da como resultado las propiedades del fallo (tipo de fallo y tiempo de reparación) así como el instante de tiempo en el que se producirá el próximo evento *FailureInit*.

En el manual de usuario (sección 2.3.4) se describirá en profundidad la estructura que debe tener dicho módulo en nuestra aplicación.

### 2.2.5. Algoritmos de protección

#### Módulo de planificación (PR1)

El módulo de planificación o algoritmo PR1 es empleado en la ventana "*Design virtual topology and flow routing*" (Figura 1.12) de MatPlanWDM para calcular un diseño sobredimensionado de la red que permita protección. Este algoritmo recibe una topología física (archivo .phys) y una matriz de tráfico (archivo .traff) y obtiene un archivo XML con el diseño de la red (encaminamiento del tráfico y topología virtual) en el que se habrán considerado las necesidades de protección.

A continuación, mostramos la estructura que debe presentar un módulo PR1. Empezamos por su signatura que debe ser como sigue:

```
[exitMsg exifFlag netState] = PR1Algorithm (trafficmatrix, phys, algorithmParameters)
```

donde los parámetros de entrada son:

- *trafficmatrix*, es una matriz de tráfico donde cada posición  $(i,j)$  recoge el tráfico entre el par de nodos  $(i,j)$ .
- *phys*, es una estructura que contiene variables de la topología física de la red.
- *algorithmParameters*, se trata de una variable de tipo string introducida en la aplicación en el campo *Algorithm Parameters*.

y los parámetros de salida son:

- *exitMsg*, se trata de una cadena de caracteres que permite al algoritmo devolver un mensaje informativo.
- *exifFlag*, es un número que permite a la función informar de posibles errores.
- *netState*, es una estructura que recoge las variables de estado de la red.

El módulo de planificación PR1 correspondiente será el encargado de convertir la variable string *algorithmParameters* a las variables específicas de dicho algoritmo.

### Módulo de protección (PR2)

El módulo de protección, al que también llamaremos algoritmo PR2, es el encargado de tomar las decisiones de protección y restauración. El algoritmo recibe o bien un evento fallo (*FailureInit*), o bien un evento reparación (*failureEnd*).

Cuando se trata de un evento fallo, el algoritmo PR2 toma las decisiones oportunas para proteger los flujos afectados por dicho fallo estableciendo nuevos lightpath cuando sea necesario. En el caso de tratarse de un evento reparación, el módulo deshace los cambios realizados durante el evento fallo para proteger el flujo de datos e intenta devolver el estado de la red a su forma original.

Es importante notar que el módulo de protección sólo toma las decisiones oportunas, y es el módulo *eventLoop* el que realiza las modificaciones sobre la variable que mantiene el estado actual de la red (*currentNetworkState*).

La estructura del algoritmo PR2 se detalla en la sección 2.3.5 del manual de usuario.

### 2.2.6. eventLoop

El eventLoop es el módulo central del toolbox y tiene como misión intercambiar información con el resto de módulos. Nuestro módulo central se comunica con el módulo generador de fallos, mediante el cual es capaz de actualizar la lista de eventos futuros (*Event List*) planificando el próximo evento fallo. Por otro lado, el modulo eventLoop se comunica con el módulo de protección y, con la información recibida, actualiza la variable que contiene el estado actual de la red (*currentNetworkState*).

### 2.2.7. Estadísticos calculados

En cualquier momento durante la simulación, podemos obtener dos tipos de estadísticos:

#### Información sobre la disponibilidad de la red

- Porcentaje de tiempo con tráfico perdido, es la suma de los intervalos de tiempo con tráfico perdido entre el tiempo total de simulación, medido en porcentaje.
- Capacidad de restauración (*Restorability*), se trata de la relación entre el tráfico recuperado exitosamente y el tráfico afectado por los fallos de la red,

$$Restorability = \frac{\sum_{f_{aff-rec} \in B} M_{f_{aff-rec}}}{\sum_{f_{aff} \in A} M_{f_{aff}}}$$

donde A representa el conjunto de flujos de tráfico afectados por la caída de lightpaths debida a los fallos en la red, B es un subconjunto de A que recoge los

flujos de tráfico afectados que han sido exitosamente recuperados y  $M$  representa el tráfico del flujo correspondiente.

- Promedio temporal del número medio de saltos virtuales, donde este número se calcula como la relación entre el tráfico virtual cursado (suma de los tráficos cursados por los lightpaths) y el tráfico total ofrecido (no virtual, es decir, la suma de los flujos entre cada par de nodos).

$$\bar{n}_{vhops} = \frac{\sum_{i=1}^{N_t} n_{vhops_i} \cdot (t_i - t_{i-1})}{T_{sim}}$$

donde  $N_t$  es el número total de intervalos de simulación,  $T_{sim}$  es el tiempo total de simulación y  $n_{vhops_i}$  es el número medio de saltos virtuales en el intervalo de tiempo  $(t_i - t_{i-1})$ .

- Promedio temporal de la tasa de bloqueo, entendida como la relación entre el tráfico perdido y el tráfico ofrecido.

### Información acerca del generador de fallos

- Disponibilidad de la red, se calcula como la relación entre la suma de los intervalos de tiempo en los que no hay ningún fallo en la red y el tiempo total de simulación.
- Disponibilidades de transmisores/receptores/conversores de longitud de onda/longitudes de onda/fibras/ductos, se calculan como la relación entre la suma de los intervalos de tiempo en los que no hay ningún elemento averiado y el tiempo total de simulación.
- Promedio temporal de transmisores/receptores/conversores de longitud de onda/longitudes de onda/fibras/ductos, fallados, se calcula como el promedio temporal del numero de elementos fallados de cada tipo.

### Estructura del cálculo de estadísticos

Las métricas obtenidas como promedios temporales descritas en los apartados previos son calculadas siguiendo la siguiente expresión:

$$\bar{x} = \frac{\sum_i x_i \cdot (t_i - t_{i-1})}{T_{sim}}$$

donde  $\bar{x}$  es el promedio temporal de la métrica,  $x_i$  es el valor de la métrica en el intervalo de tiempo  $(t_i - t_{i-1})$  y  $T_{sim}$  es el tiempo total de simulación.

Así, cada vez que se produce la llegada de un evento las métricas son recalculadas ( $x_i$ ) por la función *accumulateProtectionNetMetrics* para ser multiplicadas por la duración del ciclo  $(t_i - t_{i-1})$  y posteriormente sumadas al valor acumulado. Cuando

## Capítulo 2. Descripción de la extensión desarrollada

la simulación termina cada una de las sumas acumuladas se divide por el tiempo de simulación total ( $T_{sim}$ ) en el que no se tiene en cuenta el tiempo de transitorio. Este punto es llevado a cabo por la función *getProtectionNetMetrics*.

### 2.2.8. Extensión de la herramienta

La herramienta puede ser extendida fácilmente implementando cualquier topología física creando su archivo XML con el formato descrito en el apartado 2.3.3 del manual de usuario y guardándolo en el directorio *physicalTopology*. Otra manera de extender la herramienta sería creando nuevos módulos del tipo Generador de fallos y Algoritmo de protección como funciones de MATLAB con la signatura que es descrita en los puntos 2.3.4 y 2.3.5. Los módulos de tipo *Generador* se deben guardar en el directorio *protectionFailureGenerators* y los de tipo *Planificador* en el directorio *protectionPlanning*.

## 2.3. Manual de usuario

### 2.3.1. Instalación y arranque de la aplicación

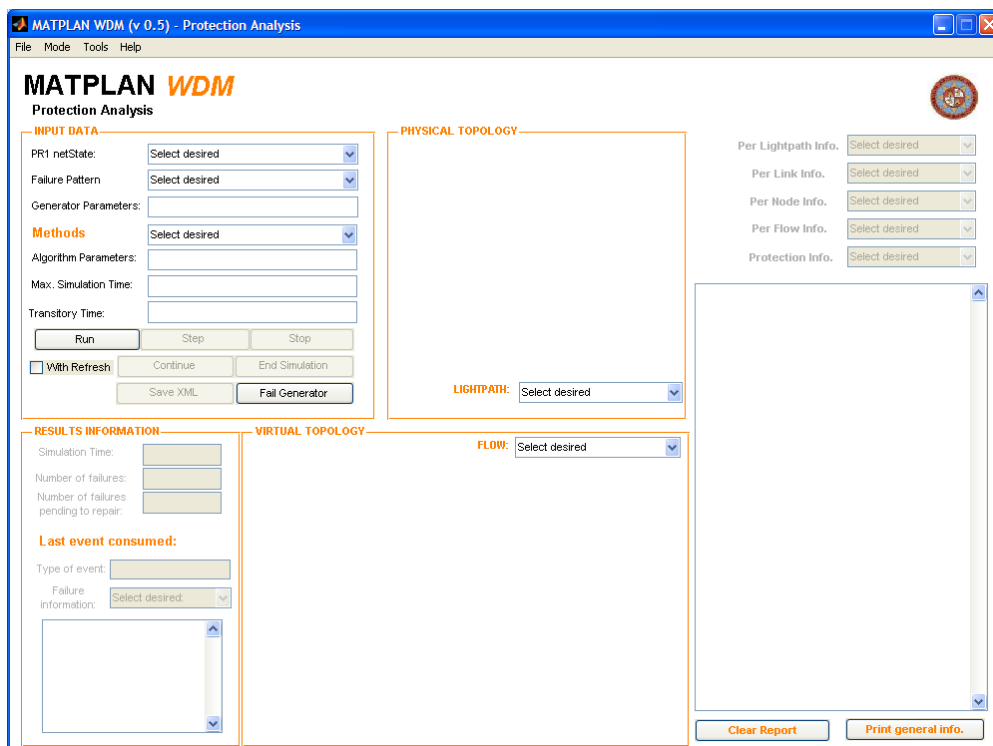


Figura 2.2: Ventana principal

La aplicación MatPlanWDM se instala descomprimiendo el archivo .rar proporcionado. A continuación, será suficiente con seleccionar en la pestaña Current Directory



de MATLAB el path de la carpeta *Protection-MatPlanWDM*.

Una vez seleccionado el directorio correctamente, basta con ejecutar el comando 'startup' en la línea de comandos de MATLAB y se abrirá la ventana principal de *Protection-MatPlanWDM*.

### 2.3.2. Ventana principal

Tras iniciar la herramienta, nos encontramos con la ventana principal de la aplicación tal y como se muestra en la figura 2.2

La aplicación nos recibe con cinco partes diferenciadas que pasamos a describir a continuación.

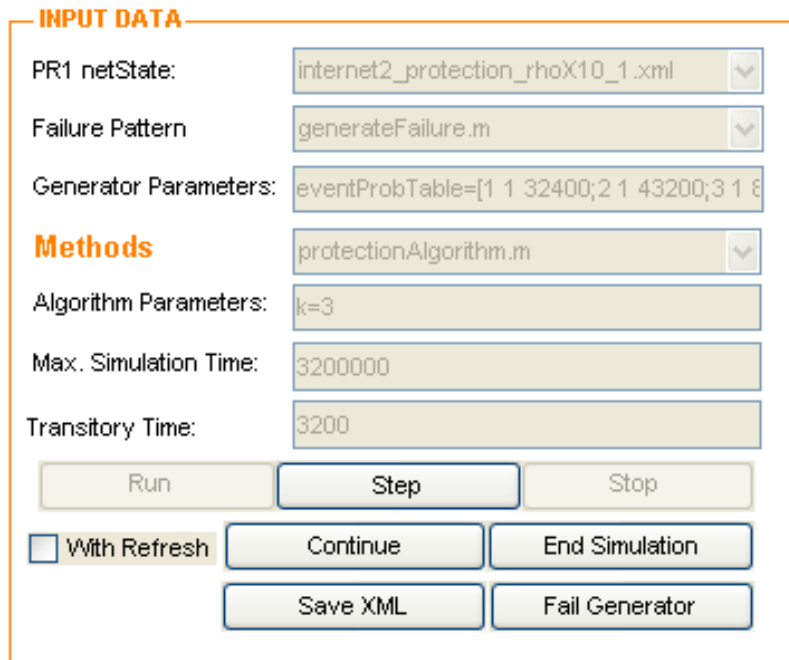


Figura 2.3: Detalle Input Data

El panel *INPUT DATA* nos permite elegir los datos de entrada de la simulación a realizar: el diseño virtual y físico de la topología de red, el generador de fallos, los parámetros de dicho generador, el algoritmo de protección, los parámetros del algoritmo, el tiempo máximo de simulación y tiempo del transitorio.

El menú desplegable *PR1 netState* mostrará las topologías físicas que se encuentran en el directorio physicalTopology entre las que tendremos que elegir una. El siguiente menú *Failure Pattern* nos muestra los generadores de fallos disponibles guardados en la carpeta protectionFailureGenerators. Lo mismo sucede con el menú *Methods* pero en este caso los algoritmos serán los del directorio protectionPlanning.

Para proporcionar correctamente los parámetros de nuestro generador, es necesario conocer el formato que espera e introducirlos en el apartado *Generator Parameters* sep-

## Capítulo 2. Descripción de la extensión desarrollada

---

arados por espacios. Por ejemplo, en la figura 2.3 vemos que para el caso del generador *generateFailure* implementado es necesario introducir los siguientes parámetros:

- *eventProbTable* que presenta tantas filas como tipos de eventos fallo son considerados y en cada fila se recoge: primero, el identificador del tipo de evento fallo; segundo, la probabilidad del fallo, y, tercero, el tiempo medio de reparación (MTTR (Mean Time To Repair)).
- *MTBF* (Mean Time Between Failures) que recoge el tiempo medio entre fallos que será igual para todos los tipos de evento fallo.

De manera similar, en el apartado *Algorithm Parameters* deberemos introducir los parámetros esperados por nuestro algoritmo de planificación. En el ejemplo de la figura vemos que el algoritmo *protectionAlgorithm* tiene como parámetro el valor de k que se refiere al número de caminos más cortos calculados durante la planificación.

Los dos últimos apartados nos permiten elegir el máximo tiempo de simulación y el tiempo transitorio.



Figura 2.4: Results Information

En la parte inferior del panel contamos con los botones encargados de controlar la simulación. El usuario puede comenzar la simulación haciendo click sobre el botón 'Run'. Si el checkbox *with refresh* está activado, la información del panel *RESULTS INFORMATION* se actualizará durante la simulación. Como se muestra en la figura 2.4, la aplicación nos muestra el tiempo de simulación hasta el último fallo, el número de fallos generados hasta ese momento, el número de fallos pendientes de reparar y, con respecto al último evento consumido, el tipo de fallo y un desplegable en el que podremos seleccionar distinta información sobre el fallo (fibras, flujos y lightpaths afectados).

Una vez iniciada la simulación, ésta se puede detener momentáneamente pulsando 'Stop' y puede finalizarse por completo con el botón 'End Simulation'. Cuando se ha pulsado 'Stop', el botón 'Step' se activa dando la posibilidad de consumir un evento cada vez que se hace click en él. Para continuar con la simulación es suficiente con pulsar el botón 'Continue'.

Con la simulación parada, es posible consultar los valores de los estadísticos hasta ese momento haciendo click en 'Print General Info' que mostrará los resultados el apartado situado a la derecha de la ventana principal. Para borrar los datos mostrados en el apartado anterior basta con hacer click sobre el botón 'Clear Report'. Y si lo que queremos es guardar el estado actual de la simulación, es suficiente con pulsar sobre 'Save XML' y elegir nombre de archivo e ubicación.

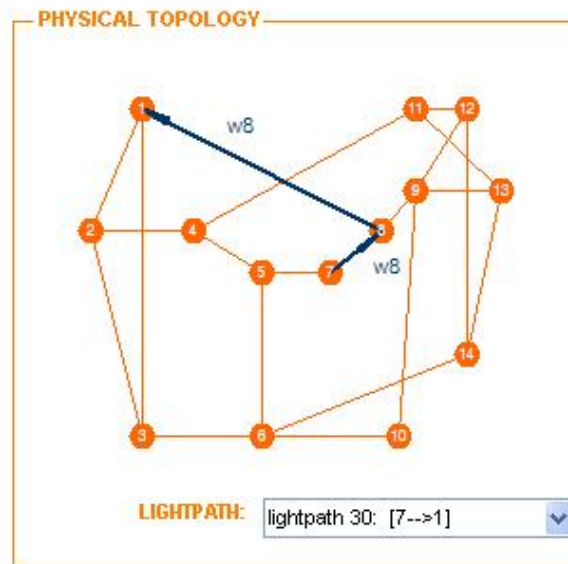


Figura 2.5: Physical Topology

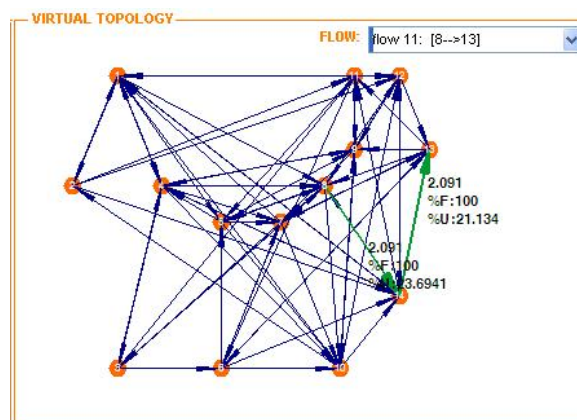


Figura 2.6: Virtual Topology

## Capítulo 2. Descripción de la extensión desarrollada

Cuando la simulación está detenida, podemos ver gráficamente el estado de la red en los paneles *Physical Topology* y *Virtual Topology*. El usuario puede escoger en el menú del panel physical topology el lightpath que quiere representar sobre la topología física. El gráfico mostrará además del lightpath, la longitud de onda que le ha sido asignada en cada salto.

En el ejemplo de la figura 2.5 vemos un lightpath del nodo 7 al 1 sobre la topología física. Este lightpath sigue la secuencia de saltos físicos {7, 8, 1} y la secuencia de longitudes de onda {w8, w8}.

El panel virtual topology representa la topología virtual cuando la simulación está parada y sobre ella podemos representar los flujos existentes. Disponemos de un menú para elegir el flujo a representar y veremos así que lightpaths atraviesa.

En la figura 2.6 vemos que hemos optado por representar el flujo desde el nodo 8 al nodo 13. Este flujo atraviesa la siguiente secuencia de lightpaths {(8,14);(14,13)} que vemos en verde. También se muestra la tasa del flujo en Gbps y la fracción que ocupa en cada lightpath. Si el lightpath sólo es atravesado por ese flujo tendremos un porcentaje Usage de valor 100 %.

The image shows a software interface for generating failures in a network simulation. It is organized into two columns: 'Link failures' and 'Node failures'.  
The 'Link failures' column has three radio buttons: 'Channel failure' (selected), 'Fibre failure', and 'Duct failure'.  
- 'Channel failure' includes a 'LinkId' dropdown (2), a 'wavelength' dropdown (6 (DOWN)), and a 'Mean time to repair' input (100).  
- 'Fibre failure' includes a 'LinkId' dropdown and a 'Mean time to repair' input (100).  
- 'Duct failure' includes a 'DuctId' dropdown and a 'Mean time to repair' input (100).  
The 'Node failures' column has three radio buttons: 'Transmitter failure', 'Receiver failure', and 'TWC failure'. Each includes a dropdown for the specific node or component and a 'Mean time to repair' input (100).  
At the bottom, there are buttons for 'Add to list', 'Remove from list', 'Clear list', and 'Generate'. A list box at the bottom contains one entry: 'Channel failure => linkId: 2, wavelength: 6 (DOWN), Mean time to repair: 100'.

Figura 2.7: Generador de fallos

Por último, en el panel Input Data tenemos acceso a un generador de fallos del que podremos hacer uso en cualquier momento de la simulación accesible mediante el botón 'Fail Generator'. Tras pulsar el botón 'Fail Generator', se nos abrirá una ventana emergente similar a la mostrada en la figura 2.7 en la que es posible seleccionar el tipo

de fallo a generar y las características del mismo: (i) elemento afectado (enlace, nodo, etc.) y (ii) tiempo medio de reparación. Pulsando sobre 'Add to list' se añadirá a la lista en la que podemos añadir tantos como queramos. También podemos eliminar eventos fallo de la lista con sólo pulsar el botón 'Remove from list' o bien borrarlos todos haciendo clic en 'Clear list'. Por último, para generar los fallos elegidos hacemos clic en 'Generate'.

### 2.3.3. Estructura del fichero PR1-netState

Los archivos pr1-netState permiten guardar los datos relativos a la topología física, topología virtual y el encaminamiento de la demanda de tráfico de la red en un archivo XML, que nuestro simulador será capaz de interpretar. En la figura 2.10 se muestra el

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE network (View Source for full doctype...)>
- <network author="" date="23-Aug-2011 13:10:05" description="" flowGeneratorFile="" flowGeneratorParametersString="" multihourPlanning="false"
  planningAlgorithmFile="HLDA_protection" planningAlgorithmParametersString="" title="" trafficMatrixFile="simplenet4.traff">
- <layer id="trafficDemand">
- <flow destNodeId="2" flowAverageRateDemand="0.75" id="1" origNodeId="1" serialNumber="1">
- <flowRouting>
  <lightpathInFlow averageRate="0.75" id="3" />
</flowRouting>
</flow>
- <flow destNodeId="3" flowAverageRateDemand="0.4" id="2" origNodeId="1" serialNumber="2">
- <flowRouting>
  <lightpathInFlow averageRate="0.4" id="7" />
</flowRouting>
</flow>
- <flow destNodeId="4" flowAverageRateDemand="0.5" id="3" origNodeId="1" serialNumber="3">
- <flowRouting>
  <lightpathInFlow averageRate="0.5" id="5" />
</flowRouting>
</flow>
- <flow destNodeId="1" flowAverageRateDemand="0.45" id="4" origNodeId="2" serialNumber="4">
- <flowRouting>
  <lightpathInFlow averageRate="0.45" id="6" />
</flowRouting>
</flow>
- <flow destNodeId="3" flowAverageRateDemand="0.1" id="5" origNodeId="2" serialNumber="5">
- <flowRouting>
  <lightpathInFlow averageRate="0.1" id="6" />
  <lightpathInFlow averageRate="0.1" id="7" />
</flowRouting>
</flow>
```

Figura 2.8: Ejemplo de fichero pr1-netState (trafficDemand)

formato que debe tener este archivo y la información que debe incluir.

Tras un preámbulo obligatorio en todos los archivos XML, comenzamos con la capa etiquetada como *trafficDemand* en la que se recoge para cada flujo su información general (nodos origen y destino, tasa del flujo y número de serie), así como, su encaminamiento a través de los *lightpaths* (*flowRouting*).

A esta capa le sigue la etiquetada como *virtualTopology* la cual guarda como su nombre indica la información relativa a los *lightpaths*, es decir, la topología virtual. Por un lado, recoge el encaminamiento de los *lightpaths* sobre la topología física (*light-pathRouting*) y, por otro, los recursos requeridos en cada nodo (*plannedNodeResources*).

Por último, tenemos la capa *physicalTopology* con la descripción de la topología física de red. Observamos como tras la etiqueta *physicalTopology*, comienza la descripción de cada uno de los nodos de red indicando sus propiedades entre las que cabe

```

- <layer id="virtualTopology">
- <lightpath destNodeId="4" id="1" origNodeId="2" serialNumber="1">
- <lightpathRouting>
  <fibreInLightpath id="3" wavelengthId="1" />
</lightpathRouting>
</lightpath>
- <lightpath destNodeId="3" id="2" origNodeId="4" serialNumber="2">
- <lightpathRouting>
  <fibreInLightpath id="5" wavelengthId="1" />
</lightpathRouting>
</lightpath>
- <lightpath destNodeId="2" id="3" origNodeId="1" serialNumber="3">
- <lightpathRouting>
  <fibreInLightpath id="1" wavelengthId="1" />
</lightpathRouting>
</lightpath>
- <lightpath destNodeId="2" id="4" origNodeId="3" serialNumber="4">
- <lightpathRouting>
  <fibreInLightpath id="1" wavelengthId="2" />
  <fibreInLightpath id="7" wavelengthId="2" />
</lightpathRouting>
</lightpath>
- <lightpath destNodeId="4" id="5" origNodeId="1" serialNumber="5">
- <lightpathRouting>
  <fibreInLightpath id="1" wavelengthId="3" />
  <fibreInLightpath id="3" wavelengthId="3" />
</lightpathRouting>
</lightpath>
...
<plannedNodeResources id="1" numberOfRequiredRxs="8" numberOfRequiredTWCs="0" numberOfRequiredTxS="6" />
<plannedNodeResources id="2" numberOfRequiredRxs="6" numberOfRequiredTWCs="0" numberOfRequiredTxS="4" />
<plannedNodeResources id="3" numberOfRequiredRxs="4" numberOfRequiredTWCs="0" numberOfRequiredTxS="6" />
<plannedNodeResources id="4" numberOfRequiredRxs="6" numberOfRequiredTWCs="0" numberOfRequiredTxS="8" />
</layer>

```

Figura 2.9: Ejemplo de fichero pr1-netState (virtualTopology)

destacar: el número de identificación (*id*), la población (*nodePopulation*), las coordenadas (*xCoord* e *yCoord*), número de transmisores (*eoTransmitter number*), número de receptores (*oeReceiver number*), y número de conversores TWC (*wc number*). A continuación, se describen los enlaces, de manera que cada entrada contendrá la siguiente información: identificador del enlace (*id*), identificador del nodo origen (*origNodeId*), identificador del nodo destino (*fibre destNodeId*), número de longitudes de onda (*numberWavelengths*), longitud del enlace (*linkLengthInKm*) e identificadores de ductos atravesados (*listOfTraversedDuctIds*). Por último, se incluye información de los niveles y la capacidad de los lightpath expresada en Gpbs.

### 2.3.4. Estructura del generador de fallos

En este apartado veremos la estructura que debe tener un módulo Generador de fallos. En primer lugar, su signatura debe ser de la siguiente forma:

$$[timeToNextEvent, arrayOfFailureProperties] = generateFailure(generatorParameters, netState, phys)$$

```

- <layer id="physicalTopology">
- <node id="1" nodeLevel="1" nodeName="" nodePopulation="8393" nodeTimezone="1" xCoord="0.5" yCoord="1.5">
  <eoTransmitter number="10" />
  <oeReceiver number="10" />
  <wc number="0" />
</node>
- <node id="2" nodeLevel="1" nodeName="" nodePopulation="5002" nodeTimezone="1" xCoord="1.5" yCoord="1.5">
  <eoTransmitter number="10" />
  <oeReceiver number="10" />
  <wc number="0" />
</node>
- <node id="3" nodeLevel="1" nodeName="" nodePopulation="6539" nodeTimezone="1" xCoord="0.5" yCoord="0.5">
  <eoTransmitter number="10" />
  <oeReceiver number="10" />
  <wc number="0" />
</node>
- <node id="4" nodeLevel="1" nodeName="" nodePopulation="8127" nodeTimezone="1" xCoord="1.5" yCoord="0.5">
  <eoTransmitter number="10" />
  <oeReceiver number="10" />
  <wc number="0" />
</node>
<fibre destNodeId="2" id="1" linkLengthInKm="1" listOfTraversedDuctIds="1" numberWavelengths="3" origNodeId="1" />
<fibre destNodeId="1" id="2" linkLengthInKm="1" listOfTraversedDuctIds="1" numberWavelengths="3" origNodeId="2" />
<fibre destNodeId="4" id="3" linkLengthInKm="1" listOfTraversedDuctIds="3" numberWavelengths="3" origNodeId="2" />
<fibre destNodeId="2" id="4" linkLengthInKm="1" listOfTraversedDuctIds="1 2" numberWavelengths="3" origNodeId="4" />
<fibre destNodeId="3" id="5" linkLengthInKm="1" listOfTraversedDuctIds="4" numberWavelengths="3" origNodeId="4" />
<fibre destNodeId="4" id="6" linkLengthInKm="1" listOfTraversedDuctIds="2" numberWavelengths="3" origNodeId="3" />
<fibre destNodeId="1" id="7" linkLengthInKm="1" listOfTraversedDuctIds="1" numberWavelengths="3" origNodeId="3" />
<fibre destNodeId="3" id="8" linkLengthInKm="1" listOfTraversedDuctIds="2" numberWavelengths="3" origNodeId="1" />
- <levelInformationMatrix>
  <factor idDest="1" idOrig="1" value="1" />
</levelInformationMatrix>
<lightpathCapacity value="40" />
</layer>
</network>

```

Figura 2.10: Ejemplo de fichero pr1-netState (physicalTopology)

Donde los parámetros de entrada son:

- *generatorParameters*, variable de tipo string introducida en la aplicación en el campo Generator Parameters (ver figura 2.2)
- *netState*, estructura que contiene las variables del estado actual de la red.
- *phys*, estructura que contiene las variables de la topología física de la red.

El módulo se encargará de realizar la traducción de la variable *generatorParameters* según se haya especificado desde el string a las variables correspondientes que espere, obteniendo los siguientes parámetros de salida:

- *timeToNextEvent*, instante de llegada del próximo evento fallo.
- *arrayOfFailureProperties*, estructura que contiene las variables que describen el fallo: (i) tipo de evento fallo y, (ii) el tiempo de reparación de dicho fallo.

Diferentes tipos de eventos son considerados en este generador, en la tabla 2.1 incluimos la clasificación utilizada en nuestra solución.

ID	Elemento fallado	Propiedades del fallo
1	Longitud de onda de un enlace	Longitud de onda e identificador del enlace
2	Fibra completa	Identificador del enlace
3	Ducto	Identificador de ducto
4	Transmisor en un nodo	Identificador del transmisor e identificador del nodo
5	Receptor en un nodo	Identificador del receptor e identificador del nodo
6	TWC en un nodo	Identificador del TWC e identificador del nodo
7	Nodo completo	Identificador del nodo

Tabla 2.1: Propiedades y tipos de fallos

### 2.3.5. Estructura del módulo de protección (PR2)

Un módulo de protección compatible con nuestra herramienta debe seguir la siguiente signatura:

$[actionList, snOfAffectedLps] = protectionAlgorithm(originalPhys, netState, originalNetState, failuresPendingToRepair, arrayOfFailureProperties, algorithmParameters)$

Donde los parámetros de entrada son:

- *originalPhys*, estructura que contiene variables originales de la topología de red.
- *netState*, estructura de variables del estado de la red actualizada.
- *originalNetState*, estructura de variables del estado de la red antes de actualizar (diseño original de la red).
- *failuresPendingToRepair*, estructura que contiene tres campos: (i) *eventTime*, (ii) *eventSerialNumber* y (iii) *arrayOfFailureProperties*.
- *arrayOfFailureProperties*, estructura que contiene las variables que describen el fallo: (i) tipo de evento fallo y, (ii) el tiempo de reparación de dicho fallo.
- *algorithmParameters*, variable con el número de caminos mas cortos a calcular (k).

Con los parámetros recibidos, el módulo tomará las decisiones de protección y restauración, y devolverá los siguientes parámetros de salida:

- *actionList*, array de estructuras con la información de las acciones a realizar.
- *snOfAffectedLps*, vector que contiene los números de serie de los lightpaths afectados por el fallo.

### 2.3.6. Estructura de la variable netState

El parámetro *netState* es una estructura de variables que definen el estado de la red en un instante determinado. La estructura consta de los siguientes campos:



- *netState\_lightpathTable* (Lx2), donde cada fila es un lightpath. La primera columna es el nodo origen del lightpath y la segunda el nodo destino. (L es el número de lightpaths en la red).
- *netState\_lightpathRoutingMatrix* (LxM), donde cada fila i es un lightpath, de manera que un elemento (i,j) es la longitud de onda del lightpath i en la fibra j (sería 0 si el lightpath i no pasa por la fibra j). Consta de tantas columnas como enlaces físicos y debe garantizarse que la fila i de esta matriz haga referencia al mismo lightpath que la fila i de lightpathTable. (L es el número de lightpaths en la red y M es el número de enlaces físicos).
- *netState\_flowTable* (Fx5): Cada fila es un flujo y cada columna representa los siguientes campos para un flujo dado:
  - *Número de serie del flujo*, necesario únicamente para que el evento 'fin de flujo existente' sepa a qué flujo se refiere. Tiene que ser un número consecutivo y único asignado por el bucle principal a cada flujo.
  - *Nodo Origen del flujo*
  - *Nodo Destino del flujo*
  - *Tasa media del flujo*
  - *Instante inicial del flujo*
  - *Duración del flujo*

(F es el número de flujos cursados)

- *netState\_flowRoutingMatrix* (FxL), donde cada fila i es un flujo, de manera que un elemento (i,j) es el número de Gbps del flujo i que circula por el lightpath j. Hay tantas columnas como lightpaths y debe garantizarse que la fila i de esta matriz haga referencia al mismo flujo que la fila i del flowTable. (F es el número de flujos cursados y L es el número de lightpaths en la red).
- *netState.numberOfRequiredTx*s (1xN), vector que contiene el número de transmisores que se necesitan en cada nodo.
- *netState.numberOfRequiredRx*s (1xN), vector que contiene el número de receptores que se necesitan en cada nodo.
- *netState.numberOfRequiredTWC*s (1xN), donde cada elemento i es el número de convertidores TWC disponibles en el nodo i. (N es el número de nodos en la red).
- *netState.arrayOfPerNodeResourcesAssignment*, estructura de tres campos: vectorOfLpSNPerTransmitter, vectorOfLpSNPerReceiver y vectorOfLpSNPerTWC, donde cada campo guarda un vector de tamaño 1xN. El campo vectorOfLpSNPerTransmitter recoge la configuración de transmisores del diseño donde el número de columna se corresponde con el identificador del transmisor y el contenido de esa posición con el número de serie del lightpath que está usando dicho transmisor. Los otros dos campos se configuran de manera análoga para los receptores

en el caso de `vectorOfLpSNPerReceiver` y los conversores para `vectorOfLpSNPerReceiver`.

## 2.4. Algoritmos implementados

### 2.4.1. Generador de fallos

Los parámetros de entrada del generador implementado en este proyecto son los especificados en la sección 2.3.4 del manual de usuario. A continuación, describimos los parámetros específicos de nuestro generador (*generatorParameters*):

- Propiedades de los eventos fallo (*eventProbTable*), se trata de una tabla con tantas filas como tipos de fallos y tres columnas: (i) identificador del fallo, (ii) probabilidad de que ese fallo se produzca y (iii) un escalar con el tiempo medio de reparaciones para ese fallo (asumiendo una distribución exponencial).
- Un escalar con el tiempo medio entre fallos (*MTBF*), asumiendo una distribución exponencial independiente para cada fallo.

En la figura 2.3 del manual de usuario se muestra cómo se deben introducir estos parámetros en la interfaz de usuario.

El generador elegirá aleatoriamente un tipo de fallo y sus propiedades. A continuación, calculará el tiempo de reparación para el fallo obtenido anteriormente. Y por último, obtendrá el tiempo al siguiente evento utilizando el tiempo medio entre fallos proporcionado.

### 2.4.2. Algoritmo PR1

El módulo de planificación o algoritmo PR1 se encarga de obtener un diseño estático sobredimensionado de la red que permite protección. Los parámetros de entrada del algoritmo PR1 implementado son los especificados en la sección 2.2.5 de descripción de la herramienta.

A continuación, se describen los principales pasos realizados por el algoritmo implementado en este proyecto.

1. El algoritmo empieza considerando el par de nodos con mas tráfico ofrecido entre ellos. Mientras que exista tráfico por cursar y recursos físicos (transmisores y receptores) libres, el algoritmo continuará intentando encaminar tráfico.
2. Calculamos los  $k$  caminos más cortos entre el par de nodos con el fin de obtener el enrutamiento del nuevo lightpath. Además, si es posible encontrar una ruta y recursos físicos disponibles, entonces calculamos otra vez los  $k$  caminos más cortos entre el par origen y destino con el fin de obtener el camino para el lightpath de protección.

3. A continuación, actualizamos el encaminamiento del tráfico para mostrar que se ha cursado el flujo ofrecido entre los nodos origen y destino.
4. Si el tráfico entre los nodos es menos que la capacidad de los lightpaths, la capacidad remanente puede utilizarse para encaminar tráfico aún no cursado. Teniendo en cuenta que:
  - a) Para cada par de nodos obtenemos los  $k$  caminos mas cortos empleando una implementación del algoritmo K-shortest path en su versión *path removing* [9] incluida en el directorio libraryGraph del toolbox. con capacidad para soportar el tráfico remanente entre dicho par.
  - b) Si se encuentra una ruta, actualizamos nuestra estructura netState para reflejar que dicho tráfico ha sido cursado.
5. Por último, se obtiene el par de nodos con el máximo tráfico ofrecido que queda por cursar para volver a comenzar el bucle desde el principio.

### 2.4.3. Algoritmo PR2

En este proyecto se implementa un algoritmo de planificación al que denominamos PR2 que aplica protección y restauración. Tal y como se describe en el apartado 2.2.5, el algoritmo recibe un diseño estático en formato XML obtenido por un algoritmo PR1, así como una lista de eventos *fallo-reparación* que suceden simultáneamente, y obtiene una lista de acciones a tomar en función del tipo de evento.

A continuación, se describen los principales pasos que realiza el algoritmo ante un evento:

1. En primer lugar se obtiene el estado actual de la red, así como, la topología física actualmente operativa a partir del diseño estático original y de la lista de fallos pendientes de reparar.
2. Según el evento recibido por el algoritmo (*FailureInit* o *FailureEnd*) tomamos diferentes decisiones:
  - a) Cuando se trate de un evento *FailureInit*, el primer paso es obtener una lista de lightpaths afectados por dicho evento. Cada posición de la lista es procesada individualmente para obtener la relación de flujos que también se ven afectados por el fallo.  
Con la relación de lightpaths obtenida, añadimos a la lista resultante el primer conjunto de acciones a realizar, es decir, eliminamos los lightpaths correspondientes.  
A continuación, intentamos recuperar tantos lightpaths afectados como sea posible. Con este fin, se procesa uno a uno cada lightpath afectado y se prueba a crear un nuevo lightpath entre los nodos origen y destino del lightpath original afectado, obteniendo los  $k$  caminos más cortos entre dicho par,

donde  $k$  es un parámetro de entrada de nuestro algoritmo PR2. Cuando se encuentra un posible camino, el nuevo lightpath se establece añadiéndose a la lista resultante la acción correspondiente.

Finalmente, se procesa la lista de flujos afectados uno a uno comprobando si todos los lightpaths atravesados han sido recuperados. Cuando la ruta atravesada por el flujo es recuperada por completo, se añade una nueva acción a la lista resultante que recoge el nuevo encaminamiento. En caso de no ser posible la recuperación, también se añade una nueva acción a la lista para reflejar en el estado de la red que el flujo deja de cursarse al 100 %.

- b) En el caso de un evento *FailureEnd*, se obtiene la lista de los lightpaths de protección existentes y la lista de flujos que son atravesados por dichos lightpaths. A continuación, se añade el primer conjunto de acciones a la lista resultante: (i) cambio de encaminamiento de los flujos relacionados, y (ii) eliminación de los lightpaths de protección.

Según el tipo de evento *FailureEnd* recibido, se obtiene la lista de lightpaths originales afectados. Dichos lightpaths son procesados uno a uno para comprobar si su ruta original esta de nuevo disponible por completo. En caso afirmativo, se añaden las acciones correspondientes a la lista resultante para volver a establecer los lightpaths originales.

A continuación, procesamos uno a uno la lista de lightpaths originales recuperados actualizando de nuevo el encaminamiento de los flujos que atravesaban dicho lightpath, es decir, añadiendo la acción correspondiente a la lista resultante.

Seguidamente, intentamos recuperar tantos lightpaths que siguen afectados (por otros fallos) como sea posible. Con este fin, se procesa uno a uno cada lightpath que continúa afectado y se prueba a crear un nuevo lightpath entre los nodos origen y destino del lightpath original afectado, obteniendo los  $k$  caminos más cortos entre dicho par, donde  $k$  es un parámetro de entrada de nuestro algoritmo PR2. Cuando se encuentra un posible camino, el nuevo lightpath se establece añadiéndose a la lista resultante la acción correspondiente.

Por último, se actualiza el encaminamiento virtual de los flujos recuperados por los lightpaths de protección establecidos incluyendo las acciones necesarias en la lista resultante.

# Capítulo 3

## Resultados obtenidos

---

### 3.1. Introducción

En este capítulo se presentan y analizan los resultados obtenidos mediante los algoritmos propuestos bajo diferentes escenarios. Todos los algoritmos se han implementado en código MATLAB, integrados en la extensión desarrollada para este Proyecto Fin de Carrera de la herramienta MatPlanWDM [3] que interactúa con TOMLAB/CPLEX solver [4].



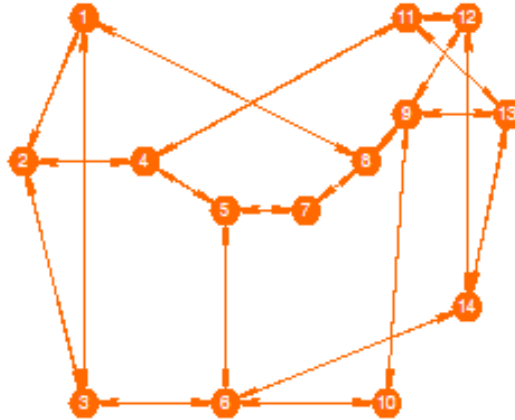
Figura 3.1: Topología de Internet2

### 3.2. Escenarios

Dos topologías de referencia, con sus correspondientes matrices de tráfico se han empleado en estas pruebas: Internet2 [5] que consta de 9 nodos y 26 enlaces unidirec-

cionales y NSFNET [6] con 14 nodos y 42 enlaces unidireccionales. En las figuras 3.1 y 3.2 se muestra la topología de cada una de ellas.

El número de longitudes de onda por fibra elegido para las pruebas es de 20, tanto para Internet2 como para NFSNET. La capacidad del lightpath elegida para el estudio es de 10Gbps.



**Figura 3.2:** Topología de NSFNET

En estas pruebas hemos considerado diferentes cargas de tráfico mediante la variable  $\rho$  tomando valores entre 0.1 para referirnos a cargas de tráfico más bajas y 0.8 para cargas de tráfico altas. Dada una topología de red, la matriz de tráfico de referencia de dicha topología  $T^{BASE}$  medida en Gbps, un número de longitudes de onda por fibra  $\omega$  y un factor de carga de tráfico  $\rho$ , las demandas de tráfico empleadas son calculadas como sigue. En primer lugar, se obtiene el factor multiplicativo máximo  $\alpha^{MAX}$ . Para este factor se cumple que la matriz de demanda de lightpaths obtenida aplicando el algoritmo presentado en [7] al tráfico  $T^{MAX}$  admite una solución factible para una instancia del problema conocido como 'Integral multicommodity flow' [8]. Ésto garantiza que la demanda  $T^{MAX}$  medida en Gbps puede ser completamente cursada por la red con un 0% de bloqueo.

$$T^{MAX} = \alpha^{MAX} * T^{BASE}$$

En este estudio se ha asociado el factor  $\rho = 1$  a la matriz de demanda  $\frac{T^{MAX}}{2}$  con el fin de no sobrecargar en exceso la red para disponer de suficientes recursos para poder recuperarse ante sucesivos fallos.

Por tanto, el conjunto de matrices de tráfico empleadas en las pruebas son calculadas utilizando la siguiente expresión:

$$T^{\rho} = \rho * \alpha^{MAX} * \frac{T^{BASE}}{2}$$

donde  $\rho$  tomará valores del conjunto  $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ .

Utilizamos los algoritmos PR1 y PR2 propuestos en este Proyecto ambos con los parámetros de entrada  $k=3$  caminos mas cortos empleando una implementación del algoritmo K-shortest path [9]. Con respecto al generador de fallos, no consideraremos posibles los fallos de nodo completo por ser de escasa probabilidad y, en caso de producirse, alterarían considerablemente los datos obtenidos. El resto de eventos Fallo se consideran equiprobables y sus valores de MTTR son extraídos de la referencia [11]. El valor del parámetro MTBF lo hemos ido variando entre 3 valores posibles para comprobar como afecta a los resultados:  $\{30, 100, 180\}$  días. El tiempo total de simulación considerado es de un año completo (360 días), y el tiempo transitorio ha sido calculado como la milésima parte de un año.

### 3.3. Resultados obtenidos

A continuación se muestran las gráficas que representan el promedio temporal del bloqueo en función de la carga para las dos redes elegidas y los escenarios establecidos previamente. Recordemos que el promedio temporal del bloqueo se calcula como el promedio temporal de la relación entre el tráfico perdido y el tráfico afectado por los fallos que se producen en la red.

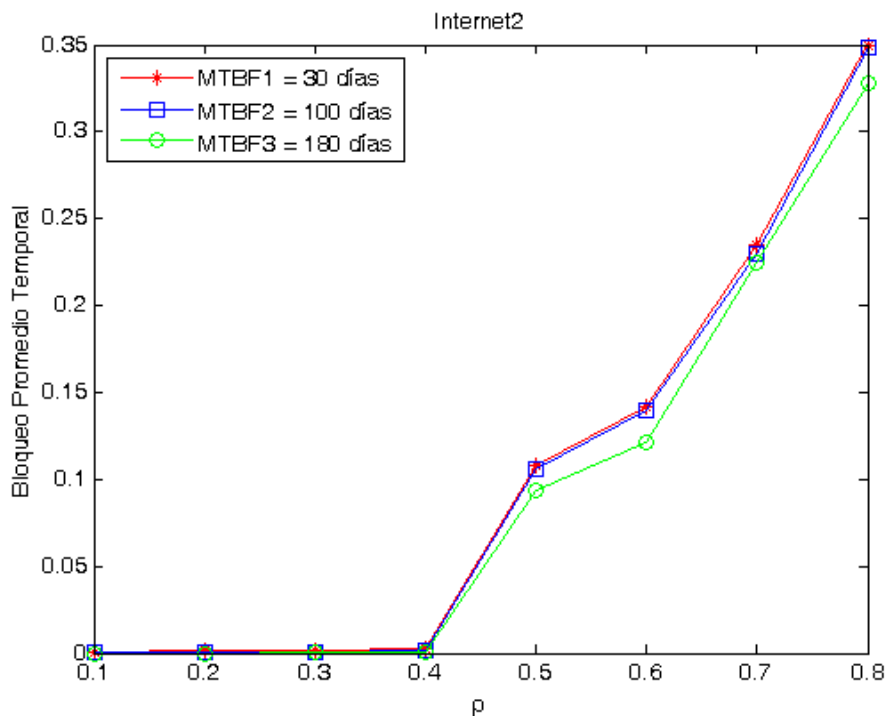


Figura 3.3: Bloqueo en Internet2

En la figura 3.3 podemos comprobar como el promedio temporal del bloqueo crece

de forma casi exponencial cuando la carga del tráfico aumenta. Este crecimiento era esperado ya que al aumentar la carga de tráfico en la red el efecto de los fallos será muy superior al no poder recuperarse el tráfico afectado por estar la red más cargada.

En cuanto a las variaciones obtenidas para los diferentes valores del MTBF, es claro que las diferencias son prácticamente nulas para los valores de 30 y 100 días mientras que en el caso de 180 días los valores son ligeramente inferiores a partir de  $\rho=0.5$ .

Es razonable que al aumentar el tiempo promedio entre fallos (MTBF) el valor del promedio del bloqueo temporal se vea reducido ya que al espaciarse los fallos en el tiempo cuando se produce un nuevo evento fallo la red ya estará recuperada de posibles fallos anteriores. Sin embargo, es posible apreciar que el promedio del bloqueo temporal es mucho más sensible a las variaciones del tráfico que al valor del parámetro MTBF.

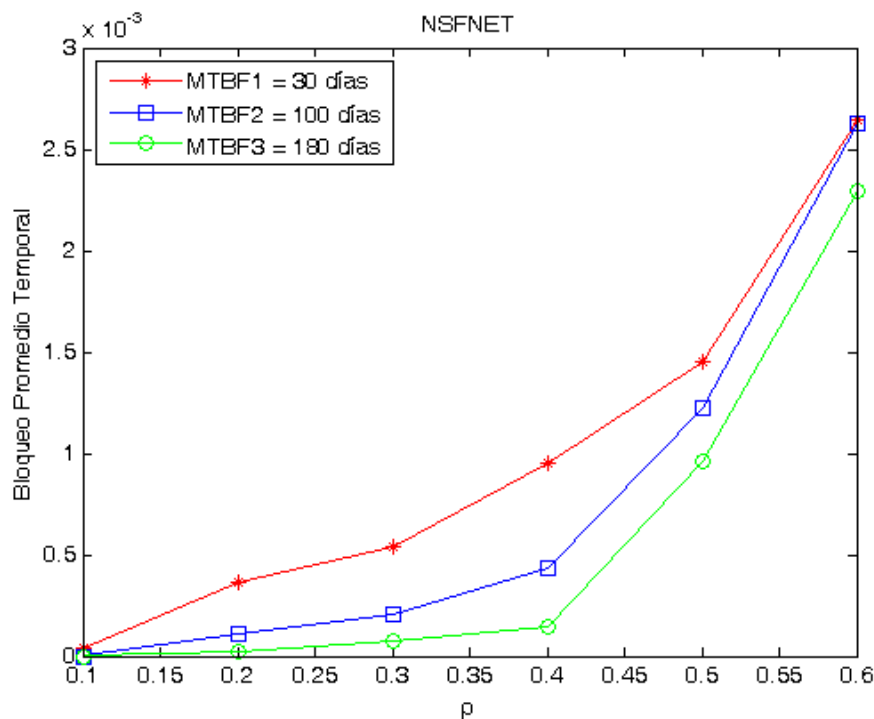


Figura 3.4: Bloqueo en NSFNET

En la figura 3.4 se recogen los resultados obtenidos para NFSNET. Tal y como muestra la representación, el patrón de crecimiento en este caso es similar al obtenido para la red Internet2 y se justifica por el efecto que causa el aumento progresivo del tráfico. Si nos fijamos ahora en las diferencias obtenidas para los distintos valores del parámetro MTBF, comprobamos que aunque las líneas apenas coinciden, hay que destacar que las diferencias se encuentran en valores del orden de  $10^{-3}$ . Por tanto, en este caso el efecto del parámetro MTBF sigue siendo débil comparado con las consecuencias del aumento de la carga del tráfico.



## Capítulo 4

# Conclusiones y líneas futuras

---

### 4.1. Conclusiones y líneas futuras

Actualmente las redes de fibra óptica constituyen uno de los sectores de mayor desarrollo en el sector de las comunicaciones. Ofrecen una serie de prestaciones frente a los medios de transmisión tradicionales (elevados anchos de banda, bajísimas tasas de error, bajos niveles de atenuación y distorsión de señal, bajos requerimientos de potencia,...) que las hacen especialmente indicadas como soporte físico para redes de troncales de larga distancia. Pero a la hora de la aplicación práctica de esta tecnología en redes de comunicación se nos plantea el problema del llamado 'cuello de botella electrónico'. Este cuello de botella consiste en el desajuste existente entre las elevadas tasas de transmisión que la fibra óptica permite (tasas de hasta 40 gigabits por segundo (OC-768) para un canal sencillo de longitud de onda en sistemas SONET-SDH de 160 longitudes por fibra) y la baja velocidad de procesamiento electrónico a la que los datos son tratados en los nodos de la red (nodos de enrutamiento o nodos de acceso de un usuario final). Esto significa que el factor que limita actualmente las tecnologías basadas en la fibra óptica no es el ancho de banda de los enlaces de fibra, sino las limitaciones de velocidad del equipamiento electrónico de los nodos.

Para superar el cuello de botella electrónico se han propuesto diferentes soluciones, pero una de las más interesantes son las redes WDM (Wavelength Division Multiplexing) de conmutación de lightpaths, también llamadas redes WR (Wavelength Routed). Estas redes WDM intentan eliminar o disminuir la necesidad de procesamiento electrónico en los nodos intermedios de enrutamiento de la red (no se puede eliminar el procesamiento electrónico en los nodos de acceso de la red). Para disminuir esta necesidad de equipamiento electrónico, se establece una serie de rutas continuas de longitud de onda entre los nodos, sin requerimiento de conversión óptica- electrónica-óptica y almacenamiento electrónico en los nodos intermedios de las rutas. Estas rutas son llamadas lightpaths y son, por tanto, caminos de comunicación todo-óptico, con lo que se elimina la necesidad de electrónica en los nodos intermedios, ya que la conmutación y el enrutamiento será realizado por conmutadores ópticos. En cualquier caso, puede resultar que un flujo de tráfico deba cambiar de lightpath para llegar a su destino,

en tal caso, en el nodo donde conmute de lightpath tendrá lugar un procesamiento electrónico y las necesarias conversiones opto electrónicas. Estos nodos son también los puntos de grooming de la red (los nodos donde se puede introducir o sacar datos de los lightpaths). El conjunto de todos los lightpaths establecidos sobre una red recibe el nombre de topología virtual. Además, las redes WR deben ser capaces de gestionar los fallos en los distintos recursos de red que la componen.

La pregunta que nos podemos hacer llegados a este punto es cómo protegemos los lightpaths afectados por los posibles fallos de nuestra red con el fin de evitar en el mayor grado posible el bloqueo del tráfico. El objetivo del presente Proyecto Fin de Carrera es el análisis del impacto de fallos físicos en los elementos de la red que afectan a dichas topologías virtuales sobre la topología física real de nuestra red. Para realizar dicho estudio hemos implementado un módulo de la herramienta MatplanWDM mediante código Matlab. Quizá en términos de tiempo de computación no sea el entorno idóneo, pero por otra parte nos ofrece una amplia funcionalidad matemática que implica la implementación de nuevos algoritmos y generadores sea más sencilla. Además la herramienta se puede extender con gran facilidad con sólo añadir las nuevas funciones con la signatura adecuada en los correspondientes directorios. Cabe decir también que el lenguaje empleado por MATLAB es sencillo con lo que requiere menos entrenamiento para el usuario. Todas estas razones hacen que este entorno se haya elegido para el desarrollo del Proyecto frente a otras opciones como la implementación en C++.

En cuanto a las posibilidades futuras, se pueden dirigir esfuerzos al desarrollo de algoritmos de protección con la finalidad de poder obtener comparativas. Así mismo, sería interesante avanzar en la interfaz gráfica para el soporte de esta herramienta en la docencia de contenidos asociados a la optimización en redes. Otra posible línea futura es la utilización de la *MatPlanWDM-Protection* para la investigación de algoritmos de protección mas complejos, y su posterior evaluación.

# Bibliografía

---

- [1] C.S.R. Murthy and M. Gurusamy, *WDM Optical Networks (Concepts, Design and Algorithms)*. Upper Sadle River, NJ: Prentice Hall PTR, 2002, ch 2.
- [2] *Cygwin information and installation, Home Page*. Página web, 2008.  
<http://www.cygwin.com/>
- [3] P. Pavon-Marino, R. Aparicio-Pardo, G. Moreno-Munoz, J. Garcia-Haro, and J. Veiga-Gontan, *MatPlanWDM: An educational tool for network planning in wavelength-routing networks*, in IFIP ONDM, Athens, Greece, 2007, pp. 58-67.
- [4] K. Holmstrom, A. O. Goran, and M. M. Edvall, User's Guide for TOMLAB-CPLEX v11.2 2009.
- [5] Internet 2 Global Research Network Operations Center [Online].Available: <http://www.abilene.iu.edu/>
- [6] R. Ramaswami and K. N. Sivarajan, 'Design of logical topologies for wavelength-routed optical networks', *IEEE J. Select. Areas Commun.*, vol. 14, no. 5, pp. 840-851, Jun. 1996.
- [7] V.R. Konda et al., 'Algorithm for traffic grooming in optical networks to minimize the number of transceivers,' in Proc. IEEE Workshop on High Performance Switching and Routing, Dallas, TX, 2001, pp. 218-221.
- [8] L. Kleinrock, *Queueing Systems: Computer Applications*. New York: Wiley, 1976, vol. 2.
- [9] E.Q. Martins, 'An algorithm to ranking paths that may contains cycles', *European Journal of Operational Research*, vol. 18, pp. 123-130, 1984
- [10] Biswanath Mukherjee, 'Optical WDM Networks', Springer, 2006
- [11] Sofie Verbrugge UGent, Didier Colle UGent, Piet Demeester UGent, Ralf Huelsermann and Monika Jaeger, 'General availability model for multilayer transport networks', New York, NY, USA: IEEE, 2005