

# OpenSimMPLS: Herramienta para la Innovación Docente e Investigación en Redes y Comunicaciones<sup>U</sup>

F. Rodríguez-Pérez<sup>1</sup>, M. Domínguez-Dorado<sup>1</sup>, J. L. González-Sánchez<sup>1</sup>, J. L. Marzo Lázaro<sup>2</sup>, A. Gazo-Cervero<sup>1</sup>

<sup>1</sup>Dpto. de Informática. Universidad de Extremadura, Avda. de la Universidad s/n, CP-10071 Cáceres

<sup>2</sup>Institut d'Informàtica i Aplicacions, Universitat de Girona, Girona

fjrodri@unex.es, ingeniero@manolodominguez.com, jlgs@ex.es, marzo@eia.udg.es, agazo@unex.es

**Abstract.** MPLS (Multiprotocol Label Switching) provides interesting mechanisms to integrate network technologies like ATM and IP with Quality of Service. It is a last generation technology that presents great interest for teaching at the university. In this project, a simulator, as a didactic resource for MPLS teaching innovation, is presented. It allows to the student to configurate, interact and analyze the operation of a MPLS domain in a simple and efficient way. On the other hand, due to its free software license, it can also be used as a protocol engineering platform for creating of analysis and results validation tools by researchers who work in MPLS related projects.

## 1 Introducción

MPLS (Multiprotocol Label Switching) [1] es una tecnología orientada a conexión que surge para orientar los problemas que plantean las redes actuales en cuanto a velocidad, escalabilidad e ingeniería de tráfico [2]. Al mismo tiempo ofrece Calidad de Servicio (QoS) extremo a extremo, mediante la diferenciación de flujos y reserva de recursos. Por otro lado, elimina el problema de la gestión de los diferentes planos de control que tienen lugar en redes IP/ATM, proporcionando mecanismos para conseguir la convergencia entre ambas tecnologías.

MPLS actúa como nexo entre los protocolos de red y el correspondiente protocolo de nivel de enlace. Para ello, en la estructura de una trama, la cabecera MPLS se situará después de la cabecera de nivel de red y antes de la cabecera de nivel de enlace [3]. De hecho, el reenvío de paquetes MPLS está basado en etiquetas y no en el análisis de los datos que encapsulan (protocolos de nivel de red).

Es una tecnología multiprotocolo, como su propio nombre indica; admite cualquier protocolo de red, pero al mismo tiempo permite cualquier tecnología de capas inferiores (enlace o físico). De esta forma, se ha proporcionado un atractivo mecanismo para aprovechar la infraestructura actualmente desplegada en ámbitos troncales, facilitando así la migración de tecnologías [4]. Los esfuerzos realizados desde hace años para desarrollar mecanismos innovadores que den soporte a IP sobre ATM no se han perdido, ya que la mayoría de las técnicas desarrolladas serán válidas para disponer de IP sobre MPLS y MPLS sobre ATM.

En el presente artículo se presenta un simulador MPLS al que hemos denominado OpenSimMPLS. Es una herramienta funcional y visual que puede utilizarse en la docencia de asignaturas de redes y/o comunicaciones [5]. Contempla los aspectos

fundamentales de funcionamiento y configuración de un dominio MPLS [6]; al mismo tiempo ha sido mejorado al incluir compatibilidad con dominios que soporten Garantía de Servicio (GoS) [7]. Un dominio MPLS con capacidad GoS puede entenderse como un entorno capaz de llevar a cabo recuperaciones locales de paquetes descartados junto con la posibilidad de recomponer localmente LSPs (Label Switched Paths) [8]. Éste es el ámbito actual de nuestra investigación y OpenSimMPLS está siendo empleado para validar los resultados obtenidos. En [9] y [10] podemos encontrar otros simuladores MPLS que permiten el diseño y configuración de los componentes de un dominio, así como la simulación y análisis estadístico de los resultados, todo ello desde un punto de vista docente. En el caso de OpenSimMPLS, también serán usuarios del simulador los investigadores de proyectos MPLS con necesidad de comprobar la bondad de sus resultados. Para flexibilizar el futuro empleo del simulador por diferentes investigadores, el software es multiplataforma y está liberado bajo licencia GPL v2.0 de *Free Software Foundation*.

En la siguiente sección se hace una breve descripción del entorno visual del simulador, así como de algunos aspectos funcionales. En el tercer apartado se hace hincapié en la importancia de OpenSimMPLS en entornos docentes o para la validación de resultados de investigación. En el cuarto apartado se comentan algunos detalles generales relativos a su programación. Finalmente, el artículo concluye indicando las contribuciones del simulador.

## 2 Aplicaciones docentes e investigadoras de OpenSimMPLS

OpenSimMPLS es una herramienta multiplataforma, cuyo objetivo fundamental es el de servir al profesorado universitario como recurso didáctico para innovar en la docencia y análisis del funcionamiento de redes MPLS. El simulador dispone de una interfaz gráfica que permite un

entorno de usuario simple. La programación de cada uno de los elementos que componen la aplicación está orientada a objetos; además genera procesos que funcionan de manera concurrente mediante hilos independientes, lo cual permite el estudio de los distintos eventos de una forma más flexible. En cuanto a su funcionamiento, el simulador dispone de tres niveles de operación: el primero trata el diseño y configuración de la topología de un dominio MPLS; el segundo considera la exploración visual de los diferentes eventos que van sucediéndose durante la simulación; y el tercero permite la evaluación de las prestaciones del dominio MPLS diseñado.

El alumno de asignaturas de redes y comunicaciones refuerza su aprendizaje gracias a ejemplos prácticos, ya que el simulador ofrece resultados sobre el comportamiento de la red al introducir servicios particulares como, por ejemplo, tráfico multimedia. De esta forma puede apreciarse cómo se priorizan unos flujos con respecto a otros, permitiendo contrastar resultados gracias al sistema de reconfiguración de los dispositivos previamente definidos en el dominio. También se puede modificar la topología, incorporando otros elementos para identificar los parámetros físicos de diseño de la red que garantizan una mayor QoS. De esta forma el alumno puede realizar propuestas para la mejora de supuestos de redes MPLS, con el objetivo de establecer los criterios mínimos de calidad de servicio. Así, el estudiante puede analizar y diseñar la integración de servicios de banda ancha (multimedia) en redes MPLS, para conocer el comportamiento de éstas, la calidad del servicio ofrecida y dónde se producen los efectos más perniciosos sobre el tráfico, con el objetivo de aprender a evitarlos en lo posible.

En cualquier simulación MPLS, el alumno resuelve problemas y situaciones, aprende procedimientos, llega a entender las diferentes características de los eventos, aprende cómo controlarlos y qué acciones realizar en situaciones particulares. Se puede emplear OpenSimMPLS de forma que el estudiante se trace hipótesis basadas en su experiencia y conocimientos teóricos acumulados, a modo de síntesis o repaso de lo que ya ha estudiado. Tiene la posibilidad de poner en práctica sus ideas, obtiene información de retorno inherente a las respuestas del simulador, las cuales debe descifrar para saber qué ocurre en el interior del dominio y determinar cuál es la norma o principios que rigen su funcionamiento. Este proceso experimental y analítico es el que ayuda al estudiante a desarrollar sus propias estrategias de pensamiento. Un ejemplo de uso del simulador en el aula podría consistir en proponer el diseño de dominios MPLS que presenten situaciones conflictivas de congestión. Dicha problemática deberá ser detectada por el alumno durante la simulación del sistema, así como en el Área de Análisis de resultados (estadísticas de paquetes entrantes, descartados, salientes, etc.). El alumno también deberá plantear medidas para solventar la problemática localizada: modificación de la topología, cambios en las características de los

tráficos, empleo de técnicas de priorización de flujos, etc. De igual forma, podrá comprobar cómo las variaciones introducidas contribuyen a mejorar el rendimiento final del sistema.

OpenSimMPLS permite al usuario variar múltiples parámetros de configuración para intervenir directamente en las operaciones simuladas en cualquier instante, ofreciendo una visión interactiva de los diferentes eventos. De esta forma, la característica más importante de este simulador interactivo es permitir al estudiante ser un agente activo durante la simulación, profundizar en los procesos simulados de funcionamiento MPLS, para luego saber interpretar y manipular los resultados que se obtengan durante la ejecución. El alumno, además de participar en la situación, debe ser también capaz de procesar la información que el simulador le proporciona, propiciando el conocimiento de tipo experimental y el aprendizaje por descubrimiento.

Debido a la dificultad para la implementación de todo este tipo de pruebas sobre redes MPLS reales, OpenSimMPLS es una solución para enfrentar al estudiante de redes con los diferentes casos de funcionamiento que plantearía un posible dominio MPLS. Así, el docente puede investigar, planificar y dimensionar los recursos de una red MPLS sin necesidad de correr riesgos al modificar las configuraciones de dispositivos reales. Por otra parte, y como objetivo colateral, los resultados analíticos obtenidos pueden ser de utilidad al alumno para desarrollar nuevas metodologías de diseño de cara a su futuro trabajo como planificador de redes de última generación. Para los estudiantes y futuros profesionales es, por tanto, muy útil la adquisición de una disciplina que primero conlleva la simulación de la topología de red, la obtención de resultados y los sucesivos refinados hacia la arquitectura final antes de su implantación, la cual implica una racionalización de la infraestructura MPLS de telecomunicaciones necesaria, así como una estimación del coste o inversión que ésta puede implicar.

Por tanto, el empleo del simulador puede ser un aspecto crucial para la comprensión clara de los conceptos de comunicaciones en MPLS. Refuerza los conceptos teóricos de la tecnología y ofrece al estudiante una motivación a la hora de comprender la interacción entre los diferentes componentes que forman un escenario. La simulación de un dominio MPLS es un mecanismo para orientar la comprensión del funcionamiento del protocolo, así como de las diferentes cuestiones de diseño y sus repercusiones en el rendimiento.

En resumen, como apoyo a la docencia sobre redes MPLS, el empleo del simulador presenta diversas ventajas:

- Con OpenSimMPLS se puede modificar la configuración de los distintos componentes, para

luego analizar las consecuencias de dichos cambios. Sobre una red MPLS real no siempre estará permitido realizar esos cambios de configuración.

- La simulación permitirá la obtención de estadísticas detalladas, las cuales se pueden utilizar para comprobar posibles compromisos de QoS sobre tráfico particulares.
- El empleo del simulador supondrá siempre una solución docente y de validación del aprendizaje más económica que la implantación de un dominio MPLS real.

En nuestro caso, OpenSimMPLS se utiliza para innovar en la docencia de asignaturas universitarias de redes de ordenadores, tales como *Comunicaciones en Banda Ancha* o también *Planificación, Especificación, Diseño y Evaluación de Redes*, ambas impartidas en el segundo ciclo de Ingeniería Informática, en la Universidad de Extremadura. También se considera un recurso idóneo para impartir cursos de doctorado o líneas de investigación sobre redes avanzadas.

## 2.1 Validación de resultados

OpenSimMPLS ofrece también la posibilidad de ser usado como herramienta de validación de resultados en proyectos de investigación. Por tanto, el otro gran colectivo de usuarios estaría formado por los investigadores que deseen disponer de una herramienta de simulación potente para avanzar en la generación de nuevas técnicas MPLS, basadas en

propuestas diferentes a las actuales, con las consiguientes medidas de prestaciones de estos nuevos conceptos en una red. El simulador se ha desarrollado como software libre y así se facilita su empleo como plataforma para la ingeniería de protocolos: el investigador de proyectos relacionados con MPLS puede añadir nuevas clases o módulos al proyecto en función de sus necesidades. De esta forma puede obtener una extensión del simulador original para dar cabida al análisis de los resultados específicos de sus investigaciones.

Para nuestra investigación, el simulador ha sido ampliado de forma que ofrezca compatibilidad con escenarios capaces de ofrecer Garantía de Servicio (GoS). Podemos entender como GoS a la posibilidad de ofrecer a un cierto flujo de tráfico la seguridad de que en condiciones normales será tratado preferentemente con respecto al resto de flujos y que, en caso de la existencia de problemas (pérdidas de paquetes y caídas de enlaces), la arquitectura pondrá los medios necesarios para que en cualquier caso el flujo que requiera GoS sea favorecido, tanto más cuanto más GoS haya sido especificada para ese flujo y siempre en función de las posibilidades de los nodos que atraviese. En la figura 1 se presenta, a modo de ejemplo, una topología en la que cuatro nodos emisores generan flujos de diferentes niveles de GoS. Todos los LSP creados atraviesan un LSR (Label Switch Router) central, con el signo de admiración, el cual sufrirá, por tanto, una elevada congestión. Como consecuencia de esto, se producirá el descarte de paquetes en dicho nodo.

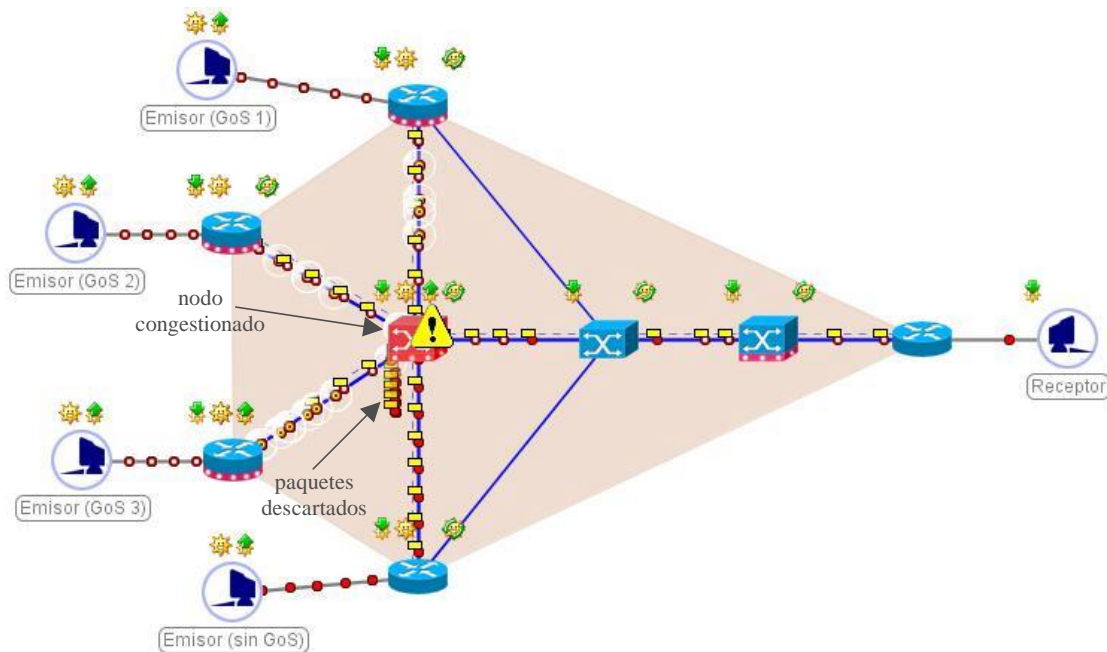


Figura 1: Escenario en el que coexisten flujos de diferentes niveles de GoS

En la gráfica de la figura 2 se muestra que los flujos más prioritarios (con mayor nivel de GoS) tendrán un menor número de paquetes descartados. Es decir, estos flujos se priorizan con respecto a los de menor GoS. De esta forma, el simulador soporta el trabajo con varios niveles de GoS. Por tanto, cada paquete puede ser marcado con estos atributos desde el nodo inicial hasta el nodo final. Cada uno de estos niveles debe entenderse como un grado de probabilidad de que un paquete se pueda localizar en cualquiera de los nodos activos por los que ha pasado (LERA, LSRA). Así se evita la necesidad de retransmisiones extremo a extremo. Por otro lado, se ha considerado también la posibilidad de recuperación de LSPs rotos de forma local adoptando una solución en un entorno mucho más local. Para validar los resultados de nuestro trabajo, hemos utilizado el simulador para comparar de manera estadística el rendimiento que ofrece el simulador para una red MPLS con los resultados de un dominio con soporte de flujos privilegiados GoS.

En resumen, además de los docentes y estudiantes universitarios, existen otros beneficiarios muy directos, entre los que se puede citar a los investigadores que centran su trabajo en estos temas, tanto en universidades como en centros de investigación o escuelas técnicas.

### 3 Simulador OpenSimMPLS

OpenSimMPLS está constituido por una aplicación *jar* auto-contenida. Su instalación, por tanto, no requiere de ningún paso significativo, y simplemente se debe invocar su ejecución a través de la Máquina Virtual Java de SUN que debe haber sido previamente instalada.

La principal característica del entorno de trabajo del simulador se basa en su simplicidad. Se divide en tres partes: *área de trabajo*, *menú principal* y *ventanas de escenarios*. El área de trabajo es el entorno principal, dentro del cual se desarrollará la simulación de los diferentes escenarios MPLS. El menú principal está situado en la parte superior izquierda, de forma similar al de cualquier otra aplicación, englobando las opciones relacionadas con la gestión de ficheros (crear, almacenar y recuperar escenarios de disco), visualización de ventanas y ayuda.



Figura 2: Paquetes descartados en el LSR Activo

Por último, las ventanas de escenarios permitirán el diseño y análisis de escenarios MPLS particulares. Su estructura se divide en varias pestañas (Fig. 3), las cuales se irán describiendo en los siguientes apartados.

#### 3.1 Área de Diseño de Topologías

La primera pestaña engloba el *Área de Diseño*, en la que se establecerán los parámetros relacionados con la topología y configuración del dominio a simular. La barra de herramientas incorpora diversos iconos que representan los elementos que se pueden insertar en un dominio de OpenSimMPLS (Fig. 3).

El primer icono hace referencia al *Emisor*, que es el tipo de nodo encargado de generar tráfico de red en el simulador. El segundo icono referencia al *Receptor*, el cual actuará como sumidero del flujo generado por un emisor. El tercero representa los *LER* (Label Edge Router), encargados de etiquetar paquetes IP o MPLS, clasificarlos, establecer un camino hacia el destino a través del dominio MPLS y, finalmente, permitir la entrada del paquete etiquetado al dominio MPLS. El cuarto icono es el *LERA* (Label Edge Router Activo), que realiza la misma tarea que el *LER*, pero además se encarga de analizar la cabecera IP para saber si los paquetes tienen requerimientos de garantía de servicio (GoS) y si es así, codificar esos requisitos en la cabecera MPLS [7].

Un flujo IP marcado con GoS sólo puede conservar esos atributos de GoS dentro del dominio MPLS si accede a él a través de un nodo LERA. El siguiente icono representa al *LSR*, encargado de conmutar tráfico MPLS en el interior del dominio. Es un componente muy rápido, pues sólo observa la etiqueta puesta sobre el paquete por el LER/LERA de entrada al dominio MPLS. Un nodo LSR nunca puede hacer de nodo de entrada al dominio MPLS pues no tiene capacidad para ello. El sexto icono hace referencia a los *LSRA* (Label Switch Router Activo), que serán los encargados de conmutar tráfico MPLS en el interior del dominio. Además, el LSRA es el componente con capacidad de recuperación local de paquetes y de reestructuración de caminos (LSP) en un entorno local. También tendrá capacidad de almacenar paquetes de forma temporal, para así satisfacer las posibles solicitudes de retransmisión local de otro LSRA del dominio. El último icono representa al *Enlace*, que es el elemento que une dos nodos cualesquiera de la red. Todo escenario de simulación debe tener sus componentes conectados mediante enlaces, por los que fluye el tráfico.

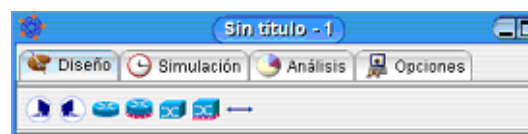


Figura 3: Detalle de la ventana de escenarios de OpenSimMPLS

### 3.2 Área de Simulación de Escenarios

Debe pasarse al *Área de Simulación* cuando se ha finalizado la creación de la topología del escenario. En este entorno puede analizarse visualmente el comportamiento de dicha topología una vez comiencen a generarse tráficos, saturaciones, caídas de enlaces, etc (Fig. 4). La topología de simulación que podremos ver será la correspondiente al escenario que se haya diseñado en el área de diseño. El área de simulación presenta una estructura similar a la de diseño. La diferencia estriba en que en el lugar donde aparecerían los elementos a insertar en el escenario, ahora aparecen unos iconos para controlar el funcionamiento de la simulación.

Si se ha terminado de diseñar y configurar la topología en el área de diseño, se puede poner en funcionamiento la simulación. Esto se hace mediante un clic en el primer icono, que simula un engranaje. Cuando la simulación está en funcionamiento, una barra de progreso indica en todo momento el porcentaje de la simulación en curso. También existe un contador que muestra el número de nanosegundos consumidos en la simulación. Por otro lado, también es posible ralentizar la simulación, lo cual es muy útil para observar con detenimiento los sucesos que van ocurriendo, sin necesidad de detener y reanudar periódicamente la simulación. De esta tarea se encarga el deslizador que se encuentra en la barra de herramientas (Fig. 5).

Toda la simulación visual que se puede observar en el área de simulación en tiempo real no es sino la representación gráfica de los valores internos generados por los elementos que componen el escenario. En la mayoría de las ocasiones, la simulación visual, junto con las estadísticas generadas por los nodos que estén configurados para ello, es suficiente para comprender los diferentes eventos ocurridos en la simulación. Sin embargo, hay ocasiones en que es necesario tener la posibilidad de acceder a una interpretación numérica de alguna situación compleja.

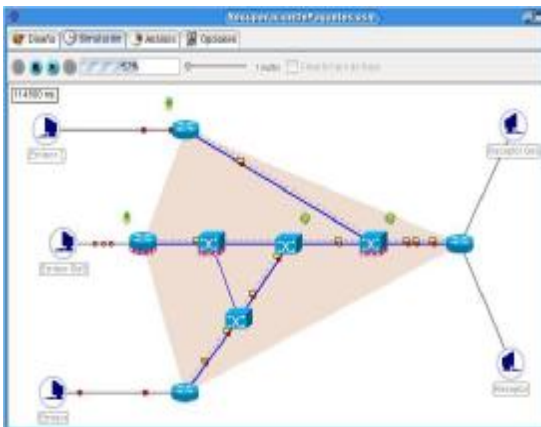


Figura 4: Área de Simulación, en ejecución de un escenario de ejemplo



Figura 5: Detalle de la barra de herramientas del Área de Simulación

Para ello, es posible generar un fichero de traza donde se almacenen, en formato de texto plano, todos los eventos que han tenido lugar durante la simulación: qué componente se ha visto afectado, en qué instante, consecuencias, etc. De esta forma se proporciona un método funcional para la revisión a posteriori de la simulación. Si se desea generar este fichero de traza, se debe hacer clic con el botón principal del ratón sobre el recuadro de selección llamado "Crear fichero de traza", de la barra de herramientas.

Durante la simulación, los diferentes elementos del escenario podrán ir modificando su aspecto visual a medida que se avanza en el tiempo. Por ejemplo, los nodos LER y LSR modificarán su color en función del nivel de congestión que sufran.

El cambio de una apariencia a otra se realiza de forma automática, a medida que los paquetes se van acumulando en el búfer del nodo. Los paquetes permitirán conocer qué tipos de flujo (clasificados según su prioridad) se dan en el escenario. También informan sobre la cantidad y tipos de tráficos que se mueven por la red, cuándo y cómo se produce la señalización, caminos por los que circulan, velocidad a la que se mueven, etc.

Por otro lado, además de circular por la red que se esté simulando, los paquetes podrán ser descartados en nodos que sufran un elevado nivel de congestión. En ese caso los paquetes aparecerán, visualmente, cayendo de dicho nodo (Fig 6).

Los diferentes aspectos comentados sobre la representación de paquetes pertenecientes a diferentes tipos de tráfico, así como del flujo de los mismos, se puede consultar durante la simulación, gracias a la leyenda que se muestra (opcionalmente) en la esquina inferior derecha del entorno (Fig 7).



Figura 6: Descarte de paquetes pertenecientes a diferentes tipos de tráfico



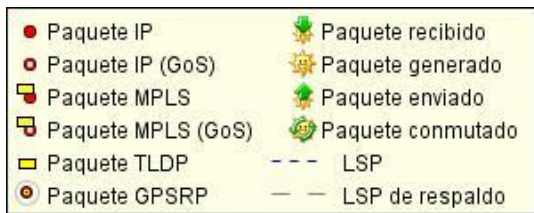


Figura 7: Leyenda informativa sobre tipos de paquetes, flujos y LSP

Hasta ahora, se han analizado algunas de las señales visuales que se deben interpretar durante la simulación para comprender los sucesos que tengan lugar. Sin embargo, la simulación es un entorno interactivo, podrán llevarse a cabo numerosas acciones durante el funcionamiento de la misma. Por ejemplo, además de mostrarse el nivel de congestión de un nodo particular, puede provocarse una congestión haciendo clic con el botón principal del ratón.

A partir de ese momento, el nodo experimentará una elevada saturación de paquetes (Fig 8). Lo habitual es que en un corto periodo de tiempo el nodo comenzará a descartar paquetes si sigue recibiendo tráfico entrante. Esta función es muy práctica para provocar pérdidas y recuperaciones de paquetes sin tener que esperar la congestión del nodo.

En condiciones reales, un enlace está sujeto a la posibilidad de averías. Obras, descargas eléctricas, fallos humanos, etc., pueden hacer que un enlace falle y el tráfico se pierda. Sin embargo, el exceso de tráfico no provoca el fallo del enlace. OpenSimMPLS permite simular este hecho, admite que un enlace pueda caer en un momento dado, pero al igual que en la realidad, tampoco es algo que ocurra como evolución de la simulación sino que se ha de provocar manualmente. Podremos simular la caída de un enlace durante una simulación mediante un clic de ratón sobre el mismo (Fig. 9). El enlace cambiará su apariencia, mostrándose como una línea roja discontinua y provocando que todos los paquetes circulantes sean descartados. De esta forma podrán simularse situaciones en que el dominio deba recuperarse de fallos de enlace.



Figura 8: Congestión/descongestión artificial de nodos



Figura 9: Rotura manual de enlace

### 3.3 Área de Análisis de resultados

Puede pasarse a trabajar al Área de Análisis cuando se han configurado en la topología algunos elementos para que generen estadísticas y se desee observar las gráficas que generan (o que ya han sido generadas, si la simulación ha concluido). Este área se divide en dos partes: una barra de herramientas para el análisis y una zona mayor donde se mostrarán las estadísticas solicitadas para los componentes particulares (Fig 10). Si la simulación está en curso, las gráficas tendrán un comportamiento dinámico, variando según va evolucionando la simulación. Si ya ha finalizado, las gráficas mostrarán los resultados definitivos.

Por otro lado, las gráficas generadas por OpenSimMPLS para cada elemento no son imágenes estáticas, sino que actúan como objetos interactivos. Se puede obtener un menú emergente con opciones sobre cada una de las gráficas, simplemente haciendo clic sobre ellas con el botón secundario del ratón (Fig. 10). De esta forma se tendrá acceso a diversas funciones, como almacenar la imagen en disco, hacer zoom o imprimir una gráfica de interés, entre otras.

### 4 Detalles de implementación

Una de las ventajas de OpenSimMPLS es su portabilidad, ya que funciona de forma independiente a la arquitectura o sistema operativo del ordenador en el que se ejecute. Para ello se ha empleado el lenguaje Java. Éste también ha permitido la implementación del simulador como una aplicación multiproceso, mediante la programación de hilos.



Figura 10: Área de Análisis, con detalle de menú emergente sobre una gráfica

Asimismo, Java es un lenguaje orientado a objetos; la clase principal del sistema, denominada *openSimMPLS*, inicia la ejecución del simulador. El método *main()*, que se encuentra en esta clase, crea un objeto de tipo *TDispensadorDeImágenes* que será el encargado de cargar todas las imágenes necesarias en la aplicación y que posteriormente será pasado como parámetro en el constructor de cualquier elemento referente a la interfaz. Posteriormente se crea un objeto de tipo *JSimulador* que es la interfaz principal de la aplicación. A partir de este momento la ejecución del simulador dejará de ser secuencial y en su lugar atenderá a los eventos generados por el usuario en la interfaz: órdenes de ratón, selección de opciones de menú, etc.

Durante la simulación, un componente *reloj* enviará avisos a los elementos de la topología (enlaces y nodos) en forma de eventos de temporización o tics (Fig. 11). El reloj es un elemento que se configura con dos valores: por un lado, la duración total de la simulación completa (número de tics); por otro lado, la duración de cada tic. El reloj, que se ejecuta en un hilo propio, avanzará desde cero hasta el número máximo de tics definido para la simulación completa.

Cada tic será enviado a todos los elementos de la topología, de manera que al llegar al máximo de duración de la simulación, el hilo se detendrá y se finalizará la simulación.

Cuando los diferentes elementos de la topología reciben un tic, también reciben la duración en nanosegundos del mismo. En este momento, cada componente activa su propio hilo de ejecución, generándose por tanto concurrencia. Cada hilo realizará su función correspondiente, en función del tipo de dispositivo; por ejemplo, conmutar, transportar paquetes, recibir tráfico, etc. El hilo de ejecución de cada elemento se detendrá cuando la duración del tic recibido se agote.

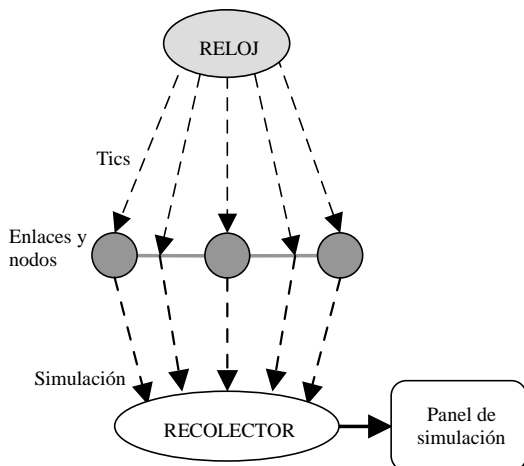


Figura 11: Esquema general de funcionamiento de la simulación de un escenario

Cuando todos los elementos han agotado su tic, el reloj lo detecta y genera el siguiente tic y la operación se vuelve a repetir. La clase que implementa el reloj del sistema se denomina *Treloj* e incorpora una lista interna de todos los elementos a los que debe enviar los eventos de temporización.

Durante el tiempo que el hilo específico de cada elemento está en funcionamiento, se dan multitud de sucesos que deben ser recogidos. Una vez representados, darán lugar a la visualización en pantalla de cada suceso. Este proceso de adquisición lo realiza el *recolector* global del escenario. Todos los elementos de la topología le notificarán la labor que están llevando a cabo durante su tiempo de funcionamiento. Mediante la reiteración de este proceso se consigue una simulación muy fluida, aunque la generación de tics sea un proceso discontinuo. Por tanto, los diferentes elementos de una topología (*TNodoEmisor*, *TNodoReceptor*, *TNodoLER*, *TNodoLSR*, *TNodoLERA*, *TNodoLSRA*, *TenlaceInterno*, *TenlaceExterno*), encapsularán métodos para llevar a cabo estas tareas.

La clase que implementa el recolector de eventos de simulación se denomina *TrecolectorSimulacion*. Implementa el método *capturarEventoSimulacion()*, que permite que los elementos de la topología puedan enviarle los eventos de simulación que van generando durante los tiempos en que sus hilos están en ejecución. Sin embargo, el recolector no muestra los sucesos, sólo los recoge. Para la visualización se han de utilizar los servicios de un componente gráfico que vaya representando en pantalla los diferentes eventos, el cual recibe el nombre de *Panel de Simulación*. De esta forma se consigue aislar las tareas de generación y recopilación de sucesos de las de representación visual de los mismos.

La pantalla de simulación está implementada en la clase *JPanelSimulación*, que realiza todas las operaciones de refresco de pantalla, simulación visual, etc.; es decir, interpreta todos los eventos que le llegan al recolector y los muestra en pantalla de una forma accesible para el usuario. Encapsula *ponerEvento()*, que es el método usado por el recolector para enviarle los eventos que ha adquirido.

#### 4.1 Topología del escenario

La topología es un objeto que almacena todos los elementos del escenario y que se encarga de conectar o interconectar enlaces con nodos y de establecer las asociaciones entre los elementos y el reloj o entre los elementos y el recolector de eventos. Se encuentra implementada en la clase *Ttopologia*. Para llevar a cabo estas tareas, cada elemento debe utilizar un identificador que le será asignado por un generador de identificadores que posee la topología.

El aspecto primordial de la topología son los importantes métodos que implementa. Cabe destacar

tres de ellos, los cuales implementan los algoritmos de encaminamiento:

- *obtenerIPSalto()*, recibe como parámetros la dirección IP origen de un paquete y la dirección IP destino a donde se desea enviar y devuelve la IP del siguiente nodo de la ruta, por el que hay que seguir.
- *obtenerIPSaltoRABAN()*, este método realiza la misma labor que el anterior, pero calcula la dirección IP según el algoritmo RABAN (Routing Algorithm for Balanced Active Networks)[7].
- *obtenerIPSaltoRABAN()*, también realiza la misma labor, pero en este caso admite otro parámetro que es la dirección IP de un nodo adyacente al origen por el que no se desea pasar. A efectos prácticos, este método devuelve el segundo mejor salto posible para llegar del origen al destino.

## 4.2 El escenario de simulación

El *escenario* es una clase que contiene todo lo referente a un entorno completo de simulación. Reloj, recolector, topología, generadores de identificadores y de IP, nodos, enlaces, etc., se encuentran en un objeto de tipo *Tescenario*, de forma que es un solo objeto el que almacena todos los elementos de cada escenario. Algunos de sus atributos más importantes son:

- Una topología, de tipo *Topologia*, donde se almacenan todos los elementos y donde se realizan todas las conexiones entre ellos, como ya se ha comentado en el apartado 4.1.
- Un objeto de simulación, de tipo *TSimulacion*, que es donde se encuentra el recolector de eventos de simulación.

El método más importante del escenario es *generarSimulacion()*, que pone en funcionamiento el reloj de la topología y de este modo la simulación comienza a funcionar. En realidad, *Tescenario* incluye todos los ingredientes necesarios para funcionar aunque no exista interfaz de usuario.

## 5 Conclusiones

El presente trabajo propone el empleo de OpenSimMPLS como herramienta de innovación docente en asignaturas de redes y comunicaciones, justificado por el creciente interés que está despertando la tecnología MPLS. Asimismo, se ha demostrado que el simulador es una herramienta apta para la validación de resultados en proyectos de investigación relacionados con MPLS. Por tanto, con filosofía multiplataforma y licencia de software libre, se pretende potenciar su evolución, al recibir e incorporar todas las sugerencias que de su uso se deriven de una forma cooperativa.

## Referencias

- [1] Jose L. Marzo, Eusebi Calle, Caterina Scoglio, and Tricha Anjali, "QoS Online Routing and MPLS Multilevel Protection: A Survey," IEEE Communications Magazine, October 2003.
- [2] M. Kodialam and T. V. Lakshman, "Restorable Dynamic QoS Routing," IEEE Communications Magazine, June 2002.
- [3] E. Rosen et al., "Multiprotocol Label Switching Architecture," RFC 3031, January 2001.
- [4] Janus Gozdecki, Andrzej Jajszczyk, and Rafal Stankiewicz, "Quality of Service Terminology in IP Networks," IEEE Communications Magazine, March 2003.
- [5] G. Ahn, W. Chun, Design and Implementation of MPLS Network Simulator. Chungnam National University of Korea, February 2001.
- [6] Análisis de la Integración entre Tráfico IP y Redes MPLS. Simulador MPLS. Miguel Ángel Martín Tardío, Miguel Gaspar Rodríguez, José Luis González-Sánchez. III Jornadas de Ingeniería Telemática. JITEL'01. Barcelona, 19-21 septiembre 2001.
- [7] An Architecture to Provide Guarantee of Service (GoS) to MPLS. A. M. Domínguez-Dorado, F. J. Rodríguez-Pérez, J. L. González-Sánchez, J. L. Marzo, A. Gazo. IV Workshop in G/MPLS Networks. Girona. April 21-22, 2005.
- [8] G. Ahn, W. Chun, Simulator for MPLS Path Restoration and Performance Evaluation, Chungnam National University, Korea, April 2001.
- [9] MPLS Simulator:  
<http://www-entel.upc.es/xavierh/mpls/>
- [10] OpenSimMPLS:  
[http://patanegra.unex.es/opensimmpls/web/es/in diceES.html](http://patanegra.unex.es/opensimmpls/web/es/indiceES.html)

---

Este trabajo está financiado, en parte, por la Consejería de Educación, Ciencia y Tecnología de la Junta de Extremadura, Proyecto AGILA, con código No. 2PR03A090 y por la CICYT, Proyecto MAIN-RA, con código No. TIC2003-05567.