

# Simulador MPLS para la Innovación Pedagógica en el Área de Ingeniería Telemática

M. Domínguez Dorado, F. J. Rodríguez Pérez, J. L. González Sánchez

**Abstract** — MPLS (Multiprotocol Label Switching) provides interesting mechanisms to integrate network technologies like ATM (Asynchronous Transfer Mode) and IP (Internet Protocol) with QoS (Quality of Service). It is a next generation technology with a great interest in teaching of networks and communications in Telematics Engineering degrees. In this paper we show a simulator as an innovative and educational tool to introduce the student to MPLS. It motivates the student to configure, interact and analyse the operation of an MPLS domain in an easy and efficient way. On the other hand, due to its free software license, it can be used as a framework for protocol engineering.

**Index terms** — simulator, MPLS, GoS, packet retransmission, LSP recovery, teaching tool.

## I. INTRODUCCIÓN

MPLS (*Multiprotocol Label Switching*) es una tecnología orientada a conexión que surge para paliar los problemas que plantean las redes actuales en cuanto a velocidad, escalabilidad e ingeniería de tráfico [1]. Al mismo tiempo ofrece QoS (*Quality of Service*) extremo a extremo [2], mediante la diferenciación de flujos y la 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, se sitúa la cabecera MPLS 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 encapsulados desde niveles superiores.

Es una tecnología multiprotocolo que admite cualquier protocolo de red, pero al mismo tiempo permite cualquier tecnología en 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; sin embargo, 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 son válidas para disponer de IP sobre MPLS y MPLS sobre ATM.

En el presente artículo se presenta un simulador de redes MPLS al que hemos llamado OpenSimMPLS. Es una herramienta funcional y visual (Figura 1) que puede utilizarse en la docencia de asignaturas de redes y/o comunicaciones. Contempla los aspectos fundamentales de funcionamiento y configuración de un dominio MPLS [4]; al mismo tiempo ha sido ampliado para incluir compatibilidad con dominios que soporten GoS (Garantía de Servicio) [5]. Un dominio MPLS con capacidad GoS puede entenderse como un entorno capaz de llevar a cabo recuperaciones locales de paquetes MPLS descartados, junto con la posibilidad de recomponer localmente LSPs (*Label Switched Paths*) [6], [7]. Esta característica se puede ofrecer a flujos privilegiados particulares sensibles de necesitar un servicio de fiabilidad y velocidad, ya que el sistema GoS permite, además, la priorización de estos tráficos con respecto a los no caracterizados como GoS.

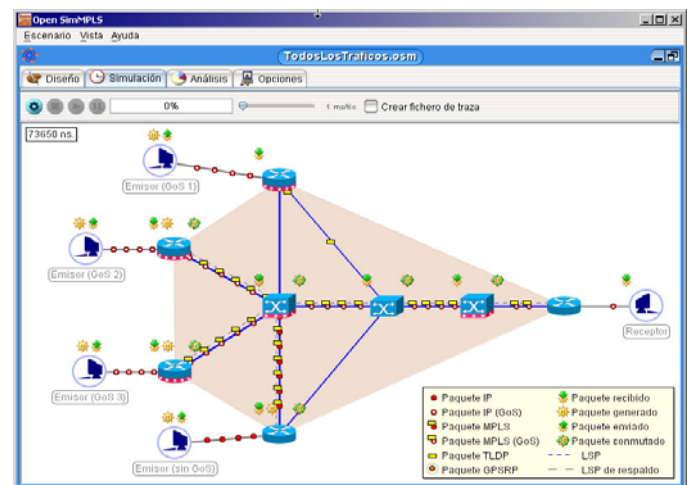


Figura 1. Vista general de una Ventana de Escenario donde coexisten flujos de diferentes niveles de GoS.

En la siguiente sección se hace una comparativa con otros simuladores de ámbito docente. En el apartado tres se muestra una breve descripción del entorno visual del simulador, así como de algunos aspectos funcionales. En el cuarto apartado se comentan diversos detalles relativos a su implementación.

M. Domínguez Dorado pertenece al DISIT de la Universidad de Extremadura. Avda. de la Universidad s/n. CP: 10071. Tlf: +34 607 417 860. Fax: +34 927 257 202. e-mail: mdomdor@unex.es.

El autor F. J. Rodríguez Pérez pertenece al DISIT de la Universidad de Extremadura. Avda. de la Universidad s/n. CP: 10071. Tlf: +34 927 257 195. Fax: +34 927 257 202. e-mail: fjrodri@unex.es.

El autor J. L. González Sánchez pertenece al DISIT de la Universidad de Extremadura. Avda. de la Universidad s/n. CP: 10071. Tlf: +34 927 257 195. Fax: +34 927 257 202. e-mail: jlgs@unex.es.

DOI (Digital Object Identifier) Pendiente

En el quinto se resaltan las aplicaciones de OpenSimMPLS en entornos docentes o para la prueba inicial de algoritmos en trabajos de investigación. En el apartado seis se muestran resultados recientes obtenidos tras el empleo del simulador en docencia. Finalmente, el artículo concluye resumiendo las contribuciones del simulador y líneas futuras de trabajo.

## II. TRABAJOS RELACIONADOS

OpenSimMPLS no es el primer esfuerzo que se realiza para conseguir un simulador docente relacionado con MPLS. Otros investigadores ya han hecho propuestas en este sentido. Por ejemplo, en [8] se puede encontrar un simulador MPLS que permite 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. Entre sus características principales se encuentran la de ser un simulador orientado a la docencia que permite un análisis estadístico elemental del tráfico de la red; cuenta con un editor visual de escenarios, es multiplataforma y, aunque de facto es gratuito y de fácil instalación (es un *applet* ejecutable desde la página de los autores), no está disponible su código fuente y tampoco se puede descargar para su ejecución local. Por otro lado, no permite interactuar con la simulación en el momento de su ejecución ni cambiar la interfaz a otros idiomas. Además, debido a su orientación docente, no está concebido para trabajar en conjunción con redes MPLS reales con componentes de fabricantes actuales.

MNS (*MPLS Network Simulator*) [9] es una extensión de NS2 (*Network Simulator 2*) para permitir la simulación de redes MPLS. Sus características son, por tanto, similares a las de NS2, a saber: está orientado a la docencia, permite un completo análisis estadístico de todos los eventos ocurridos durante la simulación, es multiplataforma, se distribuye bajo licencia de software libre y es gratuito. En su contra, se puede decir que no permite al usuario interactuar en tiempo de ejecución de la simulación para alterar el comportamiento de la misma; es decir, la simulación transcurrirá como se especificó previamente en el fichero de configuración del escenario. En docencia este comportamiento tan estático impide al alumno explotar el mecanismo de aprendizaje por descubrimiento. Ni la instalación de MNS+NS2 es sencilla, (requiere compilar y tener instaladas ciertas librerías), ni resulta trivial la generación de escenarios de simulación sin conocimientos previos de TCL (*Tool Command Language*) para la definición textual de escenarios, así como de la gran cantidad de componentes ofrecidos por el simulador; esto último confiere a MNS+NS2 la virtud de ser una completa herramienta para la validación de resultados de investigación, pero en lo que a docencia se refiere, presenta una curva de aprendizaje que dificulta el desarrollo de sesiones prácticas en un tiempo prudencial, teniendo en cuenta los ajustados calendarios docentes así como de disponibilidad de laboratorios en los centros universitarios. Al igual que el anterior simulador, tampoco es apto para trabajar en entornos de redes reales y, aunque es ampliamente utilizado en docencia, está más enfocado a la investigación sobre redes.

En el caso de OpenSimMPLS [10], para facilitar el uso futuro del simulador en diferentes centros, el software es multiplataforma y está liberado bajo licencia GPL (*General Public License*) v2.0 de la *Free Software Foundation*. Es un simulador que incorpora las ventajas de [8] y [9], pero además permite al alumno la interacción durante la simulación, está traducido a otros idiomas, permite la generación visual de escenarios, la simulación de la tecnología GoS y no requiere instalación. En la Tabla 1 se muestra un resumen de las características de los tres simuladores.

TABLA 1  
COMPARATIVA ENTRE SIMULADORES

	MNS /NS2	Simulador de Multi-Protocol Label Switching	OpenSimMPLS
Docente	•	•	•
Simulación interactiva			•
Estadísticas	•	•	•
Multiplataforma	•	•	•
Multilingüe			•
Diseño visual		•	•
Software libre	•		•
Gratuito	•	•	•
Simula GoS			•
Aplicable a entornos reales			
Facilidad instalación/ejecución		•	•

Decir que para trabajos más especializados (no docentes) existen herramientas más completas, como Totem [11] u OPNET [12] más orientados para su uso en entornos reales de redes en producción.

## III. SIMULADOR OPENSIMMPLS

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 la zona principal, dentro de la cual se desarrolla la simulación de los diferentes escenarios MPLS. El Menú Principal está situado en la parte superior izquierda, englobando las opciones relacionadas con la gestión de ficheros (crear, almacenar y recuperar escenarios a/de disco), visualización de ventanas y ayuda.

Por último, las Ventanas de Escenarios permiten el diseño y análisis de escenarios MPLS particulares. Su estructura se divide en varias pestañas, las cuales se irán describiendo en los siguientes apartados.

### A. Área de Diseño de Topologías

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

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 es el 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 GoS y, si es así, codificar esos requisitos en la cabecera MPLS [5]. 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 son 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 tiene 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. En cualquier caso, OpenSimMPLS incorpora un control de errores que permite generar topologías libres de ellos.

### B. 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 se puede analizar visualmente el comportamiento de dicha topología una vez comiencen a generarse tráficos, congestiones, descartes de paquetes (Figura 2), etcétera. La topología de simulación que se observará 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 aparecían los elementos a insertar en el escenario, ahora aparecen unos iconos para controlar el funcionamiento de la simulación.



Figura 2. Descarte de paquetes diferentes a distintos tipos de tráfico.

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 muestra 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 permite 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.

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, con la representación visual de la simulación y con las gráficas estadísticas generadas de los elementos (enlaces, nodos...), es suficiente para comprender los diferentes acontecimientos 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. Para ello, es posible generar un fichero de traza donde se almacenan, en formato de texto, 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 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 pueden ser descartados en nodos que sufran un elevado nivel de congestión. En ese caso los paquetes aparecerán, visualmente, cayendo de dicho nodo (Figura 2).

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 (Figura 3). En la práctica, esta leyenda ha resultado ser de gran ayuda para los alumnos a la hora de analizar qué ocurría durante la simulación.

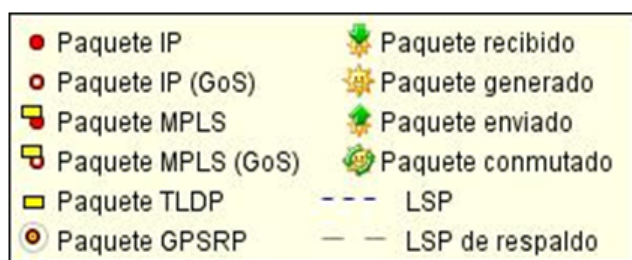


Figura 3. Ayuda visual para el alumno durante la simulación.

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 y pueden llevarse a cabo numerosas acciones durante el funcionamiento de la misma. Por ejemplo, puede provocarse la congestión de un nodo haciendo clic con el botón principal del ratón sobre él. A partir de ese momento, el nodo experimentará una elevada saturación de paquetes (Figura 4). Lo habitual es que en un corto periodo de tiempo el nodo comience 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.

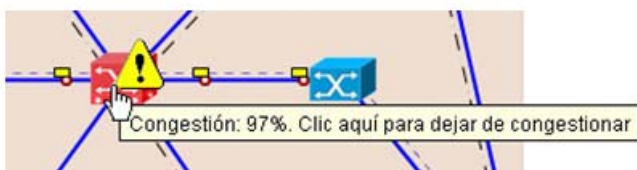


Figura 4. Congestión/descongestión artificial de nodos.

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. 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.

### C. Área de Análisis de resultados

Puede pasarse a trabajar al Área de Análisis (Figura 5) si se han configurado en la topología algunos elementos para que generen estadísticas; entonces se pueden observar las gráficas que generan (o que ya han sido generadas, si la simulación ha concluido).

Esta área se divide en dos partes: una barra de herramientas para el análisis y una zona mayor donde se mostrarán las gráficas estadísticas solicitadas para los componentes particulares.

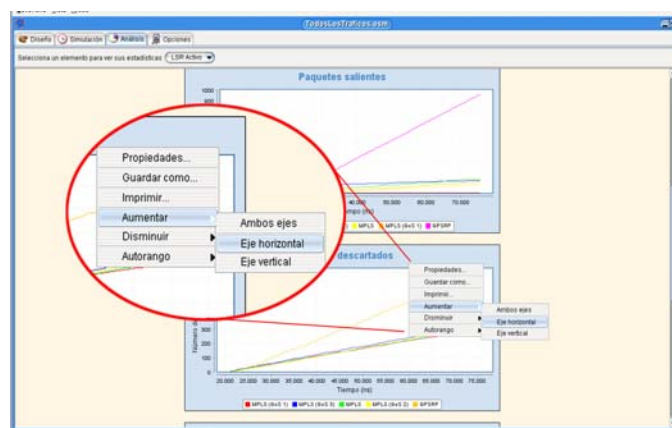


Figura 5. Área de análisis.

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. De esta forma se tendrá acceso a diversas funciones, como almacenar la imagen en disco, ampliar o imprimir una gráfica de interés, entre otras.

## IV. DETALLES DE IMPLEMENTACIÓN

OpenSimMPLS es 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.

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 multitarea (capaz de simular más de un escenario a la vez), lo que se consigue mediante la programación de múltiples hilos de ejecución.

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, mejorando el rendimiento. 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 deja de ser secuencial y en su lugar atiende 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 envía avisos a los elementos de la topología (enlaces y nodos) en forma de eventos de temporización o tics (Figura 6).

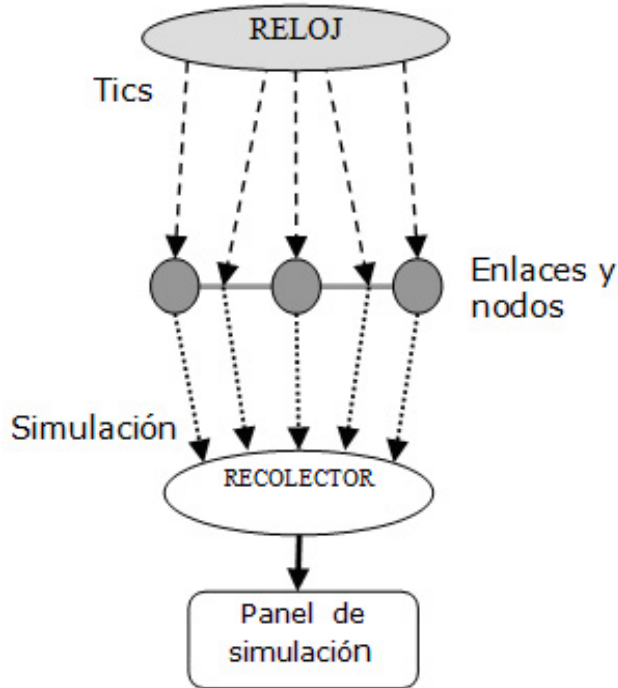


Figura 6. Esquema general de funcionamiento de la simulación de un escenario.

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, que están sincronizados, de manera que al llegar al tope de duración, el hilo se detendrá y se finalizará la simulación.

Cuando los diferentes elementos de la topología reciben un tic, también obtienen la duración en nanosegundos del mismo. 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.

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, ocurren 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 discreto. Por tanto, los diferentes elementos de una topología (*TNodeEmisor*, *TNodeReceptor*, *TNodeLER*, *TNodeLSR*, *TNodeLERA*, *TNodeLSRA*, *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 usan los servicios de un componente gráfico que va 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 *JPanelSimulacion*, 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.

#### A. Topología del escenario

La topología es un objeto que almacena todos los elementos del escenario y que se encarga de mantener el grafo de conexiones entre enlaces y nodos y de establecer las asociaciones entre los elementos y el reloj o entre los elementos y el recolector de eventos. Toda su funcionalidad se encuentra implementada en la clase *TTopologia*. Para llevar a cabo estas tareas, cada elemento debe utilizar un identificador único que le será asignado por un generador de identificadores únicos que posee la topología.

#### B. 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.

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, por lo que con pocos cambios se podría crear un intérprete de eventos que, por ejemplo, adecuara la visualización de los mismos para personas con dificultades visuales. Incluso se podría separar la generación de eventos y su visualización en programas diferentes que corrieran en máquinas separadas.

## V. APLICACIONES DOCENTES E INVESTIGADORAS DE OPENSIMMPLS

El objetivo fundamental de OpenSimMPLS es servir al profesorado universitario como herramienta para innovar en la docencia [13] y en el análisis del funcionamiento de redes MPLS a través de un sistema multiplataforma y multilingüe. Si el idioma del sistema operativo en que se ejecuta es español, OpenSimMPLS se mostrará en este idioma; en otro caso se mostrará en inglés. El simulador utiliza el sistema de internacionalización propio del lenguaje Java, por lo que está preparado para su traducción automática a otros idiomas de forma sencilla y rápida. El simulador dispone de una interfaz gráfica con 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. El simulador separa claramente las tres etapas de la simulación: primero para diseñar y configurar un dominio MPLS; después para mostrar de forma visual los diferentes eventos que van sucediéndose; y finalmente, para presentar estadísticas de la simulación.

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 cuando se introducen servicios particulares; por ejemplo, tráfico multimedia. También permite contrastar resultados gracias al sistema de reconfiguración de los elementos del dominio. De esta forma el alumno puede realizar propuestas para la mejora de supuestos de redes MPLS y detectar los posibles efectos perniciosos o beneficiosos sobre el tráfico.

OpenSimMPLS ofrece un sistema de aprendizaje por descubrimiento en el que se introduce al alumno a resolver problemas y situaciones, aprender procedimientos de la tecnología, llegando a entender las diferentes características de los eventos y decidiendo cómo controlarlos y qué acciones realizar en situaciones particulares, gracias a la interactividad que admite la herramienta durante la simulación. 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 del simulador (Figura 7), la cual debe descifrar para saber qué ocurre en el interior del dominio MPLS y determinar cuál es la norma o principios que rigen su funcionamiento. En resumen, el uso del simulador en el aula da lugar a un proceso de doble *feedback*: Por un lado la interacción con la simulación en ejecución permite al estudiante analizar el comportamiento del escenario MPLS, obteniendo conclusiones basadas en sus conocimientos teóricos previos y detectando así posibles problemas de la fase de diseño del escenario. Por otro lado, tras el análisis de los resultados estadísticos el alumno también puede obtener conclusiones que redundarán en nuevos cambios de configuración. Este proceso experimental y analítico de refinamientos sucesivos motiva al estudiante a desarrollar sus propias estrategias de pensamiento acerca del funcionamiento de la tecnología MPLS.

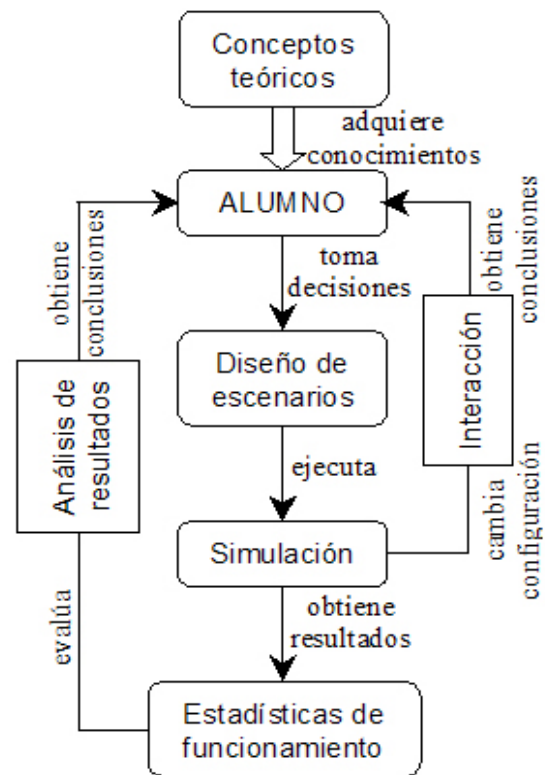


Figura 7. Proceso de aprendizaje seguido por el estudiante al emplear OpenSimMPLS.

A continuación y como muestra de las capacidades docentes del simulador, se indican las líneas a seguir para la realización de sesiones prácticas en el aula: en primer lugar el profesor debe exponer los conceptos teóricos fundamentales sobre la tecnología MPLS, así como los casos conflictivos que se pueden presentar en una red de este tipo (congestión de nodos, fallos de enlaces, pérdida de paquetes...). Tras las sesiones teóricas el alumno puede comenzar las sesiones prácticas de forma inmediata, ya que cuenta con un útil manual de usuario de OpenSimMPLS que hace innecesario dedicar una sesión práctica a explicar su funcionamiento. En estas sesiones prácticas se propondrá al alumno la simulación de:

- Escenarios MPLS de funcionamiento óptimo.
- Escenarios que presentan situaciones conflictivas.
- Escenarios donde el alumno debe interactuar congestionando nodos o provocando fallos de enlaces.

En el primero de los casos los escenarios harán que el alumno refuerce sus conocimientos teóricos. En el segundo punto el estudiante analizará situaciones en las que, sin llegar a producirse fallos en la red, ésta puede ser mejorada para optimizar su rendimiento, mediante reconfiguración de cada uno de los componentes de la red o de las características del tráfico generado. En el tercer caso el alumno tendrá la oportunidad de detectar y analizar las consecuencias de fallos eventuales en la red. Podrá comprobar cómo las variaciones introducidas afectan al rendimiento final del sistema [14].

OpenSimMPLS es interactivo, por lo que el alumno se convierte en parte actora durante la simulación (modificando

propiedades del escenario de forma dinámica), potenciando así el conocimiento experimental y el aprendizaje por descubrimiento [15], [16].

Nuestra herramienta es una solución para enfrentar al estudiante de materias de redes y comunicaciones con el funcionamiento de un dominio MPLS. Así, el alumno puede indagar, planificar y dimensionar los recursos de una red MPLS sin necesidad de correr riesgos al modificar la configuración de dispositivos de la red real del aula. Por otra parte, y como objetivo colateral, los resultados analíticos obtenidos serán de utilidad al alumno para desarrollar nuevas metodologías de diseño de cara a su futuro trabajo como planificador de redes de última generación. Permite la adquisición de una disciplina muy metódica: planificación de una red MPLS, obtención y análisis de resultados y refinamientos sucesivos hacia una arquitectura final óptima (Figura 7).

El uso de OpenSimMPLS refuerza los conceptos teóricos de la tecnología MPLS y ofrece al estudiante la motivación que necesita para comprender la interacción entre los diferentes componentes de un escenario.

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

- Simplicidad de implantación en el aula (no requiere base de datos, admite múltiples arquitecturas y sistemas operativos).
- 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 un laboratorio de la facultad.
- Con OpenSimMPLS se puede modificar la configuración de los componentes de un escenario, para luego analizar las consecuencias de dichos cambios y aprender de ellos. Sobre una red MPLS real en el aula no siempre estará permitido realizar cambios de configuración.
- La simulación permite la obtención de estadísticas detalladas con las cuales se pueden analizar comportamientos concretos del dominio MPLS.
- Su código abierto permite a los educadores impartir docencia en asignaturas de programación e ingeniería de protocolos, proponiendo para ello mejoras en la aplicación o la incorporación de nuevas funcionalidades.

## VI. EVALUACIÓN DE OPENSIMMPLS EN EL AULA

En nuestro caso, el simulador se utiliza para innovar en la docencia de asignaturas del área de Ingeniería Telemática en la Escuela Politécnica de Cáceres, tales como Comunicaciones en Banda Ancha o también Planificación, Especificación, Diseño y Evaluación de Redes, del segundo ciclo de Ingeniería Informática. También se emplea en el curso de doctorado “Integración de Redes Heterogéneas Multiprotocolo y Multimedia con Calidad de Servicio”, y en la línea de investigación “Comunicaciones Avanzadas de Banda Ancha, Multimedia y Multiprotocolo con QoS y Seguridad”, en la Universidad de Extremadura. Para el caso de Comunicaciones en Banda Ancha se ha realizado un estudio comparativo entre

las calificaciones obtenidas por los alumnos del curso 2005/2006 (en el que se impartió docencia sobre MPLS sin el empleo de OpenSimMPLS) y el curso 2006/2007, para el que se utilizó el simulador como material de apoyo. En la Figura 8 puede observarse que para el curso 2005/2006 la tasa de aprobados fue del 58,82%, con un 41,18% de suspensos y en el curso siguiente, con la incorporación de OpenSimMPLS, el porcentaje de aprobados subió al 66,67%, disminuyendo por tanto la tasa de suspensos al 33,33%.

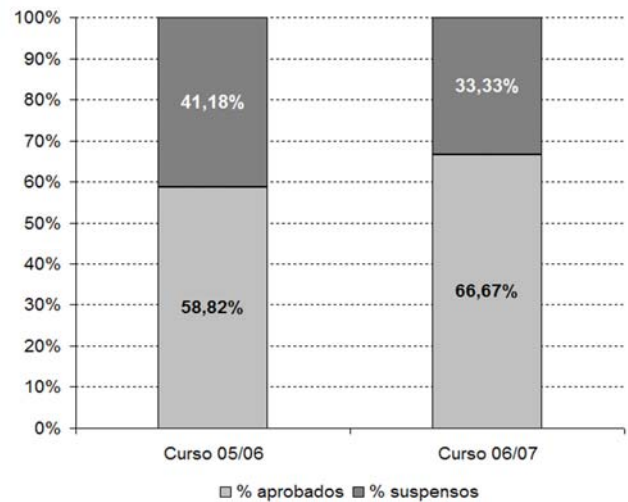


Figura 8. Comparativa de aprobados y suspensos en los cursos 2005/2006 y 2006/2007.

En la Figura 9 se observa además que la nota media del grupo en el curso 2005/2006 fue de un 4,67 sobre 10 y que en el curso 2006/2007 la media subió al 6,17 sobre 10. Al mismo tiempo, la desviación típica pasó a ser de 3,19 puntos en el 2005/2006 a 2,71 puntos en el 2006/2007, indicando esto una mejor homogeneización de las calificaciones de los estudiantes del grupo.

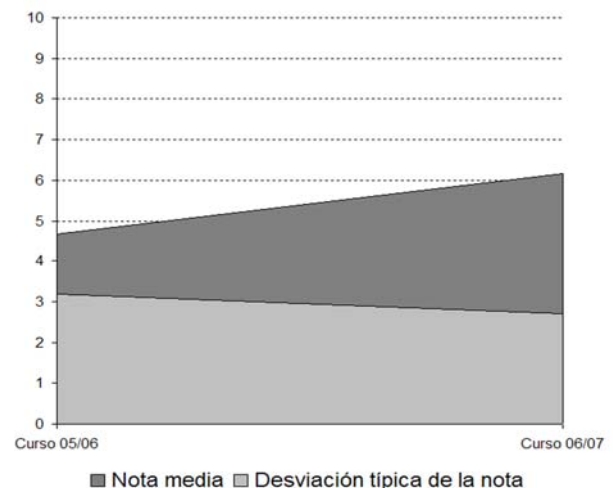


Figura 9. Variación de la nota media y la desviación típica de las notas en los cursos 2005/2006 y 2006/2007.

Como resumen, en la Figura 10 se cuantifica la mejora obtenida en el curso 2006/2007 al emplear OpenSimMPLS: Por un lado, el incremento de la calificación media ha sido del 32,07% y por otro, la disminución de la desviación estándar ha sido del 24,28%.

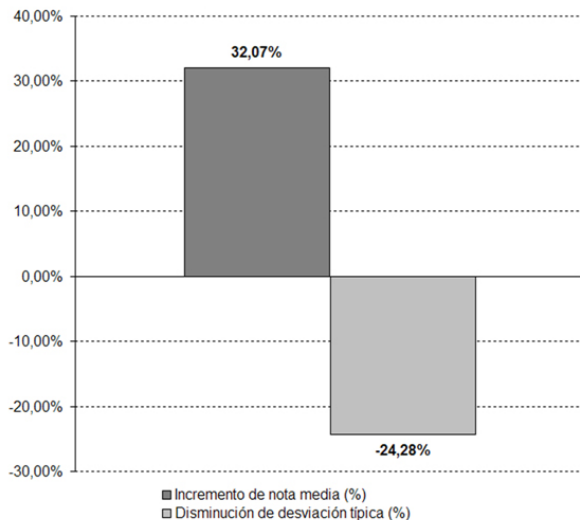


Figura 10. Variación de la nota media y de la desviación típica del curso 2006/2007 con respecto al curso 2005/2006.

## VII. CONCLUSIONES Y TRABAJOS FUTUROS

El empleo de OpenSimMPLS para realizar trabajos profesionales en entornos reales de red no es recomendable por ser un simulador orientado a docencia (no incorpora características reales de componentes de fabricantes actuales). Sin embargo, el presente trabajo propone el empleo de OpenSimMPLS como herramienta de innovación docente en asignaturas impartidas por el área de Ingeniería Telemática, justificado por el creciente interés que está despertando la tecnología MPLS.

Asimismo, el simulador es una herramienta de apoyo en proyectos de investigación implicados en MPLS (como muestra la implementación del sistema GoS sobre el simulador), así como en la docencia de asignaturas relacionadas con esta tecnología. Particularmente se ha comprobado la mejora de los resultados del grupo de Comunicaciones en Banda Ancha, observándose un aumento en la motivación e interés de los estudiantes, con una mejora de la nota media y obteniendo además unas calificaciones más homogéneas, como indica la disminución de la desviación típica.

La filosofía multiplataforma y la licencia de software libre de OpenSimMPLS posibilitan su propia evolución, ya que se puede incorporar el *feedback* recibido de otros usuarios a través de la página web del proyecto o desde el propio simulador.

Entre las líneas futuras se está estudiando la posibilidad de realizar prácticas de programación para que los alumnos de asignaturas del Área de Ingeniería Telemática desarrollen módulos, algoritmos y tecnologías de red adicionales sobre el propio simulador. Por otro lado, se va a mejorar OpenSimMPLS con la incorporación de soporte para IPv6 (*Internet Protocol version 6*) y RSVP (*Resource Reservation Protocol*) y con la adición de nuevas características visuales que permitan obtener una información más detallada y rica.

## AGRADECIMIENTOS

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.

## REFERENCIAS

- [1] M. Kodialam, T. V. Lakshman. *Restorable Dynamic QoS Routing*. IEEE Communications Magazine, Vol 40, Issue 6, June 2002, pp 72-81.
- [2] J. Gozdecki, A. Jajszyk, R. Stankiewicz. *Quality of Service Terminology in IP Networks*. IEEE Communications Magazine, Vol 41, Issue 3, Mar 2003, pp 153-159.
- [3] E. Rosen et al. *Multiprotocol Label Switching Architecture*. RFC 3031, January 2001.
- [4] M. Domínguez-Dorado, F. J. Rodríguez-Pérez, J. L. González-Sánchez, A. Gazo. *Multiplatform and Opensource GoS/MPLS Simulator*. II European Modelling and Simulation Symposium (EMSS2006). International Mediterranean Modelling Multiconference (I3M2006). Barcelona, 2006, pp 529-537.
- [5] M. Domínguez-Dorado, F. J. Rodríguez-Pérez, J. L. González-Sánchez, J. L. Marzo, A. Gazo. *An Architecture to Provide Guarantee of Service (GoS) to MPLS*. IV Workshop in G/MPLS Networks, 2005.
- [6] J. L. Marzo, E. Calle, C. Scoglio, T. Anjali. *QoS Online Routing and MPLS Multilevel Protection: A Survey*. IEEE Communications Magazine, vol 41, Issue 10, Oct 2003, pp 126-132.
- [7] G. Ahn, W. Chun. *Simulator for MPLS Path Restoration and Performance Evaluation*. Chungnam National University, Korea, 2001, pp 32-36.
- [8] MPLS Simulator: <http://www-entel.upc.es/xavierh/mpls> (07/05/2007).
- [9] G. Ahn, W. Chun. *Design and Implementation of MPLS Network Simulator*. 15th International Conference on Information Networking, 2001, pp 694.
- [10] OpenSimMPLS: <http://gitaca.unex.es/opensimmpls> (07/05/2007).
- [11] <http://totem.info.ucl.ac.be/index.html> (07/05/2007).
- [12] <http://www.opnet.com> (07/05/2007).
- [13] S. H. Thomke. *Simulation, learning and R&D performance: Evidence from automotive development*. Research Policy, Volume 27, Issue 1, May 1998, pp. 55-74.
- [14] T. J. Overbye, P. W. Sauer, C. M. Marzinzik, G. Gross. *A user-friendly simulation program for teaching power system operations*. IEEE Transactions on Power Systems, Vol 10, Issue 4, Nov 1995, pp 1725-1733.
- [15] T. de Jong, W. R. van Joolingen. *Scientific Discovery Learning with Computer Simulations of Conceptual Domains*. Review of Educational Research, Vol. 68, Issue 2 (summer, 1998), pp. 179-201.
- [16] A. Parush, H. Hamm, A. Shub. *Learning histories in simulation-based teaching: the effects on self-learning and transfer*. Computers and Education, Vol. 39, Issue 4, Dec 2002, pp. 319-332.



**Manuel Domínguez-Dorado**, Zafrá (Badajoz). España, 1977. Ingeniero e Ingeniero Técnico en Informática (Universidad de Extremadura), 2004. Actualmente es estudiante de doctorado y becario FPI del área de Ingeniería Telemática del Departamento de Ingeniería de Sistemas Informáticos y Telemáticos (UEX). Su trabajo se centra en MPLS-TE, QoS routing y encaminamiento interdominio.



**Fco. Javier Rodríguez-Pérez**, Huelva. España, 1976. Ingeniero en Informática (Universidad de Extremadura), 2000. Actualmente es profesor del área de Ingeniería Telemática del Departamento de Ingeniería de Sistemas Informáticos y Telemáticos (UEX). Su trabajo se centra en QoS routing y Garantía de Servicio sobre MPLS-TE.



**José Luis González-Sánchez**, Dr. Ingeniero en Informática (Universidad Politécnica de Cataluña). Profesor titular, coordinador del Área de Ingeniería Telemática del Departamento de Ingeniería de Sistemas Informáticos y Telemáticos (UEX) e investigador principal del grupo GITACA. Su trabajo se centra en Calidad de Servicio, protocolos de comunicación, MPLS-TE y seguridad.