

# INTERFICIE SIMPLE PARA LA UTILIZACIÓN DEL NS

*José M. Yúfera Gómez*

Departamento de Ingeniería Telemática  
Universidad Politécnica de Cataluña  
yufera@mat.upc.es

*Mari Carmen Domingo Aladrén*

Departamento de Ingeniería Telemática  
Universidad Politécnica de Cataluña  
mari.carmen.domingo@upc.es

## RESUMEN

Para que el alumno pueda asimilar mejor la complejidad de las redes telemáticas, resulta interesante el uso de simuladores y animadores que le permitan visualizar y clarificar conceptos.

En este artículo se presenta una interficie para el Network Simulator (NS) que evita que el estudiante deba invertir cantidades excesivas de tiempo en la programación y así pueda centrarse en el estudio de la temática relacionada con las redes de ordenadores.

## 1. INTRODUCCIÓN

Los cursos que se realizan en el área de la Ingeniería Telemática pueden verse sumamente beneficiados por la utilización de un soporte a los materiales clásicos de las asignaturas, basado en simulaciones y animaciones. De hecho, podemos pensar en varias situaciones en las que las simulaciones y/o animaciones pueden resultar de ayuda. Por ejemplo, como soporte en las clases teóricas mediante demostraciones y en clases de laboratorios, donde el estudiante puede afianzar o profundizar en sus conocimientos mediante la realización de prácticas (ya sea con la simulación de escenarios prediseñados o no). Es importante ofrecer a los estudiantes formas de experimentación con diferentes protocolos y conceptos de ingeniería telemática.

Además, debemos considerar otro valor añadido que poseen las herramientas de simulación de redes de ordenadores: poder trabajar con entornos no disponibles en el laboratorio y la posibilidad de que el estudiante, a un nivel ya superior, cambie y mejore las prestaciones del simulador.

El problema que aparece en estos casos es la relación entre la herramienta de simulación y la gran dosis de tiempo que deben invertir los usuarios a la hora de escribir código para este simulador o, en general, cualquier otro, cuando el objetivo de una asignatura puede no ser la programación en sí.

Por estos motivos, y considerando el beneficio que se obtiene con la utilización de herramientas de simulación en entornos de enseñanza en las áreas de la ingeniería telemática, se hace necesaria la creación de una interficie que haga innecesaria la programación de código para la herramienta de simulación.

## 2. EL NETWORK SIMULATOR (NS)

A la hora de preguntarnos cómo debía ser la interfaz, aparecieron una serie de conceptos evidentes: simple, gráfica o visualmente

clara y atractiva, con ayudas escuetas y precisas acerca de los parámetros y opciones disponibles. Ante todo, debe tratarse de una herramienta que permita a los estudiantes tener una clara conciencia de qué tipo de entorno de red están simulando. Al ser los propios estudiantes quienes deben introducir los parámetros idóneos de cara a una simulación específica, no se lanzan a simular sin haber antes reflexionado, con lo que una vez ejecutada la simulación y puesto en marcha el animador pueden comprobar si sus hipótesis acerca de los resultados a obtener son correctas o no y así pueden llegar a entender mejor la salida.

A partir de aquí, queda estudiar el simulador para el que se realizará la interficie y adaptarla a sus necesidades.

Nuestra interficie está implementada pensando en el Network Simulator (NS) [1], uno de los simuladores más extendidos en la simulación de entornos telemáticos que deriva del simulador de redes REAL y está compuesto por software no comercial. Es un simulador de eventos discretos orientado principalmente al estudio de las redes y utilizable tanto a nivel de investigación como a nivel educativo.

La versión actual del NS permite simular, y por tanto comprender, conceptos relacionados, por ejemplo, con protocolos TCP/IP, UDP, servicios diferenciados, encaminamiento unicast y multicast, entornos y modelos Bluetooth, wireless, satélites y colas, con diferentes aplicaciones simulables: FTP, telnet, http (cliente, servidor o caché WEB), etc. Con él es posible utilizar un animador de red (NAM) que ofrece una GUI para animaciones post-simulación. Este animador permite visualizar tanto los componentes de una red como las trazas de los paquetes intercambiados, facilitando además que el usuario realice interacciones durante las animaciones (figura 1).

El NS es rápido, de modo que el usuario no debe esperar mucho tiempo para ver finalizada la simulación (útil tanto a nivel de demostraciones interactivas en clases de teoría y aplicación, como en prácticas de laboratorio realizadas).

En su conjunto, el NS es por tanto una herramienta que permite la simulación, animación y emulación en un entorno de comunicaciones. Es decir, el simulador permite evaluar los componentes de un determinado escenario telemático; la herramienta de animación puede utilizarse para explicar, p.e., cómo funciona un protocolo utilizando como entrada los resultados de la simulación; y, finalmente, el emulador, un programa que funciona en condiciones reales, puede utilizarse para probar protocolos nuevos o variaciones, comprobando su efecto en entornos reales. Además, también puede utilizarse para obtener información de una red real y, p.e., utilizarla como entrada para el simulador para la evaluación de/en escenarios.

A partir de aquí podemos continuar con el desarrollo de nuestra interficie.

### 3. LA INTERFICIE DESARROLLADA

Para el desarrollo de nuestra interficie partimos de que el NS es un simulador orientado a objetos escrito en C++, que utiliza Otcl, una versión orientada a objetos del Tcl, como interfaz de comandos y configuración, es decir, un intérprete Otcl como frontend. Así pues, el usuario debería escribir (programar) un objeto tcl para el intérprete; éste generará un objeto para el animador y otro para el programa de análisis gráfico.

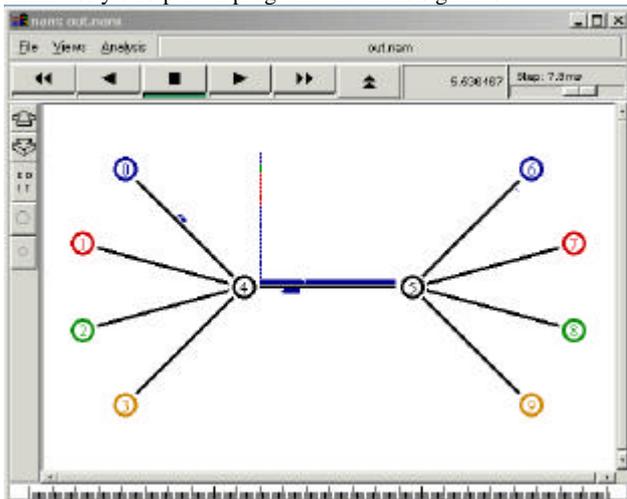


Figura 1. Captura de pantalla del NAM.

Dado que el simulador está basado en Tcl/Tk [2], decidimos programar nuestra interficie con el mismo lenguaje. El Tcl (Tool Command Language) es un lenguaje de programación interpretado y multiplataforma creado por John K. Ousterhout y su equipo de la Universidad de California que actualmente está siendo desarrollado por Sun Microsystems Laboratories. Tk fue también creado por Ousterhout y proporciona un intérprete denominado Wish que añade comandos capaces de crear interfaces gráficas de usuario. Para implementar nuestra aplicación nosotros hemos utilizado concretamente Visual Tcl [3], un entorno de programación y desarrollo muy parecido a Microsoft Visual Basic que resulta muy adecuado para la creación de elementos gráficos tales como ventanas, botones, menús, barras de scroll, etc que se denominan widgets.

Decidimos dividir la pantalla de nuestra interficie en dos partes o frames: en la primera, una serie de ventanas permiten al usuario introducir los datos para generar la simulación; en la segunda, un área de texto, el usuario puede observar los datos introducidos (figura 2).

Con el fin de diseñar nuestra aplicación hemos tenido en cuenta la estructura de un fichero típico tcl dispuesto para ser simulado con el NS. De esta manera hemos estructurado el frame de introducción de datos de nuestra aplicación, y siguiendo una estructura de ventanas, básicamente en las siguientes partes:

-Topología de la red. En esta sección se introducen los parámetros relativos al número de nodos, así como al tipo de enlaces que se va a establecer entre ellos, su ancho de banda, su retardo, el tipo de cola, etc.

-Transmisión de datos. Con el fin de que se establezca un flujo de datos de un enlace a otro es necesario crear los agentes, cuya

misión consistirá en enviar o recibir datos y que estarán asociados a un nodo determinado. Deberán declararse los valores de toda una serie de parámetros relevantes relacionados con estos agentes y que variarán en función del tipo de agente que estemos caracterizando, haciendo por ejemplo referencia al tamaño de los paquetes a enviar, la velocidad de transmisión o el tipo de aplicación a simular.

-Tráfico. Finalmente también será necesario establecer un sentido para el tráfico de nuestra red y deberemos indicarle a nuestra interfaz cuándo queremos que comience y finalice la transmisión de datos.

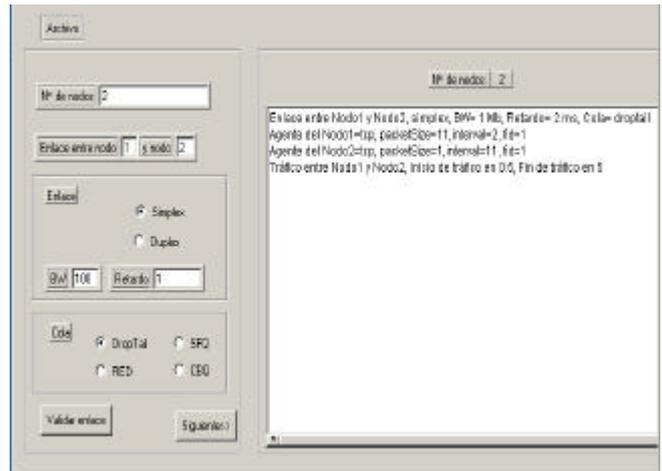


Figura 2. Captura de pantalla de la interfaz.

Conforme el usuario va introduciendo nuevos datos éstos se van validando y van apareciendo en un área de texto situada a la derecha de nuestra aplicación con el fin de que el usuario advierta en todo momento cómo está construyendo la red y, en caso de haber cometido algún error en la introducción de los parámetros, pueda detectarlo y corregirlo sin necesidad de volver a rellenar todos los datos de la simulación.

Finalmente, y una vez terminada la introducción de los parámetros, el programa añade internamente toda una serie de instrucciones estándar para crear un objeto simulador, ejecutar el NS y el NAM, abrir un fichero de traza e inicializarlo, definir un procedimiento que lo cierre y finalice NAM, etc.

### 4. CONCLUSIONES

Se ha presentado una interficie sencilla para el Network Simulator. Su utilización, que evita la necesidad de programar los objetos del simulador, facilita que el estudiante se centre en los temas relacionados con las redes telemáticas, permitiéndole profundizar y comprender conceptos mediante simulaciones y animaciones de entornos de telecomunicaciones.

### 5. REFERENCIAS

- [1] <http://www.isi.edu/nsnam/ns/ns-documentation.html>
- [2] Welch, B.B., *Practical Programming in TCL and Tk*, Prentice Hall, 1997
- [3] Young, D., *The Visual Tcl Handbook*, Prentice Hall, 1997