

INCORPORACION DE UN SIMULADOR GRÁFICO DE REDES EN UN OBJETO DE APRENDIZAJE REUTILIZABLE.

P. BELZARENA, V. GONZALEZ-BARBONE.

Departamento de Telecomunicaciones. Instituto de Ingeniería Eléctrica. Facultad de Ingeniería. UdelAR (Universidad de la República). Montevideo, Uruguay.

La enseñanza práctica de redes de datos tropieza con la dificultad de construir topologías variadas en un laboratorio, o queda limitada al trabajo sobre una topología existente; este vacío se intenta llenar con software de simulación. No obstante, la mayoría de los simuladores existentes tienen escasas cualidades didácticas, son muy limitados o cuestan dinero. El simulador de redes de datos EasySim es una herramienta gráfica fácil de usar donde pueden definirse topologías y parámetros de red, correr una simulación, ver en animación el movimiento de paquetes, registrar los resultados para análisis o visualización posterior. En este trabajo se describen las cualidades esperadas de un simulador para enseñanza, se analizan algunas propuestas de simuladores existentes y luego las características de EasySim. Se describe el uso y la funcionalidad de este simulador, analizando brevemente su arquitectura de núcleo. A partir del costo de generación de contenidos se destaca la necesidad de crear objetos reutilizables siguiendo algún protocolo ampliamente aceptado, en particular el protocolo SCORM, del cual se exponen brevemente sus características. El simulador EasySim, junto con un instructivo, manual de usuario, una simulación para TCP y dos cuestionarios de autoevaluación se integran en un paquete SCORM reutilizable. Se anotan algunas dificultades halladas durante el trabajo y las soluciones aplicadas, finalizando con algunas reflexiones sobre el uso de objetos de aprendizaje y su creación.

1. Introducción

La enseñanza efectiva del tema Redes de Datos requiere experimentación, con varios objetivos imprescindibles: ubicar al estudiante en la capa de protocolos en la que se encuentra, visualizar las características de hosts, enlaces y puertos, analizar su comportamiento en diferentes topologías y situaciones, verificar la actuación de los diferentes algoritmos implicados, examinar situaciones límite de retardos y pérdidas, comprobar los problemas de ruteo, verificar el comportamiento de las diferentes soluciones de ruteo dinámico, por nombrar sólo situaciones comunes. La experimentación libre o guiada con redes simples o medianamente complejas es un siguiente paso.

Resulta muy difícil disponer de laboratorios con equipamiento adecuado y horario suficiente para un número creciente de estudiantes, tanto por costo como por disponibilidad locativa. La excelente propuesta del Internet Lab de la Universidad de Virginia [16] requiere cuatro PCs, cuatro routers, 4 hubs, un switch para conmutar a cada PC, pantalla, teclado, ratón, un rack y una UPS por cada puesto de trabajo, un ideal difícil de alcanzar en la mayoría de los casos.

Por otra parte, aún cuando se cuente con un laboratorio de redes, es difícil que el estudiante pueda disponer del mismo cuando le sea necesario en su proceso de aprendizaje.

Un simulador, si bien no puede sustituir el trabajo directo con los equipos, puede proveer en cambio facilidad de acceso, manejo de topologías complejas, rapidez en el armado y visualización gráfica de performance, análisis de largos de cola y pérdidas, adaptaciones de ruteo por saturación o caída de un enlace y un sinnúmero de situaciones que, si bien no son reales, emulan de cerca la realidad.

El uso de simuladores permite, además, el estudio y la práctica a domicilio, algo no menor en aulas saturadas con disponibilidad de equipamiento escasa y sobre utilizada.

El simulador de redes de datos EasySim [5] es una herramienta gráfica fácil de usar donde puede definirse una amplia gama de topologías fijando los respectivos parámetros de enlaces, colas e interfaces, correr una simulación, ver en animación el movimiento de paquetes, registrar los resultados para su análisis y visualización posterior. Es además una herramienta de experimentación e investigación. Este trabajo se centra en los aspectos didácticos de EasySim, su empaquetamiento en un objeto de aprendizaje SCORM donde se propone un laboratorio sobre el protocolo TCP, y la carga del objeto en la conocida plataforma de aprendizaje Moodle. En el paquete se marcan los objetivos, se dan instrucciones sobre la realización de la práctica, se presenta el simulador con su manual de usuario, se propone un cuestionario diagnóstico previo de autoevaluación, se provee una topología simple para análisis e instrucciones para realizarla, más un cuestionario final de autoevaluación referido a la práctica propuesta. El simulador queda disponible, dentro del objeto de aprendizaje, para la libre experimentación por parte del alumno. En las conclusiones se consignan logros y dificultades de la tarea, las soluciones propuestas, más algunas reflexiones sobre eLearning y generación de contenidos para la enseñanza de las Telecomunicaciones. Finalmente, debe destacarse que este trabajo ha sido realizado en el marco del proyecto CITA'2 [2] financiado por el programa Alfa.

2. Simuladores de redes de datos

Entre las cualidades requeridas de un simulador para enseñanza se destacan:

- facilidad de manejo, tanto para el armado de topologías como para la configuración de parámetros.
- visualización gráfica:
 - de la topología.
 - de la variación de valores en distintos puntos (pérdidas, largo de cola, tiempo de retorno).
 - de rutas, trayectorias de paquetes, pérdidas.
- multiplataforma, no debe depender del sistema operativo del alumno.
- fácil de instalar.
- nulo o muy bajo costo.

Algunos simuladores conocidos:

- CNet v2.0.10 [10] permite experimentar con diferentes protocolos y topologías consistentes en combinaciones de enlaces punto a punto y segmentos Ethernet IEEE 802.3. Fue diseñado específicamente para uso en cursos de grado. Presenta una buena interfaz gráfica de usuario. Corre en Linux, Unix DEC-OSF/1, FreeBSD, NetBSD, SunOS 4.x, Solaris 5.x, y SGI IRIX (Rel. 5 o 6), pero no en MS Windows ni Apple MacIntosh. Es preciso bajar el código fuente y compilar, luego de editar el Makefile y ajustar varios #defines de C en config.h. Esta última característica lo hace poco adecuado para el uso masivo por estudiantes.
- Ns [15][1] es un simulador de eventos discretos orientado a la investigación en redes. Soporta simulación de TCP, ruteo, y protocolos de multidifusión (multicast) sobre redes cableadas o inalámbricas. Dispone de dos jerarquías de clases: compiladas en C++ para una simulación eficiente y scripts de usuario en OTcl para definir la topología, los protocolos y las aplicaciones que se desea simular. Deben bajarse diversos paquetes y los fuentes para compilar. El conjunto de paquetes necesario requiere 320 MB para la compilación. Corre en FreeBSD, Linux, SunOS y Solaris; también puede correrse en MS Windows mediante Cygwin. Escenarios simples deben correr en una máquina media, pero escenarios grandes pueden requerir grandes cantidades de memoria. Las animaciones requieren el paquete adicional nam, Network Animator [8]. Ns es universalmente aceptado como simulador de prueba para investigación en redes, pero resulta poco apto para la enseñanza.
- NCTUns [11] es un simulador y emulador de redes capaz de simular diferentes protocolos en redes cableadas o inalámbricas basada en una nueva metodología que proclama poseer

capacidades superiores a NS y OpNet. Usa directamente la pila de protocolos TCP/IP de Linux con el propósito de generar resultados de alta confiabilidad. Requiere la compilación e instalación de un nuevo kernel.

- OpNet [13], originado en un desarrollo en el MIT (USA), tuvo su primera versión comercial en 1987. Ofrece múltiples capacidades de simulación, animación y análisis. Los modelos pueden ser transferidos sin modificación entre plataformas Windows 2000, XP, Linux y Sun Solaris.
- CCNA, para autoestudio, de Cisco [3], es un producto comercial orientado al manejo de los equipos de sus fabricantes y a los exámenes de certificación. Este simulador permite una red con hasta 200 routers y switches, soporta 45 modelos de router y contiene más de 100 propuestas de trabajos de laboratorio.

Una revisión de simuladores de red puede verse en [12].

Ninguno de los ejemplos anteriores cumple las condiciones anotadas antes respecto a simuladores para aprendizaje, ya sea por sus dificultades de instalación masiva por los estudiantes, por razones técnicas o por restricciones de licencia y los costos involucrados. Esto originó el proyecto EasySim. El proyecto es un desarrollo de código libre del Instituto de Ingeniería Eléctrica de la facultad de Ingeniería, UdelaR, Uruguay. EasySim es un proyecto aún en pleno desarrollo. En estas primeras etapas del desarrollo del simulador han participado dos grupos de estudiantes [5][6].

3. Uso de EasySim

El usuario comienza definiendo una topología de red, en base a comandos o mediante una interfaz totalmente gráfica. Ambos modos operativos refieren a un único archivo de configuración, cuyo contenido son comandos; estos se modifican desde la interfaz gráfica o por digitación directa en una ventana de EasySim. El usuario crea y configura parámetros en tres tipos de entidades:

- Host, una máquina de usuario final con acceso a la red, sin otras capacidades.
- Router, un equipo con al menos dos conexiones de red y capacidades de ruteo.
- Link, enlace entre Host y Router, con una cierta capacidad.

Las tres entidades se distinguen por un nombre único, una posición implementada internamente mediante coordenadas (x, y), y una probabilidad de caída, eventualmente 0, que juega durante la simulación. Cada Host puede tener uno o más generadores de tráfico de diverso tipo: Exponencial, Poisson, Normal, Uniforme, Determinístico y On-off. Cada Router tiene una tabla de ruteo, la cual deberá ser definida. Cada Link comprende entidades origen y destino con sus respectivas interfaces, un ancho de banda o capacidad del enlace, un valor de retardo en la transmisión por este enlace, y una probabilidad de pérdida de paquetes. El usuario define también los generadores de paquetes, así como las interfaces de red con su capacidad de cola y disciplina.

Definida la topología y los parámetros de las entidades, el usuario puede correr una simulación fijando un cierto tiempo; opcionalmente puede pedir una animación del movimiento de paquetes en las entidades que elija. Las colas y los generadores admiten presentación gráfica de resultados. Las colas permiten ver buffer, paquetes y retardos versus tiempo, distribución acumulada de retardos y tráfico a la salida. Los generadores permiten ver distribución de tiempo entre paquetes, y Kbps versus tiempo. Una tabla final de estadísticas muestra valores para las distintas entidades.

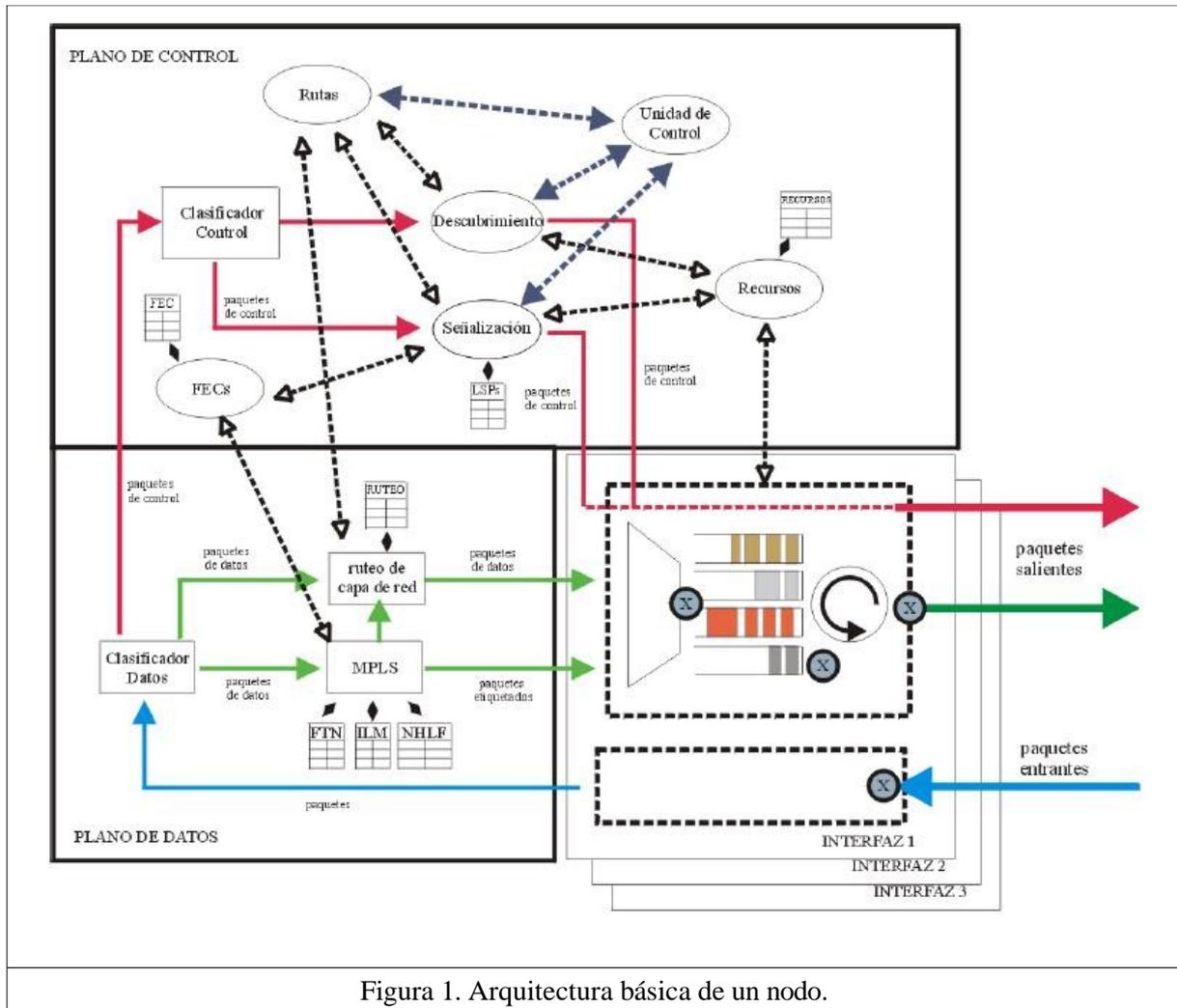


Figura 1. Arquitectura básica de un nodo.

4. Funcionalidad de EasySim

EasySim es un simulador enteramente programado en Java, lo cual permite que sea utilizado sobre cualquier tipo de plataforma o de Sistema Operativo. El único requerimiento es que se cuente con una máquina virtual java (Java Virtual Machine, JVM) compatible instalada. El simulador puede bajarse fácilmente por la red en forma de applet o de programa ejecutable.

Desde el punto de vista de la arquitectura hay cuatro paquetes bien separados y con una interfaz clara entre ellos: el simulador de eventos discretos, el núcleo de simulación de redes, la interfaz de configuración y la interfaz de análisis de resultados.

El simulador de eventos discretos se basa en el software de dominio público SimJava [7].

La interfaz de configuración resulta muy amigable para construir la red: agregar hosts, enlaces, enrutadores, y configurar cada uno de estos componentes es muy sencillo.

Los resultados de la simulación se presentan gráficamente y a través de informes, lo cual facilita el análisis.

La arquitectura del núcleo del simulador de redes busca por un lado que el simulador sea simple para no agregar complejidades innecesarias al estudiante y por otro que pueda ser usado para estudiar

arquitecturas de redes complejas. En la figura 1 se muestra la arquitectura básica de un nodo, donde se puede ver que el nodo maneja dos planos: el plano de control y el plano de datos. En el plano de control se encuentran los componentes que permiten implementar diferentes mecanismos de establecimiento de rutas, protocolos de reserva de recursos, etc. En el plano de datos se encuentran componentes que permiten procesar los paquetes que arriban al nodo: definir la interfaz de salida en base a la ruta y clase de servicio, enviarlo a la cola correspondiente, aplicarle diferentes políticas configurables de descarte y despacho de paquetes por la interfaz.

Los diferentes tipos de generadores de paquetes disponibles en un host entregan los paquetes a diversos agentes. La implementación actual incluye: agentes UDP, TCP Reno o TCP Vegas. Los agentes envían los paquetes a la interfaz correspondiente, la cual a su vez los enviará a los nodos intermedios hasta llegar al host en el otro extremo de la red.

5. Los objetos de aprendizaje y sus estandarizaciones

La enseñanza electrónica, ya sea a través de Internet o basada en computador, conocida actualmente como eLearning, ha probado ser mucho más costosa en generación de contenidos de aprendizaje de lo previsto originalmente. Esto llevó a considerar la reutilización de los mismos objetos en diferentes cursos, estandarizando sus formatos. Uno de los estándares más conocidos es SCORM (Sharable Content Object Reference Model). SCORM define la comunicación entre el estudiante y la plataforma de aprendizaje (Learning Management System, LMS), así como diferentes organizaciones para los contenidos, más un conjunto de metadatos o información complementaria donde se identifica el tema, objetivo de aprendizaje, autores, versión, institución y toda la información necesaria para garantizar las cuatro metas propuestas por SCORM para los objetos de aprendizaje [9]:

- Reutilización: independencia del contexto, aplicable en diferentes situaciones y plataformas de aprendizaje.
- Interoperabilidad: el contenido funcionará en múltiples ambientes de hardware o software, sean cuales fueren las herramientas con que fue creado.
- Durabilidad: no requerirá modificaciones ante actualizaciones de las plataformas de aprendizaje o el sistema operativo.
- Accesibilidad: podrá ser identificado y ubicado en el momento y lugar donde se lo necesite, sin necesidad de acceder al contenido interno del mismo ni pagar licencias.

Un objeto de aprendizaje o paquete SCORM es un archivo comprimido (.zip) que puede colocarse como parte de un curso en una plataforma de aprendizaje con soporte SCORM y ser ofrecido a los estudiantes, o en un repositorio de objetos de aprendizaje desde el cual obtenerlo para su inclusión en un curso.

6. EasySim como objeto de aprendizaje

La inclusión de EasySim en un objeto de aprendizaje reutilizable se prevé como complemento para un curso de redes de datos de tipo general o especializado hacia el diseño o la performance. El valor didáctico del simulador abarca diferentes aspectos:

- comprensión visual del comportamiento de las redes, alternativo y complementario a la descripción teórica o los ejemplos estáticos diagramados o numéricos.
- verificación de situaciones problemáticas comunes en redes tales como la saturación de los enlaces, la pérdida de paquetes, el aumento del retardo, la caída de un host, router o enlace, el cambio de rutas consiguiente, las variaciones introducidas al usar distintos tipos de generadores.
- el planteo de situaciones de diseño de una red con determinadas características o del estudio de comportamiento de una red de características dadas.
- la propuesta de situaciones de simulación con obtención de resultados, análisis de los mismos y presentación de conclusiones con fines evaluatorios del aprendizaje.

El simulador tiene su propia documentación de uso, concebida para el estudio en cursos de redes de datos. Si bien puede incluirse material básico en el objeto, dando prioridad a la capacidad de reutilización preferimos concebirlo como un objeto autónomo que puede actuar de complemento para otros objetos de aprendizaje de propósitos variados, desde un curso de redes básico hasta el diseño o la predicción de performance.

7. Creación del paquete SCORM

El objeto de aprendizaje incluye:

- páginas HTML, para las instrucciones generales y el manual de usuario.
- el simulador, en un archivo Java comprimido (easysim.jar).
- una página HTML con un applet Java para invocar el simulador.
- dos páginas de cuestionarios con sus correspondientes páginas de respuestas, en HTML con código JavaScript.
- código JavaScript con funciones usadas en la evaluación de los cuestionarios (creaquiz.js) y creación de las páginas respuesta.

La creación del objeto SCORM se realizó con el editor de contenidos del Proyecto Reload [14]. Aunque requiere alguna familiaridad, el editor resuelve eficazmente la creación de los contenidos: una vez definitivo el tipo de empaquetamiento (SCORM), se importan los recursos, se define una organización o secuenciamiento, y se le incorporan los recursos con sus dependencias. El editor genera un conjunto de metadatos muy elemental pero suficiente para el cumplimiento del protocolo, y el empaquetado es sólo elegir una opción de menú en el Editor Reload.

Si bien así se dispone de un paquete en cumplimiento SCORM, es necesario completar un conjunto razonable de metadatos para alcanzar las condiciones de reutilización, interoperabilidad, durabilidad y accesibilidad prometidas por SCORM. El IEEE LOM propone unos 70 metadatos, algunos de los cuales son sólo contenedores, pero para cumplir con el estándar se requiere llenar más de 60 datos. Esta es una tarea mayor, donde se detectan algunas dificultades:

- el IEEE LOM está en inglés; aunque existe una traducción al español [4] las palabras no tienen acepciones inequívocas universales. Los perfiles de aplicación, subconjuntos del LOM definidos para las necesidades de una comunidad, adolecen de la misma dificultad.
- unos cuantos metadatos tienen carácter institucional, departamental o de curso, lo cual requiere consenso a esos niveles.
- son muchos metadatos, es trabajoso llenarlos, difícil alcanzar calidad en la información.
- existen disensos y aún cuestionamientos ante la abundancia y real necesidad de estos metadatos, sobre todo al pensar si van a ser realmente usados.

No parece posible, en el marco de un grupo reducido de docentes preparando material para sus cursos, dar solución adecuada al llenado de los metadatos: se requiere una intervención institucional, y si fuera posible, un consenso o propuesta nivel de alguna organización superior, como ser un perfil de aplicación en español, basado en el IEEE LOM pero acordado entre organizaciones y países. También es necesario un fuerte compromiso de la institución educativa generadora del contenido, uniformizando sus datos propios, así como los de sus departamentos y cursos. Es posible la definición de una plantilla con datos de uso general, la cual puede importarse en el Editor Reload, y cambiar o completar allí las diferencias, seguramente mucho menores y más propias del objeto creado en cuestión. Pero faltan las definiciones más generales.

El objeto creado descrito en este trabajo tiene un conjunto de metadatos elemental sobre origen y contenidos; la profundidad del problema excede los alcances de esta propuesta.

El tema evaluación del aprendizaje también presenta dificultades. SCORM no permite la invocación mutua de objetos SCORM entre sí, ni mantiene comunicación entre los objetos SCORM. En cambio el objeto SCORM sí puede retener un puntaje mínimo de aprobación, e informar a la plataforma de aprendizaje el puntaje obtenido por un estudiante en las pruebas de evaluación. Esto debe realizarse a través de la API (Application Programming Interface) en JavaScript prevista por el propio SCORM, y que la plataforma debe implementar. Si un curso contiene varios objetos SCORM, uno por objetivo educativo, según se recomienda, será la plataforma de aprendizaje quien deberá llevar la historia de aprovechamiento del estudiante, registrando los resultados obtenidos en la evaluación de cada objeto. Esto no constituye un problema, ya que el armado del curso se hará necesariamente en una plataforma de aprendizaje incorporando distintos objetos SCORM, posiblemente provenientes de diversas fuentes.

No obstante, es necesario mantener la evaluación y su puntaje mínimo de aprobación dentro del objeto SCORM, para asegurar su reutilización, incluida la evaluación. Comparado con la facilidad ofrecida por las plataformas de aprendizaje para la creación de pruebas, crear pruebas dentro de SCORM ofrece dificultades mayores. Una solución habitual es crear cuestionarios en páginas HTML incorporando código JavaScript para usar la API JavaScript de SCORM para fijar diversos valores. Esto ya no está al alcance del común de los docentes. Una herramienta de software para crear cuestionarios en cumplimiento SCORM sería una solución.

Para la evaluación del Laboratorio TCP se usaron dos páginas HTML de cuestionarios de múltiple opción; cuando el estudiante ha contestado todas las preguntas y envía las respuestas, se pasa a otra página HTML donde puede ver, para cada pregunta, su respuesta y la respuesta correcta. Al final, obtiene el resultado global como porcentaje bruto y como porcentaje real descontando el acierto aleatorio de las preguntas, dependiente de la cantidad de preguntas y la cantidad de opciones. Funciones de JavaScript incluidas en el objeto SCORM generan estos resultados.

El Laboratorio TCP fue concebido como demostrativo, optándose en consecuencia por la autoevaluación. Tampoco se impusieron requerimientos de precedencia ni mínimos de aprobación, aunque se recomienda seguir un cierto orden de trabajo.

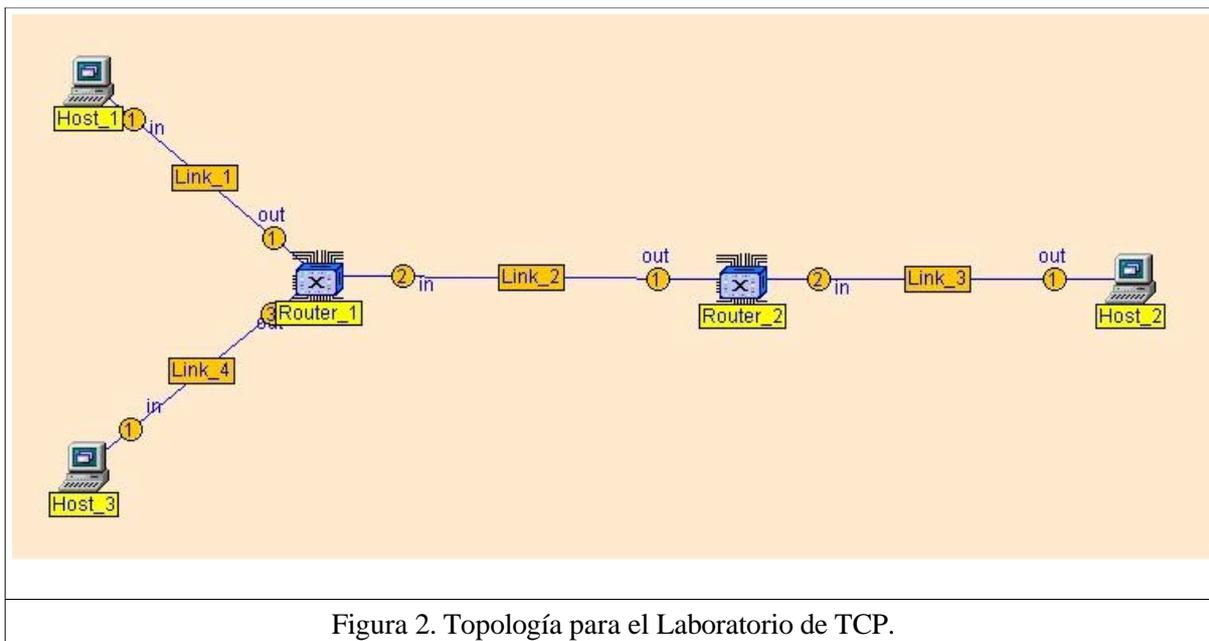
La plataforma de aprendizaje utilizada fue Moodle, y el paquete se encuentra en la sección Redes y Sistemas del Proyecto Alfa Cita'2 [2].

8. Laboratorio de TCP

Los objetivos del laboratorio propuesto son:

- Familiarizar al estudiante con el uso básico del simulador EasySim.
- Comprender aspectos básicos de la performance del protocolo TCP: throughput, impacto del retardo de ida y vuelta, efecto de los buffers del cuello de botella, etc.
- Comprender los detalles de los mecanismos de control de congestión en TCP: Slow Start, Congestion Avoidance, Fast Recovery, Fast Retransmit.

Para tal fin se propone en la configuración entregada al estudiante la topología de la figura 2. La práctica se divide en dos partes. En la primera se le pide que evalúe algunas cosas básicas de la topología de la red que luego le serán de utilidad, tales como dónde se encuentra el cuello de botella, cuántos paquetes se pueden almacenar en la cola de paquetes del enlace cuello de botella, cuáles son las diferentes componentes del retardo ida y vuelta, cuál es el valor máximo del mismo en esta configuración, etc.



En la segunda parte se le solicita al estudiante que corra la simulación y realice un análisis detallado del intercambio de paquetes entre los hosts. Este análisis, aunque simplificado, es el que el podría hacerse en un laboratorio con programas como tcpdump o Ethereal. En base a este experimento se pide al estudiante que analice por qué se utilizan los diferentes mecanismos de control de congestión y por qué se producen los cambios de uno a otro a lo largo de la simulación. Por último se le solicita obtener el valor del throughput en dos configuraciones diferentes donde se varía el retardo de los enlaces. Esta última etapa tiene como objetivo que el estudiante comprenda los factores que afectan el throughput en una conexión TCP y en particular el efecto del retardo ida y vuelta de la conexión.

9. Evaluación de uso

El objeto de aprendizaje fue evaluado por varios estudiantes. Entre las observaciones realizadas cabe destacar las siguientes, como incorporaciones futuras:

- Conveniencia de agregar en las respuestas a los cuestionarios una fundamentación o explicación de la respuesta correcta.
- Es inconveniente bajar un archivo de configuración a la máquina del usuario para correr la simulación propuesta como ejercicio; debería poder seleccionarse esta configuración directamente en el objeto.

La impresión recogida en general fue buena, sobre todo por la posibilidad de experimentar en redes fácilmente, y de poder hacerlo desde la casa o cualquier máquina disponible sin necesidad de concurrir a laboratorios. Fue señalado por los estudiantes, correctamente, que la experiencia con equipos reales es también necesaria.

10. Conclusiones

Si bien se concibe este objeto de aprendizaje como complementario en un curso de contenidos aportados por otros objetos, la posibilidad de visualizar, comprobar y experimentar en redes de datos mediante un simulador gráfico de manejo razonablemente intuitivo constituye una considerable ventaja didáctica a muy bajo costo: basta una plataforma relativamente modesta y una máquina virtual Java; no tiene problemas de licencias. El empaquetamiento como objeto SCORM permite aprovechar un desarrollo de muchas horas, en forma inmediata, simplemente incorporando el objeto como parte de cualquier curso impartido a través de una plataforma de presentación de contenidos capaz de importar

objetos SCORM.

El simulador EasySim es un producto en evolución. Este trabajo se basó en la versión 1. Una versión 2 [6] en etapas finales de desarrollo incorpora mejoras en la interfaz de usuario, la graficación de resultados, las animaciones y los protocolos soportados, incluyendo MPLS. No obstante, la imposibilidad de encontrar simuladores de red con las condiciones necesarias para el aprendizaje ha llevado a usar ya esta versión como herramienta, de ahí la iniciativa de ofrecerlo en un objeto SCORM reutilizable.

Si bien el empaquetamiento SCORM no resulta tan sencillo como pareciera a primera vista, y existen dificultades para crear conjuntos de metadatos sólidos, así como para incluir una buena evaluación, se considera necesario continuar los esfuerzos para crear contenidos y difundir el uso de objetos de aprendizaje reutilizables en base a este protocolo. Una buena definición institucional, la adopción o generación de un perfil de aplicación de LOM para idioma español, una mayor investigación y productos libres para generación de evaluaciones, serían algunos pasos decisivos hacia una más prolífera generación de contenidos.

Referencias.

1. Altman, Eitan; Jiménez, Tania. *NS simulator for beginners*. Universidad de los Andes, Mérida, Venezuela; ESSI, Sophia-Antipolis, France. <http://www-sop.inria.fr/maestro/personnel/Eitan.Altman/COURS-NS/n3.pdf> [acceso 2006-06-10]
2. Belzarena, P; González Barbone, V. *Laboratorio TCP con simulador de redes (objeto de aprendizaje SCORM)*. Proyecto Cita'2 Programa Alfa, Redes y Sistemas. <http://cita2.euitt.upm.es/moodle/> [acceso 2006-06-12]
3. Cisco. *CCNA Network Simulator (CCNA Self-Study, 640-801)*. Boson Software Inc. Cisco Press, Bk&CD edition, Published July 2004, ISBN 1587201313. Descripción: <http://www.bookpool.com/sm/1587201313> [acceso 2006-06-10]
4. IEEE. *Estándar para Metadatos de Objetos Educativos*. IEEE P1484.12.1-2002, 15 de Julio de 2002. Editores de la traducción al español: Luis E. Anido Rifón y Miguel Rodríguez Artacho http://www.gist.uvigo.es/~lanido/LOMes/LOMv1_0_Spanish.pdf [acceso 2006-04-03]
5. Instituto de Ingeniería Eléctrica. *Proyecto EasySim*. Facultad de Ingeniería, Udelar. Montevideo, 2005. <http://iie.fing.edu.uy/~belza/EasySim/webpub/Principal.htm> [acceso 2006-06-15]
6. Instituto de Ingeniería Eléctrica. *Proyecto EasySim V2*. Facultad de Ingeniería, Udelar. Montevideo, 2006. <http://iie.fing.edu.uy/~belza/EasySimV2/> [acceso 2006-06-15]
7. Institute for Computing Systems Architecture. *SimJava*. Division of Informatics, University of Edinburgh. <http://www.dcs.ed.ac.uk/home/hase/simjava/> [acceso 2006-06-15]
8. ISI, Information Sciences Institute. *Nam: Network Animator*. <http://www.isi.edu/nsnam/nam/> [acceso 2006-06-10]
9. Learning Systems Architecture Lab. *SCORM Best Practices Guide for Content Developers*. Carnegie Mellon University. <http://www.lsal.cmu.edu/lsal/expertise/projects/developersguide/>
10. McDonald, Chris. *The cnet network simulator (v2.0.10)*. The University of Western Australia. <http://www.csse.uwa.edu.au/cnet/index.html> [acceso 2006-06-10]
11. NCTU. *NCTUns network simulator*. Department of Computer Science, National Chiao Tung University (NCTU), Taiwan. <http://nsl10.csie.nctu.edu.tw/> [acceso 2006-06-12]
12. NCTU. *Network simulation tools*. Department of Computer Science, National Chiao Tung University (NCTU), Taiwan. [http://nsl.csie.nctu.edu.tw/NCTUnsReferences/Opnet%20is%20a%20network%20simulation%20ool.htm](http://nsl.csie.nctu.edu.tw/NCTUnsReferences/Opnet%20is%20a%20network%20simulation%20tool.htm) [acceso 2006-06-12]
13. OPNET Technologies, Inc. *Modeler network simulator*. <http://www.opnet.com/products/modeler/home.html> [acceso 2006-06-12]
14. RELOAD Project. *Reusable eLearning Object Authoring and Delivery*. Homepage. <http://www.reload.ac.uk/> [acceso 2006-04-17]
15. ISI (Information Sciences Institute). *The Network Simulator – ns-2*.

- <http://www.isi.edu/nsnam/ns/ns-build.html> [acceso 2006-06-10].
16. University of Virginia. Internet Lab. Virginia University.
<http://www.cs.virginia.edu/~itlab/book> [acceso 2006-06-10]
Lista de equipo para el Internet Lab: http://www.cs.virginia.edu/~itlab/book/building_a_rack.html
[\[acceso 2006-06-10\]](#)