

SimIP: un simulador didáctico de redes IP

Guillermo Rigotti

UNICEN – Fac. de Ciencias Exactas-ISISTAN

TE: +54-2293-439682 FAX: +54-2293-439681 Email: grigotti@exa.unicen.edu.ar

Resumen: En este trabajo se presenta SimIP, un simulador de niveles IP y superiores de la arquitectura TCP/IP, orientado al aspecto didáctico. Se busca reproducir de la manera más fiel posible el comportamiento de los protocolos reales. Los frames generados responden exactamente al formato real, lo que permite su análisis con cualquier analizador de protocolos y la conexión de la simulación con redes reales. Una característica interesante es que por cada equipo simulado es posible generar una consola remota, desde la cual, en línea con la simulación, se pueden ejecutar los comandos de dicho equipo; esto permite presentar a los usuarios una consola en la que se ejecuten comandos similares a los de una implementación real. El desarrollo fue realizado en lenguaje OTcl, utilizando el intérprete ampliado para Network Simulator.

Palabras clave: simulacion, tcp/ip

1 Introducción

En el dictado de cursos de redes, es fundamental complementar los conceptos teóricos con actividades prácticas que permitan al alumno fijar y comprender dichos conceptos y aplicarlos a situaciones “reales”. En la mayoría de los casos no se dispone de recursos para la realización de dichas actividades, especialmente cuando se trata de experimentar con redes (intranets) con una cantidad considerable de equipos y routers. Si bien a veces se dispone de una intranet en los laboratorios, es difícil acceder a ella para reconfigurar los equipos o para reconfigurar su topología, por razones de seguridad y de disponibilidad de los equipos para otras actividades.

Por otra parte, aún si se dispusiera de esos recursos, en muchos casos es necesario que los alumnos necesiten adquirir conocimiento sobre las características del sistema operativo de los equipos, lo cual demanda un tiempo considerable. Por eso resulta importante poder separar los conceptos de redes que nos interesan, de aquellas características particulares de cada sistema operativo.

Por ejemplo, si se decidiera utilizar Linux para realizar los trabajos prácticos, nos encontraríamos, en un curso introductorio de redes, con problemas como los siguientes:

—si se quisiera analizar el comportamiento de los procesos que componen la pila TCP/IP, sería necesario conocer elementos del Sistema Operativo, como la estructura `sk_buff`, clonación de buffers, etc.) [1].

—si se deseara experimentar con los conceptos de tablas de ruteo y de reenvío de paquetes, es probable que no hubiera tiempo para tratar el ruteo avanzado de Linux, sino que sólo se quisiera poner énfasis en el ruteo tradicional por dirección de destino.

—la familiarización con los scripts de configuración para iniciar los servicios de red, si bien es simple, requiere también cierto conocimiento y tiempo adicionales.

—es difícil, cuando se trabaja con redes reales, provocar situaciones de error, tales como configuraciones erróneas en las tablas de ruteo para ocasionar redirects ICMP [2], conteos a infinito en un protocolo de tipo Distance Vector, etc.

Estos problemas hacen que se piense en desarrollar ambientes en los cuales se trata de reflejar los conceptos de las redes, de la manera más real posible, pero que insuman un costo de aprendizaje poco significativo por parte de los alumnos y poco tiempo para la configuración de determinadas situaciones por parte de los docentes.

En este trabajo se presenta SimIP, un simulador de redes a nivel IP y superiores que fue desarrollado exclusivamente con fines didácticos y que permite la configuración de intranets y el análisis del tráfico generado, permitiendo además la interacción del usuario con la simulación en tiempo real.

2 Objetivo y características generales

El objetivo de SimIP es ofrecer un entorno para la creación de intranets simuladas, en las cuales se pueda configurar, probar y analizar el comportamiento de los protocolos de nivel IP [3] y superiores del modelo TCP/IP. Es posible también el estudio de algunos aspectos de la adaptación de IP al nivel “link layer”, como por ejemplo la resolución de direcciones y el encapsulado de los datagrams IP sobre los del medio físico, por ejemplo Ethernet o 802.11.

Debido a que el fin de la simulación es facilitar la comprensión del funcionamiento de las redes, se ha puesto énfasis en reflejar exactamente los comportamientos e información generada en los casos reales. No se intenta simular de manera exhaustiva dichos comportamientos, sino sólo aquellos que son de mayor interés conceptual para la comprensión del funcionamiento de las redes.

Las características relevantes de SimIP son:

a- Simplicidad en su instalación y uso: Si se utiliza para analizar el funcionamiento de los protocolos y el tráfico generado y para configurar la intranet, sólo es necesario conocer el uso de los comandos de configuración de una consola remota (una versión simplificada de la de un equipo real) y el uso de un analizador de protocolos. Si se utiliza para crear intranets y definir las características de los equipos involucrados (interfaces, protocolos, aplicaciones, etc) se requiere además un conocimiento básico de la sintaxis de OTcl[4]. Si se desea desarrollar nuevos protocolos y aplicaciones, es necesario conocer la estructura y detalles de la implementación.

b- Orientado exclusivamente al aspecto didáctico: A diferencia de la mayoría de los simuladores de redes, orientados a la evaluación de la performance y/o factibilidad de los protocolos, SimIP está orientado exclusivamente a mostrar el comportamiento de los protocolos implementados y la información que ellos generan. Para esto se requiere que las implementaciones realizadas en el simulador reflejen de manera exacta a las implementaciones reales. El análisis del comportamiento es posibilitado

por la generación en cada equipo y en cada link simulados, de archivos que registran respectivamente las acciones más significativas de los procesos y la información que viaja a por los links. Los procesos representativos de la arquitectura (IP, UDP [5], etc) generan mensajes por cada acción de interés. Por ejemplo, cuando el nivel IP de un router debe enviar un datagram, genera un registro de recepción, uno de envío, y, en el caso de que la interfaz de salida coincida con la de entrada, un registro de la generación de un ICMP redirect.

Los archivos de registro generados por los links contienen los paquetes que han sido transportados, representados como dígitos hexadecimales y en el formato exacto al de los paquetes reales, y adicionalmente en formato pcap, que permite visualizarla con un analizador de protocolos (Wireshark [6], Network Monitor[7], etc.),

c- Referencia de tiempos propia de cada equipo simulado: Para dar mayor realismo a la simulación y poder probar aquellos casos donde es de relevancia la no existencia de un tiempo común a todos los equipos (por ejemplo el análisis de los timestamps request y reply de ICMP), es posible arrancar a los equipos con un tiempo que resulta del reloj común de simulación, más un desplazamiento que puede ser configurado con un valor arbitrario o al azar. En los archivos de captura de los links se muestra el tiempo absoluto de simulación; en los de los equipos, cada acción tomada por los procesos muestra el tiempo absoluto y el tiempo propio de cada equipo, y en las consolas sólo es accesible el tiempo propio del equipo asociado.

d- Interacción del usuario con la simulación: Además de admitir la simulación no interactiva (ingreso de datos, ejecución del simulador, análisis de resultados), es posible, a partir de un cierto momento de la simulación, sincronizarla con el tiempo real, lo que permite, a través de las consolas, la interacción del usuario con el simulador, por ejemplo reconfigurando tablas de ruteo, arrancando aplicaciones, etc.

e- Posibilidad de creación de ambientes de configuración emulando diferentes equipos: Es posible definir en cada una de las consolas conectadas con el simulador, un comportamiento que emule la interfaz y las posibilidades de configuración de un equipo real (Linux, Windows, etc). Estas interfaces (modo texto, gráfico, etc) son independientes del resto del simulador.

f- Posibilidad de conectarse a una red real: Las características que permiten al simulador adaptarse a tiempo real y la generación de frames que se corresponden con los reales, permite su conexión con una red real, por ejemplo a través de una interfaz PPP en el equipo que ejecuta la simulación y un router remoto. Esta característica se ve limitada a aquellas configuraciones en las cuales el tiempo de ejecución de la red simulada no supere al tiempo real.

g- Posibilidad de incorporar desarrollos realizados para Ns[8]: Debido a que SimIP está desarrollado en lenguaje OTcl, y utiliza el intérprete ampliado de OTcl provisto por Ns, resulta simple adaptar las aplicaciones OTcl desarrolladas para Ns a SimIP.

3 Funcionamiento y estructura del simulador

El simulador tiene como entrada principal archivos OTcl, que definen los elementos a simular (equipos, protocolos, etc). Este software es desarrollado por aquellos que desean implementar nuevos protocolos o modificar los existentes. A continuación es

leído un archivo que contiene la topología de la red y las acciones de los protocolos (eventos para el scheduler). Esto es definido por los usuarios del simulador.

Es posible ejecutar la simulación corriendo el simulador de eventos de manera no interactiva, generando sólo información por la salida standard y los archivos de captura producidos por los equipos y los links.

Adicionalmente, es posible poner al simulador en modo servidor de consolas, en el que el tiempo del simulador se sincroniza con el tiempo real, y se habilita un servidor que recibe requerimientos de clientes. Los clientes solicitan al servidor la asociación con un equipo simulado y así obtienen un acceso restringido a las facilidades de ese equipo, que consiste en el uso de las aplicaciones definidas previamente. En modo servidor de consolas, se tiene acceso a la consola principal (desde la que se arrancó el simulador) siendo posible modificar aspectos de la simulación, como por ejemplo definir nuevos equipos, links, aplicaciones, etc. en tiempo de simulación.

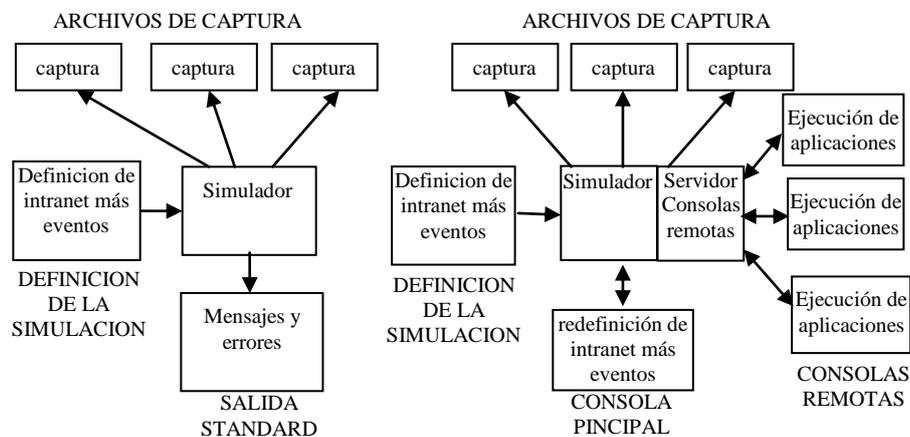


Fig 1. Estructura de SimIP. Izq.: simulación no interactiva. Der.: simulación interactiva con consolas remotas

En la figura 1 se muestra la estructura del simulador para los casos simulación no interactiva (derecha) e interactiva (izquierda).

En el primer caso, el simulador lee archivos que contienen la definición completa de la simulación, éstos incluyen la definición de la topología de la red (lenguaje OTcl) y de los eventos a ejecutar. La ejecución del simulador se produce sin ningún tipo de intervención hasta que se agota la cola de eventos y finaliza; los archivos de captura son generados por los equipos y links en los que se haya activado la captura de datos.

En el caso de simulación interactiva, la simulación no queda definida por el archivo de entrada, sino que permanece esperando eventos ingresados desde las consolas remotas. Para posibilitar esta interacción, se agrega un servidor de consolas ligado al simulador: éste atiende en un port específico requerimientos de clientes (consolas remotas), conectando cada consola a un equipo simulado. Las consolas remotas son procesos residentes en el equipo donde corre la simulación o en un equipo remoto, y pueden ser implementadas utilizando cualquier lenguaje, en modo texto o gráfico; su objeto es que un usuario ejecute aplicaciones en el equipo asociado y observe la salida obtenida. El rol de la salida standard cambia, ya que se convierte en otra consola con mayor privilegio, desde la cual es posible introducir cualquier evento al simulador.

3.1 Archivos

Los archivos de definición de la simulación contienen tres clases de información.

El primer tipo de información lo constituyen los archivos fuente en OTcl que definen las clases necesarias para implementar los elementos de la simulación (equipos, links, protocolos, aplicaciones, etc). A continuación se define la topología de la red a simular, instanciando equipos y demás elementos, y por último se definen los eventos en la simulación, por ejemplo ejecución de aplicaciones, envío de datagrams, etc. Es posible definir aquí que en un momento dado de la simulación, se pase a modo interactivo (activación del servidor de consolas).

Los archivos de captura tienen por objeto registrar las actividades de los diferentes procesos en los equipos y la información que circula por los links. Cada equipo o link generará su archivo de captura, desde el momento en que se active dicha opción.

El contenido de los archivos generados por los equipos consiste en descripciones de las acciones que realiza cada proceso. Esta descripción está orientada a resaltar el comportamiento de los diferentes protocolos. Es posible adaptar estos mensajes a cada caso, poniendo énfasis en ciertos comportamientos de los procesos. Cada línea indica el tiempo absoluto de la simulación y el tiempo propio del equipo.

Los archivos generados por los links reflejan exactamente el formato de los frames que se observarían en una implementación real. Se incluye desde el link layer (pila TCP/IP). Además se genera una salida en formato .pcap que puede ser analizada con distintos analizadores de protocolos.

3.2 Consolas remotas

El servidor de consolas remotas, tiene por objeto posibilitar el acceso a los equipos configurados en la simulación como si se tratara de equipos reales, desde procesos independientes residentes en el mismo equipo o en equipos remotos. El servidor de consolas es activado por el simulador, en un tiempo de simulación especificado en el archivo de entrada. Previamente a su activación, la simulación corre normalmente, no existiendo relación entre el tiempo simulado y el tiempo real. Una vez activado el servidor de consolas, el simulador es forzado a sincronizar su tiempo con el tiempo real. La relación existente entre ambos es configurable, por defecto se hace corresponder una unidad de tiempo de simulación con un segundo de tiempo real.

El servidor adquiere un socket TCP en un port predefinido (9000) del equipo donde está ejecutándose; para crear una consola remota, es necesario ejecutar el cliente y conectarse al servidor. Este pide que el cliente asocie esa consola con un equipo simulado (la correspondencia es uno a uno) y luego se está en condiciones de ejecutar y ver resultados de las aplicaciones en ese equipo. Además se cuenta con algunos servicios orientados al aspecto didáctica, p. ej. un help de los comandos disponibles y el envío de mensajes entre consolas.

Las consolas remotas son procesos que actúan como clientes del servidor de consolas. La interacción con el servidor es muy simple, y se limita al intercambio de mensajes para asociar la consola con un equipo simulado, y luego para ejecutar aplicaciones en ese equipo y observar los resultados. El acceso provisto a un usuario desde una

consola se limita a la ejecución de aplicaciones (por ejemplo, pruebas de ICMP, — icmp_test—, configuración del equipo con ip2 [9], etc).

Una consola remota puede ser implementada en cualquier lenguaje y ofrecer una interfaz modo texto o modo gráfico. En la implementación actual, se han desarrollado en OTcl y ofrecen una interfaz modo texto. Las características de la interfaz pueden responder a un tipo de equipo en particular. Si se desea ofrecer utilidades de red de determinados equipos, éstas deben estar implementadas como aplicaciones en el simulador, y ser habilitadas en los equipos simulados.

Desde las consolas sólo es posible invocar a “aplicaciones” definidas en los equipos. Estas aplicaciones en general son específicas de red, por ejemplo ejecutar el servidor echo, ejecutar ip2 para configuración, etc. Hay otras aplicaciones que no están relacionadas con funciones de red en sí, sino que están orientadas a facilitar el uso del simulador, por ejemplo, la aplicación “help” que muestra el conjunto de comandos disponible y cómo se utilizan, la aplicación “msg”, que permite intercambiar mensajes entre los usuarios de las consolas, etc. Desde una consola remota, sólo es posible acceder al tiempo de simulación del equipo asociado a ella.

3.3 Aplicaciones

Las aplicaciones representan procesos que se ejecutan en los equipos. En ambientes reales, estos procesos serían todos aquellos que corren en el nivel aplicación de la arquitectura TCP/IP, por ejemplo ping, traceroute, un cliente DNS, etc. Algunas de las aplicaciones simuladas acceden a las facilidades de red a través del mecanismo de ports implementado en el simulador, que les permite comunicarse con el UDP simulado (p.ej.Echo). Otras acceden directamente a los “procesos” internos que conforman el nivel de red (p. ej. ip o icmp); este tipo de acceso se corresponde con el uso de sockets Netlink; son aplicaciones que necesitan acceder o modificar parámetros del nivel de red (p. ej. ip2 e icmp_test). Las aplicaciones pueden ser invocadas desde las consolas remotas, de una manera similar a la que se invocaría una aplicación real en una consola modo texto.

4. Componentes de las redes simuladas

Los elementos que componen las redes a simular son equipos y links. Los equipos se conectan entre sí a través de los links, definiendo así la topología a simular.

Los links representan los diferentes tipos de redes físicas que interconectan a los equipos (nivel “link layer“ de la arquitectura TCP/IP).

Debido a que el objetivo de este desarrollo es el estudio del nivel IP y superiores, las características que nos interesan de los links son: 1-el protocolo en el cual se encapsula IP y la manera de montar IP sobre dicho tipo de link (Información provista en los RFC “IP over ...”); 2- demoras, errores y pérdidas de bloques que puedan producirse, No es de interés el comportamiento de los protocolos de ese nivel (p. ej. el mecanismo de acceso al medio en una red 802.11). Hasta el momento, sólo se ha implementado un link simple broadcast de acceso múltiple, que funciona conectado a

una interfaz Ethernet. Este tipo de link soporta la conexión de varias interfaces de red (residentes en los equipos) y se limita a entregar los frames emitidos por una interfaz al resto de ellas, luego de una demora configurable.

Los equipos representan hosts o routers, y son los elementos que presentan más interés para el objetivo de estas simulaciones, ya que contienen los elementos de los cuales interesa comprender el funcionamiento. Hasta el momento se ha desarrollado Ethernet y ARP en el nivel link layer, IP e ICMP en el nivel de red, y UDP para probar a IP desde el nivel de transporte y posibilitar la creación de aplicaciones. Las aplicaciones desarrolladas son de configuración y chequeo de red y Echo.

En la figura 2 se muestran los principales componentes de un equipo:

Las interfaces comprenden el software (device drivers, buffers, etc) y el hardware (placa de red) que posibilitan que el equipo se conecte a un link y sea capaz de enviar y recibir información a través de él; cada tipo de interfaz tiene sus características propias, el concepto de interfaz se corresponde con el de interfaz o device de Linux.

Los protocolos son los procesos que se comunican con sus pares a través de los niveles inferiores de la arquitectura TCP/IP. Tienen asociado un formato de frame, y pueden encapsular frames de protocolos de mayor nivel. Por ejemplo Ethernet, que encapsula frames ARP e IP; IP, que encapsula ICMP y UDP, etc.

Las aplicaciones representan procesos del nivel aplicación en la arquitectura TCP/IP. Estos pueden tener alcance local (p. ej. ip2, similar a la ip de Linux Kernel 2.4, para configurar la red), o alcance remoto, p. ej. Echo que escucha en un port y responde con el eco. Las aplicaciones son los únicos procesos que pueden ser ejecutados interactivamente desde una consola remota.

Además de estos tres tipos de elementos, se desarrolló una

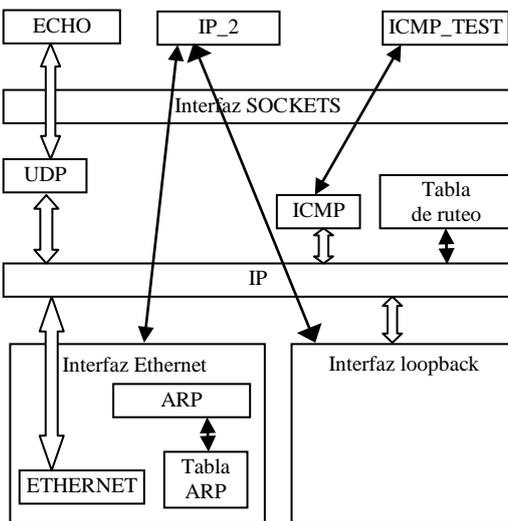


Fig 2. Componentes de Equipo. Las flechas llenas indican el intercambio de frames. Las flechas simples indican interacciones locales

interfaz entre las aplicaciones y los protocolos de nivel transporte, que se corresponde con los sockets. Posibilita que las aplicaciones adquieran ports para la comunicación.

5. Uso de SimIP

El uso de SimIP requiere la definición de tres aspectos en la simulación: la topología y configuración de la red y el tráfico.

Sólo el primer aspecto requiere el conocimiento de OTcl para la definición de equipos y links; se desarrollará una herramienta gráfica para la configuración de la topología para que el usuario no necesite este conocimiento. Los otros dos aspectos pueden especificarse mediante el uso de aplicaciones, lo que permite que los usuarios puedan trabajar con SimIP conociendo sólo el uso de estas aplicaciones, que en su mayor parte son aplicaciones de red reales limitadas en cuanto a su funcionalidad y sintaxis.

```
#Link 1
set L1 [new eth_Link link1 0.1]
$L1 activar_captura

#Equipo 1
set equipo1 [new Equipo PC1]
set lo_1 [new lo_Interface $equipo1 lo]
set eth0_1 [new Eth_Interface $equipo1 eth0 C1:11:11:11:11:10]
set ip1 [new IP_Protocol $equipo1]
set icmp1 [new ICMP_Protocol $equipo1]
set udp1 [new UDP_Protocol UDP $equipo1]
set test1 [new ICMP_test test_icmp $equipo1]
set ip2_1 [new IP2 ip2 $equipo1]
$equipo1 activar_captura

#Equipo 2
set equipo2 [new Equipo PC2]
set lo_2 [new lo_Interface $equipo2 lo]
set eth0_2 [new Eth_Interface $equipo2 eth0 C2:22:22:22:22:20]
set ip2 [new IP_Protocol $equipo2]
- - - Igual a Equipo 1

#Conexion de equipos (interfaces) al link
attach $L1 $eth0_1
attach $L1 $eth0_2
```

Fig. 3 Definición de la topología de la red

5.1 Topología de la red

Consiste en la definición de equipos y líneas, y en la especificación de su interconexión. En el caso normal, este aspecto se define en el archivo de entrada al simulador, antes del arranque de la simulación. Es posible definir parte de la topología en un tiempo arbitrario de la simulación, utilizando ese mismo archivo de entrada. Una alternativa adicional que ofrece SimIP cuando se trabaja en modo interactivo (servidor de consolas) es definir parte de la topología en momentos arbitrarios de la simulación, desde la consola principal; en este caso, a diferencia de los anteriores, la topología de la red no está totalmente definida en el momento de la lectura del archivo de entrada, sino que podría ser alterada por el usuario que tiene acceso a la consola principal, de acuerdo a los resultados que se observen en la simulación. En la fig 3 se muestra la definición de una topología simple consistente en un vínculo broadcast de acceso múltiple con dos equipos conectados. En primer término se define el link y se activa la captura de datos, que hace que se genere el archivo de captura correspondiente. En cada uno de los equipos (PC1 y PC2 — en la figura sólo se muestra PC1—) se define una interfaz loopback (lo) y una interfaz Ethernet (eth0), especificando su dirección de hardware. Luego se define para ambos

```
#Equipo 1
$ip2_1 address add 127.0.0.0/8 dev lo
$ip2_1 address add 200.2.3.1/24 dev eth0
$ip2_1 route add 127.0.0.0/8
$ip2_1 route add 200.2.4.0/24 via 200.2.3.4
$ip2_1 route add 200.2.5.0/24 via 200.2.3.4
$ip2_1 route add 200.2.6.0/24 via 200.2.3.3
$ip2_1 route add 200.2.3.0/24
$ip2_1 route add default via 200.2.3.3
```

Fig. 4 Configuración de la red

equipos, los protocolos y aplicaciones residentes en ellos: el protocolo IP (incluye tablas de ruteo y demás elementos necesarios para su funcionamiento), el protocolo UDP, y dos aplicaciones, ip2 e ICMP_test. En ambos equipos, al igual que en el link, se activa la captura.

5.2 Configuración de la red

Consiste en la asignación de direcciones IP a las interfaces de red y la asignación de rutas. La configuración puede realizarse a continuación de la definición de la topología en el archivo de entrada (fig 4), o utilizando las consolas remotas (fig 5) o consola principal en forma interactiva con la simulación. Es posible reconfigurar los equipos en cualquier momento de la simulación. En las figuras 4 y 5 se muestran diferentes aspectos de la configuración: en la primera se crea la tabla de ruteo, mientras que en la segunda se la consulta y modifica; ambas operaciones se realizan utilizando la aplicación ip2.

Se asignan las direcciones 200.2.3.1 y 200.2.3.2 en la red de prefijo 24 a los equipos PC1 y PC2 respectivamente; luego se definen rutas a redes locales, a redes conocidas de la intranet (no mostradas aquí por razones de espacio) y la ruta por defecto. En la figura 5, se invoca a ip2 para visualizar la tabla de ruteo; luego se elimina una ruta y se comprueba la acción con una nueva invocación.

En las figuras 4 y 5 se puede apreciar la diferencia entre trabajar desde el archivo de entrada de la simulación y desde una consola: en el primer caso debe especificarse el objeto, por ejemplo ip2_1, que hace referencia a la aplicación ip2 en el equipo 1 (PC1); en la consola (fig 5), al estar asociada a un equipo (en este caso PC3), no es necesario especificar de qué equipo se trata, lo cual permite utilizar una sintaxis más parecida a la de un equipo real.

5.3 Generación de tráfico en la red

Permite generar paquetes para comprobar el comportamiento de los protocolos y chequear la configuración de la red. Puede definirse en el

archivo de entrada, especificando los tiempos de simulación en los que se generarán paquetes, o bien a través de las consolas en forma interactiva. Para la generación de paquetes, es conveniente utilizar aplicaciones (no invocar directamente a los métodos de los protocolos), ya que esto se asemeja a un equipo real y no requiere el conocimiento del lenguaje del simulador por parte de los usuarios. Si bien existen

```

Símbolo del sistema - ns.ptcl
Ingrese direccion y puerto del servidor (por defecto 127.0.0.1:9000)
127.0.0.1:9000
Ingrese equipo:
PC3
Consola de PC3 lista.
ip2 route show
ip route show(PC3):
Destination Gateway Genmask IFace
200.2.4.0 200.2.3.4 255.255.255.0 eth0
127.0.0.0 0.0.0.0 255.0.0.0 lo
200.2.5.0 200.2.3.4 255.255.255.0 eth0
200.2.6.0 0.0.0.0 255.255.255.0 eth1
0.0.0.0 200.2.6.3 0.0.0.0 eth1
200.2.3.0 0.0.0.0 255.255.255.0 eth0

ip2 route del 200.2.5.0/24
ip2 route show
ip route show(PC3):
Destination Gateway Genmask IFace
200.2.4.0 200.2.3.4 255.255.255.0 eth0
127.0.0.0 0.0.0.0 255.0.0.0 lo
200.2.6.0 0.0.0.0 255.255.255.0 eth1
0.0.0.0 200.2.6.3 0.0.0.0 eth1
200.2.3.0 0.0.0.0 255.255.255.0 eth0

```

Fig. 5. Ejemplo de una consola remota asociada al equipo PC3. Se observa el uso de la aplicación ip2.

aplicaciones que permiten generar tráfico, deben desarrollarse algunas que permitan por ejemplo, la generación directa de datagrams IP, para que el usuario pueda comprobar entre otras cosas las consecuencias de generar datagrams con valores arbitrarios en sus campos.

6 Conclusiones y extensiones

En este trabajo se presentó SimIP, un simulador de redes IP, orientado exclusivamente a fines didácticos. El objetivo de la simulación se centra en el nivel de red de la arquitectura TCP/IP, incluyendo también el nivel de transporte y en el futuro el nivel de aplicación. Está implementado enteramente en lenguaje OTcl, lo cual lo hace totalmente portable y fácil de extender. Utiliza como intérprete el desarrollado para Ns, lo que posibilita incorporar desarrollos realizados para dicho simulador, de uso muy difundido en el ambiente académico.

Las características salientes de SimIP son la posibilidad de estudiar el comportamiento de los protocolos a través de las capturas de los equipos, la posibilidad de analizar el tráfico generado con herramientas existentes para redes reales, y la realización de prácticas de configuración y chequeo de redes en forma interactiva con la simulación en tiempo real (a través de las consolas).

Hasta el momento se ha desarrollado la parte central del nivel de red (IP) de la arquitectura TCP/IP, poniendo énfasis en aquellos aspectos necesarios para facilitar al alumno la comprensión de los conceptos más importantes de esa arquitectura. Debido al lenguaje utilizado y a la simplicidad del desarrollo, es posible extender la funcionalidad de SimIP en dos aspectos: el agregado de funciones de nivel de red y superiores, y el agregado de funcionalidad al simulador para el mejoramiento de su interfaz o conectividad. Las extensiones previstas, incluidas en el primero de los aspectos, son entre otras protocolos como RIP, DCCP y TCP (simplificado) y mecanismos como Qdisc y ruteo avanzado de Linux. En lo que respecta al segundo de los aspectos, el desarrollo de consolas que reflejen diferentes equipos para realizar prácticas de configuración y la conexión de SimIP con redes reales.

Referencias

- [1] Martin A. Brown "Guide to IP Layer Network Administration with Linux Version 0.4.5", 2007, <http://linux-ip.net/html/>
- [2] J. Postel, "Internet Control Message Protocol RFC-792", September 1981
- [3] J. Postel, "Internet Protocol", RFC-791, September 1981.
- [4] "MIT Object Tcl", <http://otcl-tclcl.sourceforge.net/otcl/>
- [5] J. Postel, "User Datagram Protocol", RFC 768, , August 1980
- [6] "Wireshark", <http://www.wireshark.org/>
- [7] "Network Monitor 3", <http://support.microsoft.com/kb/933741>
- [8] K. Fall, K. Varadhan, Ed., "Network Simulator Manual". January 2003, <http://isi.edu/nsnam/ns/>
- [9] Matthew G. Marsh, "Policy Routing Book", 2001, <http://www.policyrouting.org/>