



**UNIVERSIDAD CATÓLICA DE SANTIAGO DE
GUAYAQUIL**

**TESIS DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO(A) EN TELECOMUNICACIONES CON
MENCIÓN EN GESTIÓN EMPRESARIAL**

TEMA:

**ANÁLISIS COMPARATIVO ENTRE LOS DIFERENTES
SOFTWARE PARA EL DISEÑO Y SIMULACIÓN DE REDES,
PARA VALORAR LA UTILIZACIÓN DE UNO DE ELLOS EN
LA ASIGNATURA DE TELEMÁTICA I.**

ALUMNOS:

Andrea Stefanie Castro Maridueña
Paola Andrea Santos Estupiñán

DIRECTOR

ING. LUIS PINZON BARRIGA

GUAYAQUIL 2012



TESIS DE GRADO

ANÁLISIS COMPARATIVO ENTRE LOS DIFERENTES SOFTWARE PARA EL DISEÑO Y SIMULACIÓN DE REDES, PARA VALORAR LA UTILIZACIÓN DE UNO DE ELLOS EN LA ASIGNATURA DE TELEMÁTICA I.

**Presentada a la Facultad de Educación Técnica para el Desarrollo, Carrera
de Ingeniería en Telecomunicaciones de la Universidad Católica de Santiago
de Guayaquil**

REALIZADO POR:

**Andrea Stefanie Castro Maridueña
Paola Andrea Santos Estupiñán**

**Para dar cumplimiento con uno de los requisitos para optar por el título de:
Ingeniero en Telecomunicaciones con Mención en Gestión Empresarial**

**Ing. Luis Pinzón Barriga
Director de Tesis**

Ing.....

Vocal

**Ing. Manuel Romero Paz
Decano de la Facultad**

Ing.....

Vocal

**Ing. Luis Córdova Rivadeneira
Director de Carrera**



CERTIFICACIÓN

Certifico que el proyecto de grado titulado “ANÁLISIS COMPARATIVO ENTRE LOS DIFERENTES SOFTWARE PARA EL DISEÑO Y SIMULACIÓN DE REDES, PARA VALORAR LA UTILIZACIÓN DE UNO DE ELLOS EN LA ASIGNATURA DE TELEMÁTICA I” desarrollado por Andrea Stefanie Castro Maridueña y Paola Andrea Santos Estupiñán fue realizado, corregido y terminado, razón por la cual está apto para su presentación y sustentación.

Andrea Stefanie Castro Maridueña
Paola Andrea Santos Estupiñán

Ing. Luis Pinzón Barriga
DIRECTOR DE TESIS

Dedicatoria

Dedicatoria

A mi padre, que no alcanzo a ver los resultados de mi esfuerzo, pues partió tempranamente de esta vida y aunque ya no esté entre nosotros sigue vivo en mi pensamiento; fue su estímulo mi impulso para llegar al final; y a mi madre por su paciencia, por su comprensión, por su empeño, por su fuerza, por su amor, por ser tal y como es, uno de los pilares fundamentales en mi vida, que me ha mostrado siempre el camino hacia el éxito.

Paola

Dedicatoria

Dedicatoria

Deseo dedicarles este esfuerzo a mis padres ya que fueron un apoyo incondicional para que este sueño se vuelva realidad, mi papa por su perseverancia y mi mama por su comprensión y amor.

A Dios por haberme guiado hasta la meta y por haberme bendecido con una familia espectacular ya que siempre me alentaron a seguir.

A mis amigos, compañeros y profesores por su confianza.

Andrea

Agradecimiento

Agradecimiento

Son muchas las personas a quienes debería agradecer por haberme impulsado a la consecución de esta nueva meta; Dios porque sus tiempos son perfectos y me dio vida y salud para llegar hoy aquí; mis padres que fueron, son y serán siempre el pilar fundamental en todas las etapas de mi vida, y aunque mi padre ya no se encuentre junto a nosotros es mi ejemplo a seguir de lucha y persistencia, sé que desde donde está ahora sigue iluminando mi camino; mis hermanas, Javier, demás familiares y amigos gracias también por su apoyo incondicional y como no mencionar a mi todos mis profesores ya que gracias a sus conocimientos transmitidos y exigencias, hoy me estoy convirtiendo en una profesional.

Paola

Agradecimiento

Agradecimiento

Deseo agradecer a muchas personas especiales que han logrado que siga adelante y nunca darme por vencida; me han brindado su amistad, apoyo, compañía y ánimo en diferentes etapas de mi vida, sé que algunas están conmigo y otras las llevo en mis recuerdos y corazón, pero ambas tendrán la satisfacción de saber que el objetivo fue logrado.

A Dios por haberme bendecido con una familia que siempre me dará su apoyo, a mis padres que hicieron lo imposible para cumplir este sueño; a mis profesores, amigos y compañeros con los que compartí extensas horas en las aulas.

Todos son parte de este gran éxito y quisiera decirles gracias.

Andrea

Resumen

En el presente trabajo se realiza un análisis comparativo entre los diferentes software existentes para el diseño y simulación de redes, con el objetivo de que los docentes de la asignatura Telemática I, de la carrera de Ingeniería en Telecomunicaciones de la Facultad de Educación Técnica para el Desarrollo de la UCSG, dispongan de un material que les permita estudiar, valorar y escoger el software más adecuado para utilizarlo en las prácticas de laboratorio de su asignatura.

El trabajo se estructura en tres capítulos, conclusiones, recomendaciones y bibliografía.

En el primer capítulo se introducen conceptos importantes sobre simulación de sistemas, tales como su definición, tipos, características fundamentales, ventajas, desventajas entre otras.

En el segundo capítulo se estudian las características, así como ventajas y desventajas de algunos de los simuladores de redes más relevantes que existen actualmente en el mercado del software, haciendo especial énfasis en dos que resultaron de interés durante el proceso de desarrollo e investigación para este trabajo: éstos son el Opnet Modeler y el Packet Tracer.

En el tercer capítulo se realiza una comparación entre el Opnet Modeler y el Packet Tracer, mediante la realización de una actividad práctica con cada uno de estos software, cuyo desarrollo y resultados, permiten finalmente valorar el desempeño de estos simuladores, y analizar detalladamente sus ventajas y desventajas.

Abstract

Presently work is carried out a comparative analysis among the different existent software for the design and simulation of nets, with the objective that the educational of the Telematic subject I, of the career of Engineering in Telecommunications of the Ability of Technical Education for the Development of the UCSG, have a material that allows them to study, to value and to choose the most appropriate software to use it in the practices of laboratory of its subject.

The work is structured in three chapters, conclusions, recommendations and bibliography.

In the first chapter important concepts are introduced on simulation of systems, such as their definition, types, characteristic fundamental, advantages, and disadvantages among others.

In the second chapter the characteristics are studied, as well as advantages and disadvantages of some of the pretenders of more outstanding nets that exist at the moment in the market of the software, making special emphasis in two that you/they were of interest during the development process and investigation for this work: these are the Opnet Modeler and the Packet Tracer.

In the third chapter he/she is carried out a comparison among the Opnet Modeler and the Packet Tracer, by means of the realization of a practical activity with each one of these software whose development and results, they allow finally to value the acting of these pretenders, and to analyze their advantages and disadvantages detailedly.

Índice

Índice

Introducción	1
Problema	2
Objeto	2
Objetivo.....	2
Objetivos específicos	3
Hipótesis.....	3
Técnicas y métodos empleados en la investigación.	3
Capítulo I: Introducción a la simulación.	5
1.1 Reseña histórica de la simulación por computadora.	5
1.2 Definición de simulación y simulación por computadora.	6
1.3 Simulación de sistemas.	7
1.3.1 Estados, eventos y acciones.....	7
1.4 Modelos de simulación.....	8
1.4.1 Tipos de modelos de simulación	9
1.5 Tipos de simulación.....	11
1.5.1 Simulaciones de presentación.....	11
1.5.2 Simulaciones de práctica	12
1.6 Ventajas de la simulación.....	13
1.6.1 Ventajas de la simulación aplicada a la docencia.....	14
1.7 Desventajas de la simulación.....	15
1.8 Metodología para realizar un estudio de simulación.....	16
Capítulo II: Estudio de algunas herramientas software para la simulación de redes.	19
2.1 Introducción a los software de simulación de redes	19
2.2 Clasificación de los software de simulación de redes	21
2.3 Características deseables de un simulador de redes	23
2.4 Principales herramientas de simulación de redes	24
2.5 Opnet Modeler	25
2.5.1 Aplicaciones típicas de Opnet	29
2.5.2 Arquitectura de Opnet	30
2.5.3 Librería estándar de modelos de Opnet	31
2.5.4 Cómo modelar en Opnet.....	35

Índice

2.5.5	Tipos de análisis que pueden realizarse en Opnet.....	36
2.5.6	Sistemas reales y modelos de Opnet	37
2.5.7	Mecanismos de comunicación.....	46
2.5.8	Modelado del tráfico de red.....	47
2.5.9	Tipos salidas de simulación.....	49
2.5.10	Requerimientos del sistema.....	50
2.6	Packet Tracer	51
2.6.1	Espacios de trabajo	53
2.6.2	Modos de Operación	577
2.6.3	Conexiones y enlaces	60
2.6.4	Modelado en Packet Tracer	61
2.6.5	Activity Wizard	64
2.6.6	Multiusuario	65
2.6.7	IPC.....	66
2.6.8	Patrones de diseño de actividades	67
2.6.9	Requerimientos del sistema.....	68
2.7	FLAN (F- Links And Nodes)	69
2.7.1	Características generales	69
2.7.2	Interfaz gráfica de usuario	69
2.7.3	Requerimientos del sistema.....	70
2.7.4	Ventajas	70
2.7.5	Desventajas.....	71
2.8	Comnet III	71
2.8.1	Características generales	72
2.8.2	Interfaz gráfica de usuario	73
2.8.3	Requerimientos del sistema.....	73
2.8.4	Ventajas	74
2.8.5	Desventajas.....	74
2.9	OMNET++	75
2.9.1	Características generales.	75
2.9.2	Interfaz de usuario	76
2.9.3	Ventajas	77
2.9.4	Desventajas.....	77
2.10	NS.....	77

Índice

2.10.1	Características generales	78
2.10.2	Interfaz de usuario	79
2.10.3	Requerimientos del sistema.....	80
2.10.4	Ventajas	80
2.10.5	Desventajas.....	81
2.11	NCTUns.....	81
2.11.1	Características generales	82
2.11.2	Interfaz gráfica de usuario	83
2.11.3	Requerimientos del sistema.....	84
2.11.4	Ventajas	84
2.11.5	Desventajas.....	85
2.12	KIVA	85
2.12.1	Características generales	85
2.12.2	Interfaz gráfica de usuario	86
2.12.3	Requerimientos del sistema.....	87
2.12.4	Ventajas	87
2.12.5	Desventajas.....	88
Capítulo III: Desarrollo de ejercicios prácticos para comparar el desempeño de los software Opnet Modeler y Cisco Packet Tracer.....		89
3.1	Introducción.....	89
3.2	Ejercicio práctico en Opnet Modeler.....	89
3.3	Ejercicio práctico en Packet Tracer	96
3.4	Comparación de los software Opnet Modeler y Packet Tracer...	100
Conclusiones		102
Recomendaciones		103
Bibliografía		104
Anexo 1. Routers modelados en Packet Tracer.....		107
Anexo 2. Switches modelados en Packet Tracer		108
Anexo 3. Dispositivos finales modelados en Packet Tracer.....		109
Anexo 4. Otros dispositivos modelados en Packet Tracer.....		112

Introducción

El diseño de redes de comunicaciones es una actividad fundamental para los estudios relacionados con el campo de las telecomunicaciones o la informática. La creciente presencia de estos sistemas en nuestra sociedad hace imprescindible la formación de profesionales en esta área; para los cuales, un conocimiento adecuado del diseño, la configuración o el mantenimiento de las mismas es un requisito indispensable en su preparación.

Existen ciertas tareas que se realizan durante el diseño de una red. En primer lugar, deben hallarse las tecnologías adecuadas para las necesidades de comunicación que se plantean. Posteriormente, se elegirán los componentes a utilizar, con sus interfaces y conexiones; dando lugar a la topología de la red. Para llevar a cabo este proceso con fiabilidad, son necesarios amplios conocimientos sobre las tecnologías, los componentes y los servicios que se desean proveer. Además, distintos tipos de redes tendrán distintos requisitos, pudiendo existir grandes diferencias. Por ejemplo, siendo ambas redes de comunicaciones, no es lo mismo una red de un gran operador de telefonía, que una red dedicada a compartir documentos. Los requisitos del servicio son completamente distintos, por lo que también lo serán los componentes y tecnologías utilizadas.

Adquirir todos los conocimientos necesarios para realizar este tipo de trabajo es una tarea ardua. Los alumnos que se aproximan a estos conceptos por primera vez pueden encontrarse perdidos.

Es por estas razones que en la asignatura Telemática I, que reciben los alumnos de la carrera de Ingeniería en Telecomunicaciones de la Facultad de Educación Técnica para el Desarrollo de la UCSG, los docentes se han involucrado en la tarea de mejorar la forma de transmitir estos conocimientos a sus alumnos. Para ello, han determinado que sería de gran utilidad que los alumnos pudieran poner en práctica sus conocimientos mediante la utilización de software para el diseño y simulación de redes. El uso de los software de simulación se ha extendido en diversas ramas de la ciencia y ha

Introducción.

experimentado gran aceptación, y particularmente en las universidades, ha demostrado ser de gran efectividad para el aprendizaje.

Actualmente en el mercado del software existen varios software para el diseño y simulación de redes. Teniendo en cuenta la variedad de simuladores existentes, así como las ventajas, limitaciones y especificaciones de cada uno de ellos, se impone ante los docentes la interrogante de cuál software utilizar.

Problema

El problema de este trabajo se origina en la necesidad de los docentes de la asignatura Telemática I, de la carrera de Ingeniería en Telecomunicaciones de la Facultad de Educación Técnica para el Desarrollo de la UCSG, de disponer de un material para estudiar, analizar y comparar los diferentes software para el diseño y simulación de redes que existen actualmente en el mercado, y que cuya información les permita determinar cuál de éstos software es el más adecuado para utilizarlo en las prácticas de laboratorio de su asignatura.

Objeto

Software para el diseño y simulación de redes.

Objetivo

Realizar un estudio, análisis y comparación entre los diferentes software para el diseño y simulación de redes que existen actualmente en el mercado, que le permita a los docentes de la asignatura Telemática I, determinar cuál de éstos software es el más adecuado para utilizarlo en las prácticas de laboratorio de su asignatura.

Objetivos Específicos

- Realizar un estudio de los conceptos importantes relacionados con la simulación de sistemas.
- Realizar un estudio general de los software para el diseño y simulación de redes, sus características, clasificación, ventajas y desventajas, así como sus aplicaciones.
- Realizar una caracterización exhaustiva de los software para el diseño y simulación de redes más relevantes que existen actualmente en el mercado.
- Realizar una comparación que demuestre, las ventajas y desventajas de cada uno de los software estudiados.

Hipótesis

Si se realiza un estudio, análisis y comparación entre los diferentes software para el diseño y simulación de redes que existen actualmente en el mercado, se dispondrá de un material que le permitirá a los docentes de la asignatura Telemática I, evaluar y determinar cuál de éstos software es el más adecuado para utilizarlo en las prácticas de laboratorio de su asignatura.

Técnicas y métodos empleados en la investigación

- Método de observación documental y científica: Se emplea con el objetivo de obtener información y lograr la definición del problema, la hipótesis, así como la elaboración del marco teórico y del informe de resultados.
- Método analítico: Se emplea con objetivo de analizar los elementos de forma separada para ver las relaciones entre ellos.

Introducción.

- Métodos experimentales: Se utiliza con el objetivo de determinar características propias de los elementos utilizados y dar solución al problema planteado.

Capítulo I: Introducción a la simulación.

1.1 Reseña histórica de la simulación por computadora.

La evolución de la simulación por computadora ha ido paralela al desarrollo de la informática. Su origen tuvo lugar durante la Segunda Guerra Mundial cuando dos matemáticos, J. VNeumann y S. Ulam, tenían el reto de resolver un problema complejo relacionado con el comportamiento de los neutrones. Los experimentos tradicionales basados en prueba y error eran muy caros y el problema resultaba demasiado complicado para resolverlo mediante técnicas analíticas. La aproximación que escogieron se basa en la utilización de números aleatorios y distribuciones de probabilidad. El método desarrollado fue llamado "método de Montecarlo". [22] [17]

El uso de la simulación se acrecentó durante la Guerra Fría con el objetivo de resolver problemas de interés militar: trayectorias y dinámicas de satélites artificiales, guiar misiles, etc. Muchos de estos problemas exigían la resolución de sistemas de ecuaciones diferenciales no lineales, para lo cual se utilizaron computadoras analógicas que usaban elementos electrónicos para resolver las operaciones matemáticas de integración, suma, multiplicación, generación de funciones, etc.

A partir de la década de los 60 empezaron a aparecer en el mercado programas de simulación de sistemas de acontecimientos discretos que poco a poco se empezaron a utilizar para resolver problemas de ámbito civil. Los más destacables fueron el GPSS (*General Purpose System Simulator*, Simulador de Sistema de Propósito General) de IBM y el SIMSCRIPT. Los modelos de acontecimientos discretos son muy utilizados en la actualidad para estudiar problemas de fabricación de procesos, logística, transporte, comunicaciones y servicios. [8]

A partir de los años 80, la revolución que se produjo en la informática tuvo un impacto importante en la simulación por computadora, lo que condujo a la generalización de los simuladores en prácticamente todos los ámbitos de la ciencia y la ingeniería, por ejemplo:

- Predicción del tiempo

- Entrenamiento de pilotos

- Diseño y simulación de redes

En la actualidad, las simulaciones por computadora se han convertido en una parte relevante y útil de los modelos matemáticos de muchos sistemas naturales de ciencias como la física, la astrofísica, la química y la biología, así como de sistemas humanos de economía, psicología y ciencias sociales. Además, se emplea en el diseño de nueva tecnología para llegar a comprender mejor su funcionamiento. En los últimos años, su uso se ha extendido al sector de ocio y ha entrado en el ámbito familiar con productos sofisticados de software que utilizan todos los recursos de la computadora tales como gráficos potentes, bases de datos, computación intensiva, etc.

1.2 Definición de simulación y simulación por computadora.

La simulación es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con el objetivo de comprender el comportamiento del sistema o evaluar nuevas estrategias para el funcionamiento del sistema, dentro de los límites impuestos por un cierto criterio o un conjunto de ellos [8].

Una simulación por computadora es entonces un programa informático que se ejecuta en una computadora o una red de computadoras, cuyo objetivo es crear una simulación de un modelo abstracto de un determinado sistema. La simulación por computadora modela sistemas reales o hipotéticos de forma que su funcionamiento puede ser estudiado y pueda predecirse su comportamiento, ya sea por cambio variables o quizás predicciones hechas acerca del comportamiento del sistema.

1.3 Simulación de sistemas.

Como primer punto para modelar un sistema hay que tener claro los límites de éste, es decir, saber cuál es el sistema y cuál es la frontera del sistema. Un sistema se puede definir como un conjunto de objetos que se reúnen de alguna interacción regular o interdependencia hacia el cumplimiento de algún objetivo. Un sistema no está aislado, sino que los cambios externos también influyen en su comportamiento. [1][4]

Existen básicamente dos tipos de sistemas:

- a) **Sistemas discretos:** Se dice que un sistema es discreto cuando las variables de estado cambian en un conjunto discreto de puntos en el tiempo. Por ejemplo, el número de clientes en una cola varía cuando llega un nuevo cliente y cuando un cliente es atendido.
- b) **Sistemas continuos:** Se dice que un sistema es continuo cuando las variables cambian continuamente a través del tiempo. Por ejemplo, el flujo de agua que trae un río.

1.3.1 Estados, eventos y acciones.



Figura 1.1. Estados, eventos y acciones.
Fuente: el autor.

- Estado: Es la condición de un sistema. Está constituido por un conjunto de variables que contienen la información necesaria para describir al sistema en cualquier momento.
- Evento: Es un estímulo que cambia el estado de un sistema.

- Acción: Es la respuesta de un sistema a un evento o estímulo.

1.4 Modelos de simulación.

El factor más importante a tener en cuenta en la realización de proyectos es que el modelado se trata fundamentalmente de equivalencia. En otras palabras, el objetivo es construir un modelo que sea equivalente al sistema real, existente o propuesto. [22]

Sin embargo, ser equivalente es un término subjetivo que debe ser definido con mucha precisión. Claramente, la equivalencia significa que el modelo se comporte en cierto sentido como el sistema real. No obstante, por razones prácticas, los modelos generalmente se limitan a representar solamente algunos aspectos del sistema de interés. Por lo tanto, para lograr la equivalencia del sistema con el proyecto modelado, se debe modelar el sistema con los siguientes objetivos en mente:

- El modelo debe responder cuestiones de interés: El modelo se utiliza para ayudar a estudiar un conjunto de problemas en particular. Se necesita definir esos problemas claramente antes de iniciar un proyecto de modelado. Conocer cuáles cuestiones son importantes permitirá ejercer un buen juicio en la inclusión u omisión de ciertas características en el modelo. Las respuestas que se obtengan del modelo y su utilidad son el punto de referencia final del éxito del trabajo de modelado.
- El modelo debe tener el nivel de exactitud deseado: La exactitud del modelo puede no ser perfecta, pero se necesita tener una noción de si se están haciendo simplificaciones en el modelo hasta un punto en el que las respuestas que el proporcionará no serán útiles. Dependiendo de qué tipo de acciones se tomarán como un resultado del modelo, se debe determinar qué tan conservador debe ser con respecto a las simplificaciones.

- El modelo debe permitir validación: A medida que se diseña el modelo, se debe tener un plan para fomentar la fiabilidad en los resultados que produce.
- El modelo debe adaptarse a un rango necesario de condiciones de operación. Usualmente, el sistema de interés, y por consiguiente el modelo, está sometido a un rango de estímulos diferentes. Para un modelo de red, esto pudiera significar, el crecimiento del tráfico de la aplicación, o nuevos modelos de aplicación. Si se conoce el rango de condiciones que se desea estudiar, se debe estudiar la forma de asegurarse de que el modelo mantendrá su validez en todo ese rango.

Así que la equivalencia está realmente en función de lo que se quiere que el modelo ayude a obtener. A medida que se toma una decisión de modelado, como por ejemplo qué componente utilizar, uno mismo debe responderse si esta decisión podría perturbar la equivalencia que se ha logrado hasta el momento, o si la aumenta. Por último, ¿se puede medir la relación pérdida/ganancia de la equivalencia para determinar si es aceptable? [14]

El punto importante es que los practicantes del buen modelado no tienen necesariamente respuestas precisas para todas las preguntas, pero ellos mantienen los problemas a resolver como cuestión fundamental a medida que toman decisiones de modelado. Esto es algo que debe hacer muy bien, como mismo se hace al considerar varias aproximaciones para representar el sistema como un modelo.

1.4.1 Tipos de modelos de simulación

Existen varios tipos de modelos de simulación, entre los que se encuentran: [22]

a) Modelos de simulación estático

Es la representación de un sistema en un tiempo en particular. En este tipo de modelos, el transcurrir del tiempo no es un cambio que varíe, afecte o se tenga en cuenta en la simulación.

b) Modelos de simulación dinámico

Es la representación de un sistema que va evolucionando con el tiempo. A diferencia del modelo estático, en un modelo dinámico, el transcurrir del tiempo es una variable que varía y afecta el resultado de la simulación.

c) Modelos de simulación determinístico

Es aquel que no contiene variables aleatorias. Este modelo obtiene una respuesta conocida mediante la entrada de una variable determinada.

d) Modelo estocástico

A diferencia del anterior este tipo de modelo si contiene una o más variables aleatorias. Aquí se les da una entrada y mediante un proceso de análisis que realiza el sistema, se genera una salida incierta.

e) Modelo de simulación continua

Son aquellos que se representan prolongadamente en el tiempo. Este tipo de modelo se caracteriza por tener un rango de tiempo predeterminado. Es utilizado cuando el sistema de estudio es considerado de forma individual.

f) Modelo de simulación discreta

Este tipo de modelos representa fenómenos en donde las cantidades varían en cantidades moderadas sobre el tiempo. Representan individualmente cada una de las partes del sistema que se vaya a estudiar, mediante el estudio de un valor establecido.

g) Modelo de simulación lógica

Esta clase de modelo son representados por un conjunto de si (*if*) y entonces (*then*) en una computadora.

1.5 Tipos de simulación.

Las simulaciones pueden clasificarse en cuatro tipos diferentes, las cuales se dividen en dos grupos: las simulaciones de presentación y las simulaciones de práctica. [22]

1.5.1 Simulaciones de presentación

Las simulaciones de presentación son las que enseñan acerca de una situación, en esta categoría se encuentran:

- Simulaciones físicas
- Simulaciones de procesos

a) Simulaciones físicas

En una simulación física, la computadora, el objeto o el fenómeno es representado en la pantalla, ofreciendo a las personas la oportunidad de aprender sobre él, es decir, el propósito de esta tipo de simulación es mostrar a las personas cómo funciona algo. En este tipo de simulación las personas pueden tratar de conocer cuál sería el resultado si cambiaran alguna reacción o movimiento. También pueden hacer n números de intentos que en la realidad no podrían.

Un ejemplo de este tipo de simulación sería un experimento mecánico en el cual el estudiante lanza un objeto. Él puede variar la velocidad, el ángulo y otros parámetros. El estudiante investiga cual sería el resultado si se cambiara la velocidad (más rápida o más lenta) o el ángulo. La ventaja de esta simulación es que el estudiante puede hacer varios intentos para obtener diferentes resultados, algo que no podría realizar en un laboratorio, ya que en éste no puede manejar la velocidad u otros parámetros.

b) Simulaciones de procesos

En una simulación de procesos, generalmente se informa a las personas acerca de un proceso o concepto que no se manifiesta visiblemente. En este tipo de simulación la persona escoge desde el principio los parámetros, y puede cambiarlo cuando lo desee hasta lograr mejor resultado.

1.5.2 Simulaciones de práctica

Las simulaciones de práctica son las que enseñan cómo realizar una situación, en esta categoría se encuentran:

- Simulaciones de procedimientos
- Simulaciones situacionales

a) Simulaciones de procedimientos

En una simulación de procedimientos las personas aprenden una secuencia de acciones que constituyen un procedimiento, es decir, enseñan a las personas cómo hacer algo. Este tipo de simulación frecuentemente requiere la simulación de objetos, la cual permite a las personas conocer e investigar cada paso que se debe seguir para lograr el objetivo de la simulación. Al igual que en los otros puede intentar diferentes pasos para lograr el procedimiento más adecuado o el de mayor conveniencia.

Un ejemplo de este tipo de simulación sería una práctica para enseñar a utilizar una calculadora o un teléfono, diagnosticar el mal funcionamiento de un equipo o también cuando un estudiante debe diagnosticar a un paciente y prescribir el tratamiento más adecuado.

b) Simulaciones situacionales

Una simulación situacional trata con las actitudes y/o comportamientos que debe asumir una persona ante una situación o problemática. A diferencia de las simulaciones de procedimientos, las cuales enseñan una serie de reglas, las simulaciones situacionales permiten a las personas tomar diferentes roles ante una situación y explorar sus efectos. Un

ejemplo de este tipo de simulación sería el diseño de una red de telecomunicaciones o de un circuito eléctrico.

1.6 Ventajas de la simulación.

- La simulación proporciona un método más simple de solución cuando los procedimientos matemáticos son complejos y difíciles.
- Una vez construido el modelo se puede modificar de una manera rápida con el fin de analizar diferentes políticas o escenarios.
- No es necesario interrumpir las operaciones de la compañía.
- La simulación proporciona un control total sobre el tiempo, debido a que un fenómeno se puede acelerar.
- Beneficia el proceso de innovación ya que permite al experimentador observar e interactuar de diversos modos con el sistema.
- Es mucho más sencillo visualizar y comprender los métodos de simulación que los métodos puramente analíticos. Brinda un entendimiento profundo del sistema.
- Da soluciones a problemas "sin" solución analítica.
- Permite analizar el efecto sobre el rendimiento global de un sistema, de pequeños cambios realizados en una o varias de sus componentes.
- Permite la experimentación en condiciones que podrían ser peligrosas o de elevado coste económico en el sistema real.
- El análisis del modelo del sistema puede permitir la sugerencia de posibles mejoras del sistema real, así como detectar las variables más influyentes en el rendimiento del mismo.
- La simulación suele ser utilizada también con una perspectiva pedagógica para ilustrar y facilitar la comprensión de los resultados que se obtienen mediante las técnicas analíticas.
- Contribuye a la reducción del riesgo inherente a la toma de decisiones.

1.6.1 Ventajas de la simulación aplicada a la docencia

El estudio “*Effectiveness of Simulation in a hybrid and online networking course*”, realizado por Brian H. Cameron, de Pennsylvania State University en el 2003, reveló cómo la efectividad de las simulaciones incide directamente en el contexto del online e-learning (aprendizaje electrónico en línea). El profesor Cameron diseñó un experimento alrededor del tema de su especialidad: redes de computadoras, tema tan complejo que tradicionalmente se había impartido sólo mediante la práctica con el apoyo de un instructor, pero que gracias a la evolución de la simulación, las redes de computadoras ya pueden modelarse en una computadora. [8][22]

El objetivo era comparar el desempeño de 85 alumnos de pregrado en un curso basado en la simulación con el simulador NetCracker, frente al mismo curso con contenidos gráficos estáticos representativos con Microsoft Visio, ambos en un contexto de enseñanza en línea. Toda la interacción de los alumnos fue en línea siendo el único encuentro cara a cara las presentaciones de los proyectos por equipos.

El impacto diferencial fue medido a través de exámenes de opciones múltiples, resultados de proyecto, y una encuesta a los alumnos. Las estadísticas mostraron que el grupo que utilizó simulación logró mejores resultados que el grupo que utilizó presentaciones estáticas, mostrando mejor comprensión de conceptos y mejor retención de información.

Los alumnos que usaron simulación reportaron haber invertido más tiempo en las tareas del curso (3.5 horas) en contraste con los otros (2 horas) y además, revelaron que emplearon más tiempo en las tareas asignadas primordialmente porque la simulación les permitía experimentar con diferentes configuraciones de redes y verificar la funcionalidad de los diseños. Varios estudiantes afirmaron que la simulación les había permitido comprender los complejos conceptos de redes, mientras que los estudiantes en el otro grupo informaron que no había forma de verificar si sus diseños de redes funcionarían correctamente.

Por otra parte, la tasa de deserción fue nula en el grupo que utilizó el simulador, mientras que en el grupo de enseñanza estática en línea desertaron varios alumnos con calificaciones inaceptables. Para el investigador Cameron,

sus resultados confirmaron los resultados de otros investigadores en el sentido de que las simulaciones:

- Permiten la aplicación de conocimiento a la solución de problemas.
- Mejoran la transferencia de conocimiento.
- Aumenta la comprensión de conceptos abstractos.
- Aumenta la motivación de los alumnos.

1.7 Desventajas de la simulación

- Los modelos de simulación orientados a obtener resultados de alta precisión suelen requerir una considerable capacidad de recursos computacionales. Esta desventaja se amplifica cuando se simulan escenarios realistas y/o casos de carga forzada. En la práctica puede ocurrir que existan límites en la dimensión de los escenarios y/o la cantidad de elementos involucrados en la simulación, si es que se desean obtener resultados en tiempos de cómputo razonables. Estas limitaciones pueden ser decisivas para determinar la utilidad de un simulador para un proyecto con objetivos específicos.
- La precisión de los resultados de la simulación está determinada por la efectividad del modelo subyacente. Aún para modelos precisos debe tenerse en cuenta que los resultados de la simulación son solamente estimaciones y predicciones sobre el comportamiento real del sistema simulado.
- Las alternativas de diseño y optimización a analizar quedan determinadas por el número de variables y los rangos de valores analizados para ellas. Cuando se analiza un número limitado de variables y/o valores, las pocas opciones que pueden surgir de la simulación, no garantizan que se cumplan los objetivos del diseñador del sistema simulado.

Sin embargo, en el contexto de las redes de telecomunicaciones, la necesidad de contar con simuladores se relaciona con el actual desarrollo vertiginoso de las tecnologías y la competencia creciente entre productos y empresas del sector. Al disponer de una gama de opciones, se hace imprescindible contar con herramientas que permitan su evaluación precisa en tiempos razonables.

1.8 Metodología para realizar un estudio de simulación

Para realizar un estudio de simulación hay que tener o crear un modelo que represente el sistema. Según referencia a Bank, se recomienda seguir las etapas mostradas en la figura 1.2, mostrada a continuación. [2]

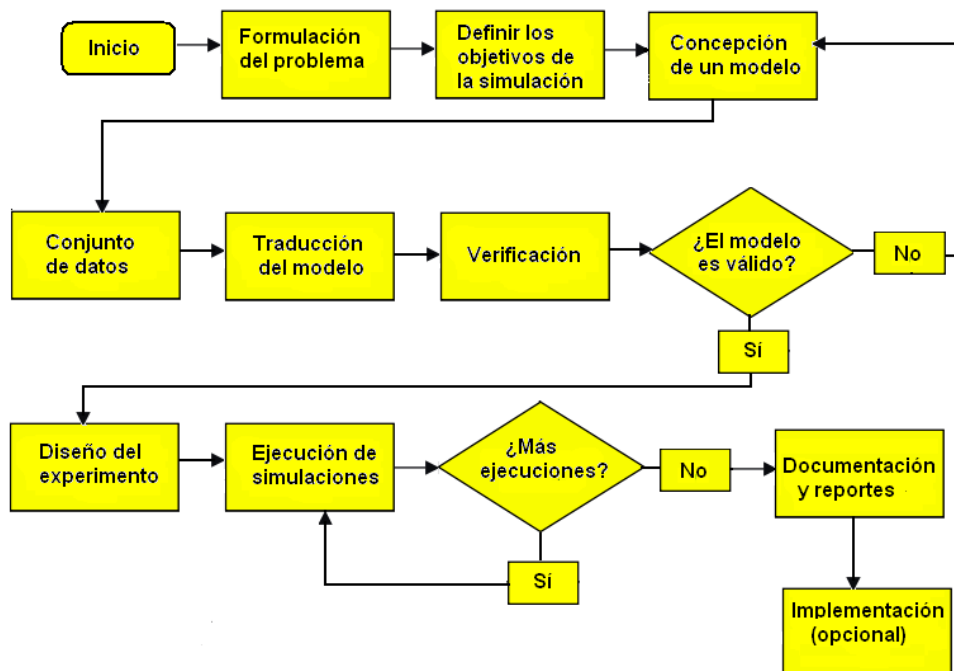


Figura 1.2. Etapas a seguir para realizar un estudio de simulación.

Fuente: el autor.

- 1. Formulación del problema:** Como en cualquier ámbito, para resolver un problema, primero hay que describirlo completamente.
- 2. Definir los objetivos de la simulación y el plan de proyecto:** Los objetivos indican qué es lo que tiene que hacer la simulación, es decir, que respuestas va a entregar. También en esta etapa se decide si la simulación es la herramienta apropiada para resolver el problema, y si

lo es, se debe organizar el proyecto, las validaciones que se van a realizar y cuántas personas y qué tiempo se van a emplear.

- 3. Conceptualización del modelo:** La construcción de un modelo no es sólo una lista de pasos a seguir. No existe una pauta que diga “así se construyen modelos perfectos”. Para modelar se necesita obtener las características esenciales del sistema, y luego, mediante suposiciones y correcciones, mejorar el modelo para que se aproxime al sistema. No es necesario que sea igual que el sistema, sino que sea una esencia del sistema real. La experiencia es la mejor guía en esta etapa.
- 4. Conjunto de datos:** Los datos requeridos dependen totalmente del modelo desarrollado. Mientras el modelo está siendo desarrollado, los conjuntos de datos requeridos también van sufriendo modificaciones. Los datos históricos, de los cuales ya se conoce su comportamiento, sirven para validar el modelo.
- 5. Traducción del modelo:** Esta es la etapa en que se pasa del modelo a la computadora, por medio de un lenguaje de simulación o un software de simulación de propósito específico.
- 6. Verificación:** Se refiere a la verificación del programa en sí, ¿está ejecutándose apropiadamente? Se buscan y corrigen errores de programación y se verifica si la estructura lógica del modelo está correctamente representada.
- 7. Validación:** La validación se refiere a si el modelo es la representación exacta del sistema real. Esto se logra mediante el proceso de comparar el modelo con el sistema y utilizar esas diferencias para ir ajustando el modelo.
- 8. Diseño del experimento:** Se refiere al diseño de lo que se va a simular, y las decisiones que conciernen a cada escenario de simulación, como el tiempo que va a durar, su complejidad, y valores de variables entre otras.
- 9. Ejecuciones de producción y análisis:** Ejecutar las simulaciones y el posterior análisis, así como estimar medidas de desempeño para el sistema que está siendo simulado.
- 10. ¿Más ejecuciones?:** El analista debe determinar si son necesarias más simulaciones y el diseño que éstas deben tener.

11. Documentación y reportes: Se consideran dos tipos de documentos, los de documentación del programa, tanto para si va a ser utilizado otra vez, o va a ser utilizado por otros analistas, o para hacer modificaciones posteriores al programa, además de que brinda mayor confianza que el programa esté documentado. Con los reportes se refiere al progreso del trabajo de simulación, y es donde se informa sobre el trabajo realizado, así como las decisiones tomadas. También se sugiere la realización de entregables periódicos que nos sean necesariamente grandes logros, sino con el fin de mantener informado al personal que no está trabajando directamente en la simulación.

12. Implementación: Una vez obtenidos los análisis de las salidas de simulación, estos deberían ser llevados al sistema estudiado.

Cabe destacar que en ciertos estudios de simulación, no se lleva a cabo el paso de implementación, el estudio finaliza con la etapa de documentación y reportes

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

2.1 Introducción a los software de simulación de redes

El acelerado desarrollo de las técnicas de procesamiento y de las tecnologías para la transmisión de la información, han provocado que las redes de comunicaciones entren en una era de continuas transformaciones que les permiten adaptarse al insostenible avance de la tecnología.

Este desarrollo permanente de las comunicaciones, así como la aparición de servicios de red novedosos y cada vez más complejos, han llevado a los proveedores de tecnología, a las empresas de telecomunicaciones, a las universidades y a los grupos de investigación, a desarrollar principalmente tres técnicas para evaluar el desempeño de las diferentes tecnologías de red. Estas técnicas son: el análisis, la simulación y la experimentación. Sin embargo, considerando la complejidad de las nuevas topologías, los protocolos y el tráfico de las redes, es generalmente imposible construir un modelo analítico que pueda cubrir todos los aspectos técnicos. Por lo tanto, la mayoría de las técnicas analíticas, son utilizadas solamente para la evaluación del desempeño de una red en funcionamiento. Por otro, generalmente resulta costoso construir una red experimental para evaluar su desempeño. [12]

Por estas razones, los programas computacionales que permiten simular diferentes tecnologías de telecomunicaciones, son ampliamente utilizados, ya que facilitan el análisis de las redes y representan además, una reducción en los costos y en el tiempo de diseño y planeación de los nuevos sistemas de comunicaciones.

Por estas razones, ha surgido un área dentro de las telecomunicaciones, dedicada al desarrollo de herramientas computacionales para la simulación de redes; permitiendo evaluar el desempeño de múltiples tecnologías dentro de arquitecturas de red complejas y verificar el comportamiento dinámico de algoritmos, protocolos y aplicaciones; y a su vez, hacer estudios de tráfico y enrutamiento para la detección de posibles fallas antes, durante y después de la implementación física.

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

Con respecto al ambiente educativo, las simulaciones de sistemas telemáticos, constituyen un importante complemento a la conceptualización teórica, especialmente en instituciones que no cuentan con la posibilidad de experimentar sobre redes físicas.

En la actualidad, el desarrollo de la informática unido a la creciente competencia en el mercado del software, han contribuido a que exista una gran variedad de software de simulación de redes en el mercado, cada uno con sus particularidades. Esta diversidad de simuladores existentes permite satisfacer las especificaciones más variadas de los usuarios.

Si bien algunos son específicos para la simulación de un tipo de red en particular, por ejemplo, algunos que son específicos para la simulación de redes inalámbricas, otros están diseñados para cubrir un amplia variedad de configuraciones tales como LAN, MAN, WAN, etc. y tipos de redes, tales como Frame Relay, VPN, MPLS, etc. [15] [17]

La mayoría de estos software disponen de una base de datos con una amplia gama de elementos de red configurables: estaciones de trabajo, servidores, cables, routers, switches, bridges, etc., pudiéndose encontrar generalmente una tecnología de red y fabricante específicos.

En cuanto a los resultados de la simulación, éstos permiten validar el diseño realizado, así como simular el tráfico en la red, la eficiencia, la latencia, el funcionamiento de un protocolo o un servicio en particular, y muchos otros aspectos.

Cualquiera que sea el caso, un usuario podrá encontrar siempre un software de simulación de redes que satisfaga su necesidad, por muy específica que sea, y que conjuntamente, sea compatible con el sistema operativo y el hardware de su ordenador.

En el presente capítulo se realiza un estudio de algunas de las herramientas de simulación de redes que existen actualmente en el mercado del software.

2.2 Clasificación de los software de simulación de redes

Los software para simulación de redes, se pueden clasificar, según el enfoque y las características de la siguiente manera: [5]

- Software específicos de simulación: programas que permiten describir modelos sin programar, cuentan con ambientes totalmente gráficos, y son más fáciles de usar. En contraparte, por ser rígidos, están limitados sólo a algunas aplicaciones. Entre éstos se encuentran OPNET MODELER, Packet Tracer, NCTuns, COMNET, OMNET++, KIVA, FLAN.
- Lenguajes específicos de simulación (LES): como su nombre lo indica, son programas de propósito específico, que permiten controlar el tiempo simulado, (cuál proceso o evento es el siguiente), la generación de números aleatorios y la realización de cálculos estadísticos. Los LES son más versátiles, ya que sus limitaciones son menos, debido a la posibilidad de simular una amplia variedad de tecnologías y modelos de red. Entre los lenguajes de simulación de este tipo más conocidos se encuentran GPSS, SIMULA, SIMSCRIPT, MODSIM, SIM++, PARSEC, JAVASIM.
- Lenguajes de propósito general (LPG): Este tipo de lenguaje no siempre contienen librerías especializadas, lo cual dificulta la implementación de dispositivos de red dentro de procesos de simulación. Esto eleva el tiempo necesario para desarrollar una simulación. En esta categoría se encuentran los lenguajes de programación de alto nivel como C++ y JAVA.
- Simulación de nivel físico: estos programas de simulación contienen generalmente librerías que facilitan el análisis de algunos parámetros de nivel físico de las redes, mediante simulaciones de eventos continuos. Dentro de este grupo se encuentran: LABVIEW, MATLAB, TOPSIM, MathCad, entre otros.

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

En la figura 2.1, mostrada a continuación se observan las principales diferencias de las herramientas software para la simulación de redes, de acuerdo a las características de: tiempo de desarrollo (tiempo invertido en la construcción de un modelo de red), tiempo de ejecución (tiempo durante el cual se debe ejecutar la simulación para obtener los resultados esperados) y grado de portabilidad (característica por la cual un caso de simulación puede transportarse de un sistema operativo a otro sin necesidad de cambiar su código fuente).[5]

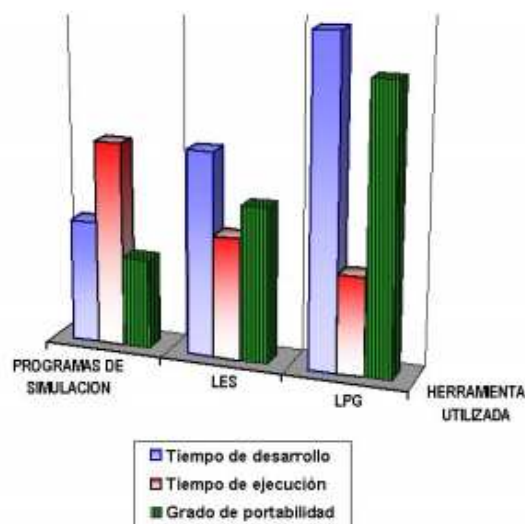


Fig. 2.1. Comparación de las herramientas software para la simulación de redes.

Fuente: [5]

Los software específicos de simulación, tienen el más bajo grado de portabilidad, ya que el proceso de simulación depende del sistema operativo sobre el cual se configuró inicialmente. Sin embargo, su mayor ventaja radica en que éstos ofrecen la posibilidad de construir modelos de red mucho menos tiempo, comparado con los lenguajes específicos de simulación (LES) y los lenguajes de propósito general (LPG). [5]

2.3 Características deseables de un simulador de redes

En el contexto descrito, el simulador debe ser un producto de software que permita simplificar el análisis de la red, integrando herramientas accesorias con alta aplicabilidad y buenas características de usabilidad (interfaz gráfica, herramientas de análisis, protocolos predefinidos, generadores de estadísticas y gráficos para el procesamiento eficiente de resultados, depuradores, etc.).

El software de simulación debe estar modelado de acuerdo a las situaciones y parámetros necesarios para evaluar el comportamiento de la red. En caso de incluir abstracciones o modelos simplificados de la realidad, debe quedar claro su alcance y las limitaciones sobre los resultados de la simulación.

Dentro de las características más deseables de un simulador se encuentra su aplicabilidad, definida como la capacidad de ser utilizado para un amplio espectro de análisis; su escalabilidad, definida como la capacidad de alcanzar resultados para escenarios de grandes dimensiones, eventualmente incorporando resultados de cómputo adicionales; su configurabilidad, que permita adaptar la parametrización al conjunto de valores relevantes para el estudio en cuestión; y su flexibilidad, que queda determinada por la capacidad de otorgar al usuario la posibilidad de incorporar elementos al modelo, diseñar e incluir herramientas de análisis, modificar comportamientos simples de los elementos predefinidos, etc.

En el aspecto concerniente a la aplicabilidad, un software de arquitectura abierta tendría notorias ventajas respecto a un producto cerrado. Un simulador diseñado bajo la filosofía de código abierto puede permitir el desarrollo de protocolos propios o variantes no incluidas en el código provisto. Asimismo brinda la posibilidad de implementar especificaciones particulares que sean de interés para el usuario y/o el proyecto en el cual se utiliza.

No existe un consenso sobre la utilidad de emplear un lenguaje específico o uno de propósito general para codificar el programa que controle la simulación. En general, muchas empresas, centros docentes y algunos investigadores se inclinan por la utilización de un lenguaje específico para la simulación, que posibilita un rápido desarrollo y permite aprovechar al máximo las características del producto

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

software. A modo de ejemplo, es posible mencionar al simulador Opnet, con su entorno y lenguaje de programación específico.

Por otra parte, algunas empresas y números importantes de investigadores, prefieren la flexibilidad y capacidad de abstracción de los lenguajes de propósito general para codificar las simulaciones, ganado claridad y comprensión del código generado. A modo de ejemplo pueden mencionarse todos los módulos de simulación codificados en C++ que integra el simulador de redes NS-2.

El simulador debe contemplar el mayor nivel de generalidad posible para los escenarios donde sea posible aplicar su metodología de análisis. En este sentido, la característica deseable es que las limitaciones que puedan existir para el tamaño de los escenarios, para la cantidad de elementos que participan en la simulación, etc., no impongan restricciones fuertes a la aplicabilidad del software y permitan su aplicación para simular entornos de aplicaciones realistas.

Por último, la ciencia computacional del software constituye un aspecto vital para determinar la utilidad de un producto de simulación. De poca utilidad sería un simulador muy completo y complejo, que permita estudiar, analizar y simular en amplio conjunto de características de la red de telecomunicaciones, si para ello requiere de tiempos de ejecución enormes, o si demanda un consumo de grandes cantidades de recursos computacionales.

2.4 Principales herramientas de simulación de redes

Dentro de las herramientas más utilizadas a nivel académico para el modelado y simulación de redes de comunicaciones se pueden mencionar: Packet Tracer, Opnet Modeler, COMNET III y algunas alternativas de software libre como FLAN, NCTuns, KIVA y OMNET++.

A continuación se realiza un estudio de las principales características de los simuladores de redes mencionados.

Ya que este trabajo pretende acercarnos un poco más a aquellas herramientas que podrían ser útiles en entornos de aprendizaje, durante el proceso de investigación para la realización del mismo, resultaron de particular interés por sus características y prestaciones los software Opnet Modeler y Packet Tracer, por lo

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

que, sin obviar el estudio de los otros software de simulación de redes mencionados, el trabajo se enfocará principalmente en estos dos.

2.5 Opnet Modeler

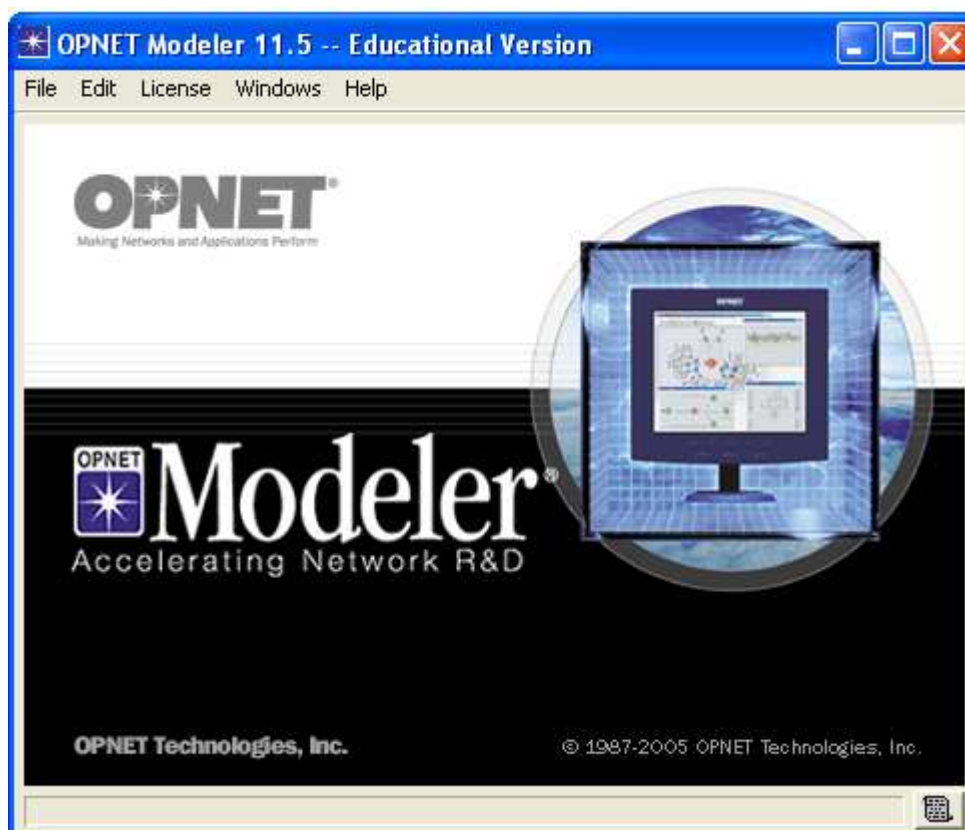


Fig. 2.2. Herramienta de simulación de redes Opnet Modeler.

Fuente: el autor

Opnet Modeler es un simulador basado en eventos orientado a la simulación de redes de telecomunicaciones creado OPNET (Optimized Network Engineering Tools). Para ser más explícitos lo podríamos definir como un simulador dinámico y discreto que puede realizar simulaciones deterministas y/o aleatorias basándose en teorías de las redes de cola.

- Dinámico porque la representación del sistema durante la simulación evoluciona con el tiempo.

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

- Discreto porque el comportamiento de los sistemas representados cambia únicamente en instantes de tiempo concretos, es decir, eventos.

OPNET es uno de los simuladores más avanzados en el campo de las redes de telecomunicaciones. Está diseñado para soportar el modelado de redes en sentido general y para dar soporte a proyectos de simulación de redes de un tipo en particular. Quizás, la característica más relevante es que es un simulador orientado a objetos, lo que permite interactuar al usuario sin problemas y ofrece una gran facilidad de interpretación y creación de escenarios aparte de tener en cada objeto una serie de atributos configurables. Entre sus características más importantes se encuentran [1]:

- Orientado a objeto: Los sistemas especificados en Opnet están constituidos por objetos, cada uno con un conjunto de atributos configurables. Los objetos pertenecen a clases, las cuales los proveen con sus características en término de comportamiento y capacidades. Las clases pueden además derivarse de otras clases, o especializarse para proveer un soporte más específico para una aplicación en particular.
- Especializado en redes de comunicaciones y sistemas de información: OPNET ofrece muchas opciones de modelado relacionadas con las comunicaciones y el procesamiento de la información, proporcionando un gran soporte para el modelado de redes y sistemas distribuidos.
- Modelos jerárquicos: Los modelos del Opnet son jerárquicos, tal como se muestra en la figura 2.3, mostrada a continuación. Naturalmente, en paralelo con la estructura de las redes de comunicaciones reales.

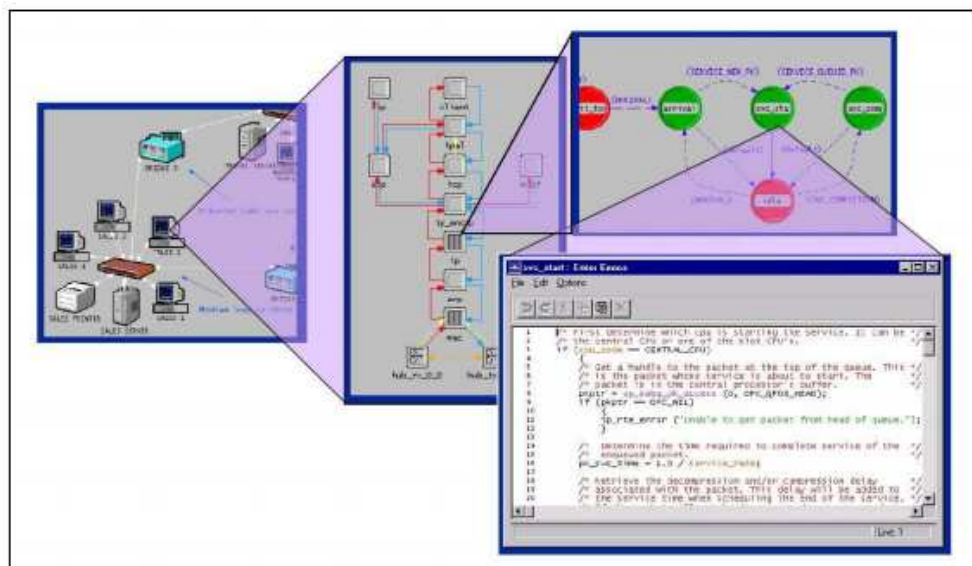


Fig. 2.3. Estructura jerárquica de Opnet.

Fuente: el autor

- Especificación gráfica: Siempre que sea posible, los modelos se introducen mediante editores gráficos. Estos editores proveen un mapeo intuitivo del sistema modelado a las especificaciones del modelo en el Opnet.
- Flexibilidad para desarrollar modelos propios detallados: Opnet provee un lenguaje de programación flexible y de alto nivel con soporte extensivo para comunicaciones y sistemas distribuidos. Este ambiente permite un modelado realista de todos los protocolos de comunicaciones, algoritmos y tecnologías de transmisión.
- Generación automática de simulaciones: Las especificaciones del modelo son compiladas automáticamente en un ejecutable implementado en el lenguaje de programación C. La avanzada construcción de la simulación y las técnicas de configuración minimizan los requerimientos de compilación.
- Estadísticas de aplicación específica: Opnet provee datos de rendimiento pre-construidos que pueden recolectarse automáticamente durante las

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

simulaciones. A estos datos se le pueden unir además, los específicos de la aplicación, que son definidos por el usuario.

- Herramientas post-simulación de análisis integradas: La evaluación de desempeño requiere un gran volumen de resultados de simulación a ser interpretados. Opnet incluye una herramienta sofisticada para la presentación gráfica y procesamiento de la salida de la simulación.
- Análisis interactivo: Todas las simulaciones del Opnet incorporan automáticamente soporte para el análisis mediante un sofisticado depurador interactivo.
- Animación: La ejecución de las simulaciones pueden configurarse para generar automáticamente animaciones del sistema modelado a varios niveles de detalle.
- Cosimulación: Opnet puede conectarse con uno o más simuladores de manera que se pueda ver cómo los modelos en esos simuladores se relacionan con los modelos del Opnet.
- API: Como una alternativa a la especificación gráfica, los modelos de Opnet y los archivos de datos pueden especificarse mediante una interfaz de programa, lo que es muy útil para la generación automática de modelos o para permitir la perfecta integración del Opnet con otras herramientas.

2.5.1 Aplicaciones típicas de Opnet

Opnet puede usarse como una plataforma para desarrollar modelos de un amplio rango de sistemas. Algunos ejemplos de posibles aplicaciones son:

- Modelado del rendimiento de las redes LAN y WAN: Una biblioteca detallada de modelo provee los protocolos principales para las LAN y MAN.
- Planificación de interconexión de redes: las definiciones jerárquicas de la topología permiten que la interconexión de subredes y nodos y las redes de gran tamaño sean modeladas de manera eficiente.
- Investigación y desarrollo en arquitecturas de comunicación y protocolos: Opnet permite la especificación de la lógica totalmente general y proporciona un extensivo soporte para aplicaciones relacionadas con las comunicaciones.
- Las redes de sensor distribuido y control: Opnet permite el desarrollo de modelos sofisticados, adaptativos y del nivel de aplicación, además de protocolos de más bajo nivel y enlaces.
- Dimensionamiento de los recursos: el modelado detallado de las políticas de petición de procesamiento de un recurso es requerido para proveer estimaciones precisas de su rendimiento cuando se somete a una demanda máxima (por ejemplo, la demora de procesamiento en la conmutación de un paquete puede depender de los contenidos específicos y tipo de cada paquete, así como del orden de llegada).
- Redes móviles de paquetes de radio: soporte específico para nodos móviles, incluyendo trayectorias predefinidas o adaptativas, modelos de enlace de radio predefinidos y completamente personalizables,

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

contexto geográfico proporcionado por el entorno de especificación de red de Opnet. (requiere el módulo inalámbrico).

- Redes de satélite: soporte específico para nodos satelitales, incluyendo posicionamiento automático sobre órbitas específicas, un programa utilitario para generación de órbitas y una visualización de órbitas y un programa animado de configuración orbital. (requiere el módulo inalámbrico).
- C3I y redes tácticas: soporte para diversas tecnologías de enlaces, modelado de protocolos adaptativos y algoritmos en Proto-C.

2.5.2 Arquitectura de Opnet

Opnet provee un entorno de desarrollo comprensivo para el modelado y la evaluación del desempeño de las redes de comunicaciones y sistemas distribuidos. El paquete está constituido por un número de herramientas, cada una centrada en un aspecto en particular del proceso de modelado. Estas herramientas se encapsulan en tres principales categorías, que corresponden a las tres fases principales de modelado y simulación de proyectos: Especificación, Colección de datos y Simulación, y Análisis [1].

Estas fases, se llevan a cabo necesariamente en secuencia. Ellas, generalmente forman un ciclo que retorna a la fase de Especificación después del Análisis. Especificación es realmente dividida en dos partes, Especificación inicial y Re-especificación, solo esta última pertenece al ciclo, como se observa en la figura 2.4, mostrada a continuación.

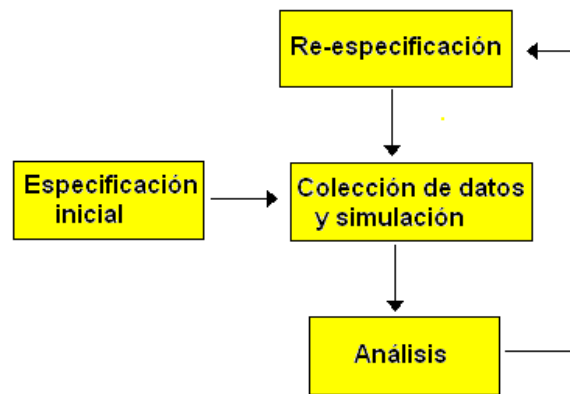


Fig. 2.4 Ciclo de un proyecto de simulación en Opnet.
Fuente: el autor

2.5.3 Librería estándar de modelos de Opnet

Opnet provee una extensa librería estándar de modelos que pueden utilizarse para la simulación de redes. Esta librería es estándar ya que los usuarios pueden desarrollar sus propios modelos.

Está constituida por los siguientes tipos de objetos:

- Dispositivos
- Enlaces
- LANs y nubes
- Objetos utilitarios

a) Dispositivos

Los dispositivos constituyen los objetos principales de la librería estándar de modelos. A esta categoría corresponden una amplia variedad de hardware de red, entre los que encuentran los dispositivos mostrados en la figura 2.5 y muchos otros.



Fig. 2.5 Ejemplos de dispositivos en la librería estándar de modelos de Opnet.
Fuente: el autor

Existen dos tipos de modelos para los dispositivos: modelos de dispositivos de fabricantes y modelos de dispositivos genéricos. Los modelos de dispositivos de fabricantes representan a los dispositivos proporcionados por una compañía o empresa en particular, como por ejemplo Cisco Systems o 3Com. Los desarrolladores de la librería estándar de modelos utilizan los datos publicados por los fabricantes para caracterizar y modelar los dispositivos de la forma más real posible. Los modelos genéricos proveen un comportamiento que es correcto para los dispositivos de su clase, aunque no están configurados para modelar los dispositivos de un fabricante en particular. En cambio, estos dispositivos proveen atributos (parámetros) que pueden configurarse por el usuario. Por ejemplo, el router genérico ofrece la posibilidad de configurar la velocidad de reenvío, la cual especifica el flujo de información por el router expresado en paquetes por segundo. En contraste, el router de un fabricante ya estaría consiente de su propia velocidad de reenvío, como es de esperar, ya que el tipo de dispositivo ya es conocido. Además, el modelo del fabricante tiene una velocidad de reenvío pre-configurada que es consistente con la del router real.

b) Enlaces

En Opnet, los enlaces representan el medio físico que tiene propiedades tales como la velocidad de la línea en bit por segundo, demora o retardo, y la probabilidad de corrupción de los datos. Los modelos de enlaces representan generalmente una alternativa para modelar la tecnología de la capa 2 del

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

modelo OSI (capa de enlace), lo que le permite al Opnet verificar la compatibilidad de dos a más dispositivos conectados, y los enlaces que los unen. Una de las características más importantes del modelo de enlace para la perspectiva de desempeño de una red, es la velocidad de transmisión, en bit por segundo. Esta característica está normalmente implícita en la elección de un modelo de enlace (por ejemplo, un enlace 10BaseT provee automáticamente una velocidad de transmisión de 10 Mb/seg)

Los enlaces son representados como segmentos de una línea o una serie de líneas en la interfaz gráfica de usuario, tal como se muestra en la figura 2.6 mostrada a continuación:



Fig. 2.6. Ejemplos de enlaces en la librería estándar de modelos de Opnet.

Fuente: [1]

c) LANs y nubes

Opnet permite modelar los sistemas finales de la red con explícito detalle, representando cada dispositivo si es necesario. Sin embargo, en muchos estudios de simulación, es preferible abstraerse de la infraestructura de la LAN, y modelarla como un objeto, al cual se le denomina objeto LAN. El objeto LAN modela muchos usuarios en la misma LAN e incluso permite un servidor dentro de la misma LAN. No obstante, al hacerlo todo dentro de un objeto, se reduce drásticamente la cantidad de información que se requiere normalmente para representar la interconexión de la LAN. Como el diseño de la red se reduce a pocos objetos en el modelo, los objetos LAN también reducen la cantidad de memoria requerida para realizar la simulación. En la figura 2.7, mostrada a continuación se observa los objetos LAN que provee Opnet.



Fig. 2.7. Ejemplos de objetos LAN en la librería estándar de modelos de Opnet.
Fuente: [1]

Del mismo modo que con el uso de los objetos LANs, a veces es útil abstraer partes de la infraestructura de la WAN. Los modelos de nubes proveen características de alto nivel, utilizadas para simular el comportamiento de esa porción de la red. En los modelos de ATM, Frame Relay e IP, vienen incluidos los modelos de nubes, como por ejemplo, la nube IP mostrada en la figura 2.8.

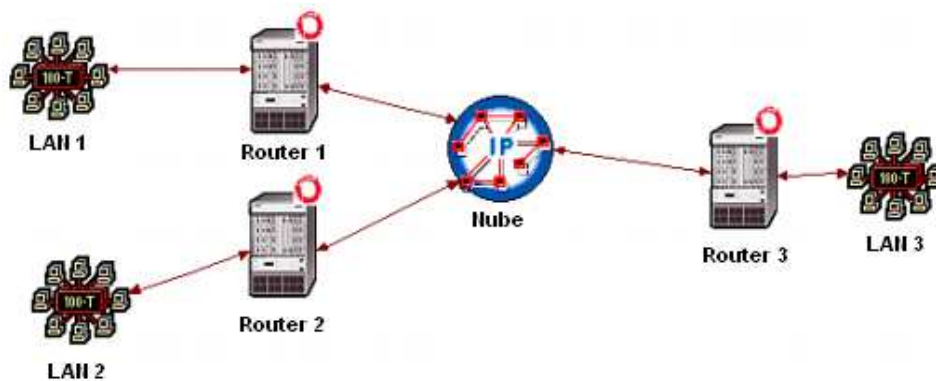


Fig. 2.8. Ejemplos de nubes en la librería estándar de modelos de Opnet.
Fuente: el autor


d) Objetos utilitarios

En Opnet, los objetos que no corresponden a la infraestructura física de la red son utilizados para construir modelos de red. En general, ellos realizan una función lógica en la red, tales como configuración de los recursos de red a nivel global (por ejemplo, la provisión de circuitos virtuales permanentes o PVCs). Otro rol típico para un objeto utilitario es para habilitar una función de simulación específica, como por ejemplo, reportar el uso de recursos de memoria. Finalmente, los objetos utilitarios pueden utilizarse para producir eventos especiales en la simulación, tales como el fallo de un elemento de red en particular, en un tiempo dado.

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

El objeto utilitario Failure-Recovery (Falla-Recuperación) es uno de los más comúnmente usados para introducir fallas en los nodos o enlaces en un tiempo en específico. En un tiempo más tarde, se puede volver a hacer funcional el nodo.

En la fig. 2.9 se muestra un ejemplo en el que se configura la falla en un dispositivo durante un tiempo de 1h y 60 segundos en la simulación.



Name	Time	Status
Office Network.LAN 1	3,600	Fail
Office Network.LAN 1	3,660	Recover

Failure Recovery

Fig. 2.9. Ejemplo del objeto utilitario Failures-Recovery en la librería estándar de modelos de Opnet.
Fuente: [1]

2.5.4 Cómo modelar en Opnet

La mayoría de las decisiones que se toman a la hora de desarrollar un modelo tienen que ver con los aspectos a seleccionar del sistema real para incluir o no incluir. Una aproximación es incluir todo de lo que uno esté consciente, para asegurarse de que no se está omitiendo ningún mecanismo importante. El problema con esta aproximación es que puede convertirse en demasiado tiempo consumido en términos de construcción del modelo y además esto pudiera producir un modelo computacionalmente muy caro.

Elegir los comportamientos a representar en el modelo es una de las tareas más difíciles. Esto tiene sentido pues el trabajo de modelar está supuesto a enseñar cuáles partes del sistema son responsables del comportamiento del sistema de una manera en particular. Por tanto, es difícil conocer por anticipado cuáles aspectos del sistema son significantes. Las decisiones tomadas se realimentan más tarde con la validación. El modelo inicial debería estar basado en una lista de suposiciones si se está investigando un problema en la red, o si se anticipan posibles problemas con una red modificada o futura. En otras palabras, se debe formular una hipótesis sobre qué va a importar en el sistema para responder las preguntas realizadas. Entonces, se puede cambiar la hipótesis y formular muchos otros modelos para enfatizar diferentes aspectos del sistema. Esto ayudará a

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

aprender qué es significativo y qué no con respecto a los objetivos particulares del proyecto.

Cuando se trabaja con un entorno orientado a objeto como Opnet, se tiene una librería de modelos que ayuda a esclarecer las decisiones. Básicamente se eligen aspectos relacionados con:

- ¿Qué objetos debería incluir de la librería?
- ¿Qué protocolos debería habilitar?
- ¿Cómo debería asignar los atributos a los objetos?
- ¿Cómo debería modelar el tráfico en la red?
- ¿Qué tipo de análisis se acomoda mejor a mis necesidades?

2.5.5 Tipos de análisis que pueden realizarse en Opnet

Existen varios tipos de análisis que se pueden realizar con Opnet:

- Simulación discreta de evento: provee los resultados más detallados pero tiene el mayor tiempo de ejecución. Esto se debe a que realiza un análisis más detallado que los otros, maneja tráfico explícito, flujo de datos, y carga de los enlaces. El otro tipo de análisis responde un tipo de preguntas específicas, pero generan resultados más rápidos que la simulación discreta de evento.
- Análisis de flujo: modela los efectos del volumen de tráfico, tipos de tráfico y la configuración de los routers en una red.
- Análisis de impacto de fallas: modela los efectos de fallas de equipamiento en la red. Para realizar análisis de flujo análisis de impacto de fallas se necesita el módulo opcional Análisis de Flujo.

- Validación NetDoctor: realiza un chequeo en la red para ver si está en concordancia con un conjunto de reglas que uno mismo especifica. Se pueden crear reglas personalizadas o utilizar las reglas predefinidas de NetDoctor. NetDoctor identifica problemas y posibles configuraciones erradas en la red y recomienda acciones de corrección. Para utilizar NetDoctor, se necesita añadir el módulo NetDoctor.

Típicamente, se utilizan diferentes tipos de análisis en diferentes puntos del proceso de modelado. Por ejemplo, se podría usar NetDoctor y Análisis de flujo mientras se desarrolla y valida el modelo, luego, se utilizaría Simulación discreta de evento para obtener resultados detallados, y más tarde, se utilizaría NetDoctor para ver escenarios de “qué pasaría si...”.

2.5.6 Sistemas reales y modelos de Opnet

Opnet divide la mayoría de las especificaciones del modelo en un conjunto de cuatro entornos denominados dominios del modelo. Éste difiere de la mayoría de los frameworks (marcos de trabajo) de modelado, los cuales utilizan un paradigma para especificar todos los aspectos del sistema. El contraste entre la aproximación de Opnet y la aproximación de un paradigma es más evidente cuando se consideran ambos problemas de modelado: el comportamiento del sistema y su estructura.

Las redes de comunicaciones y los sistemas distribuidos suelen abarcar un amplio rango de tecnologías que van desde el hardware de comunicación de bajo nivel, hasta el software de toma de decisiones de alto nivel. Un trabajo exitoso de modelado de sistema debe representar cada uno de estos subsistemas y sus interacciones con un nivel de detalle que sea suficiente para obtener predicciones válidas de desempeño y comportamiento.

Como la naturaleza de dichos subsistemas varía significativamente de un nivel a otro (por ejemplo, los protocolos de comunicación y sistemas operativos son entidades diferentes a los enlaces de comunicación), por lo que un framework de un simple paradigma debe ser muy general para desarrollar modelos adecuados.

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

La aproximación de Opnet para el modelado está basada en tres paradigmas que apuntan específicamente a los distintos niveles identificados en una red de telecomunicaciones. Un cuarto paradigma aporta los dispositivos de comunicación necesarios para la cosimulación. Estos paradigmas o dominios del modelo son dominio de red, dominio de nodo, dominio de proceso y dominio de sistema externo, cada uno de los cuales se describe a continuación.

a) Dominio de red (*Network Domain*):

Un modelo de red define el alcance global del sistema que se desea simular. Es una descripción de alto nivel de los objetos contenidos en el sistema. En la figura 2.10 que se muestra a continuación, se observa un ejemplo del modelo de red:

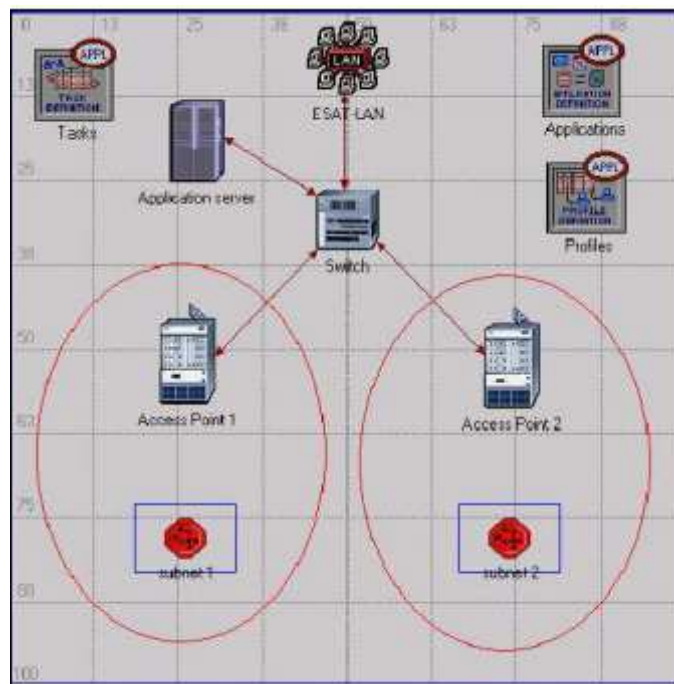


Fig. 2.10. Ejemplo de modelo de red.
Fuente: el autor

El modelo de red especifica los objetos en el sistema, así como sus ubicaciones físicas, interconexiones y configuraciones.

El tamaño y alcance de las redes modeladas pueden ser desde muy simples hasta complejos. Un modelo de red puede contener un nodo, una subred, o muchos nodos y subredes interconectados, porque la estructura y la

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

complejidad de un modelo de red trata de asimilarse lo más posible al sistema modelado. Por ejemplo, una red con una topología en estrella tiene su modelo de red correspondiente con un nodo central (que puede ser un hub) y varios nodos periféricos conectados a éste mediante enlaces punto a punto, tal como se muestra en la figura 2.11.

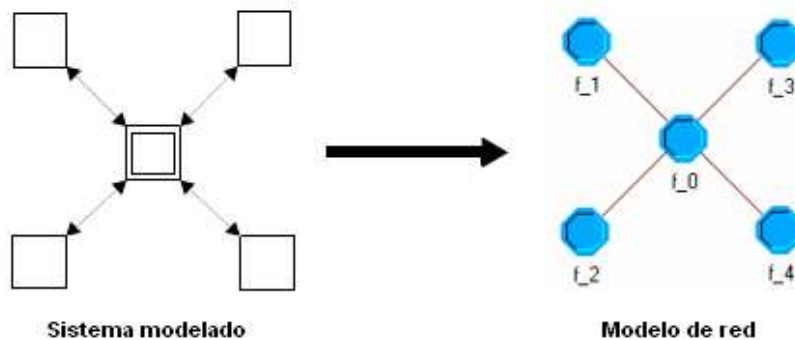


Fig. 2.11. Sistema modelado y modelo de red correspondiente.
Fuente: [1]

El dominio de red está compuesto por los siguientes bloques de construcción principales: subredes, nodos de comunicación y enlaces de comunicación. Estos objetos, ya sea por separado o como un conjunto, pueden denominarse como un sitio. Una subred, encapsula otros objetos del nivel de red. Los nodos de comunicación modelan objetos de redes con estructura interna definida, y los enlaces de comunicación proveen un mecanismo para transportar información entre los nodos de comunicación.

Subredes

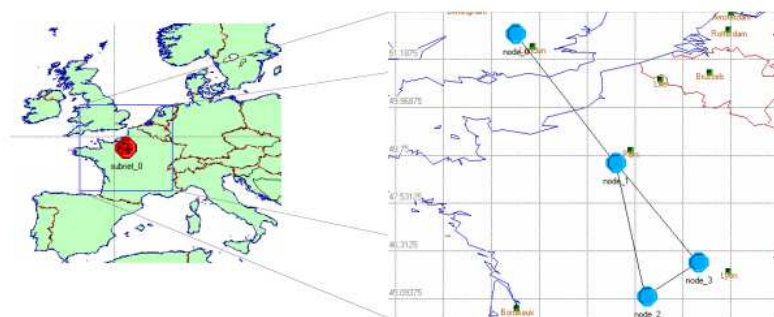


Fig. 2.12. Subredes.
Fuente: [1]

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

Las subredes son esencialmente contenedores que abstraen los componentes específicos de la red, contenidos dentro de ellas, en un solo objeto. Una subred puede abarcar un conjunto de nodos y enlaces, normalmente para representar un grupo físico o lógico de objetos, como por ejemplo, una red de área local. Una subred puede contener además otras subredes, con lo que formarían la jerarquía del modelo de red.

Las subredes proveen un poderoso mecanismo para manipular estructuras de red complejas al eliminar la complejidad de los sistemas mediante la abstracción. Una red inmensa con muchísimos componentes típicamente puede ser segmentada en distintas partes basándose en la proximidad, conectividad y otras consideraciones arquitectónicas de los elementos que la constituyen. Por ejemplo, una universidad puede tener varios campus, cada uno de estos pudiera representarse mediante un objeto de subred. Dentro de cada uno de estos campus, pudieran existir uno o más edificaciones, cada una de las cuales pudiera representarse con un objeto de subred.

Nodos de comunicación

Un nodo de comunicación existe dentro de una subred y representa un dispositivo de red con un amplio rango de capacidades posibles. La función actual y comportamiento de un nodo está determinada por su modelo de nodo, el cual es especificado por su atributo “*node model*”. El modelo de nodo se define en el *Node Editor* y especifica la estructura interna del nodo.

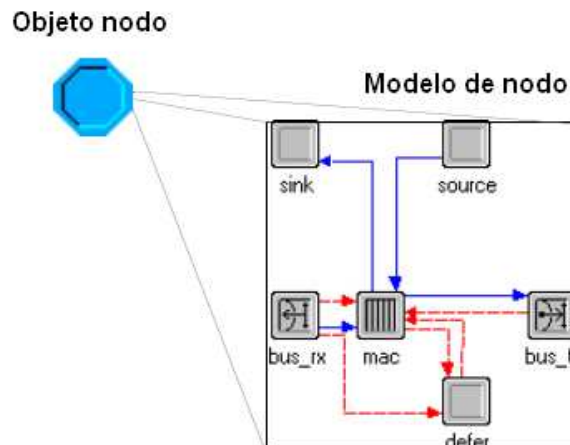


Fig. 2.13. Nodo y Modelo de nodo.
Fuente: [1]

Enlaces de comunicación

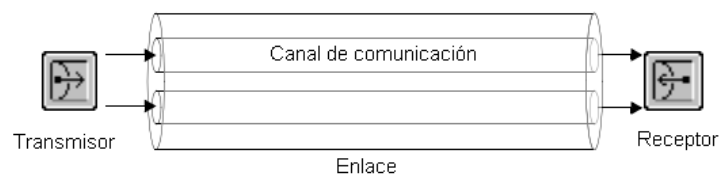


Fig. 2.14. Enlaces de comunicación.
Fuente: [1]

Los enlaces permiten la comunicación entre los nodos mediante la transmisión y recepción de estructuras de datos, conocidas como paquetes.

Opnet soporta dos tipos de enlaces: punto a punto y bus. Cada tipo de enlace provee un tipo de conectividad diferente: los enlaces punto a punto conectan un nodo fuente con un nodo destino, mientras que los enlaces en bus conectan un conjunto de nodos unos con otros. Estos enlaces se representan como objetos en el *Project Editor* del Opnet.

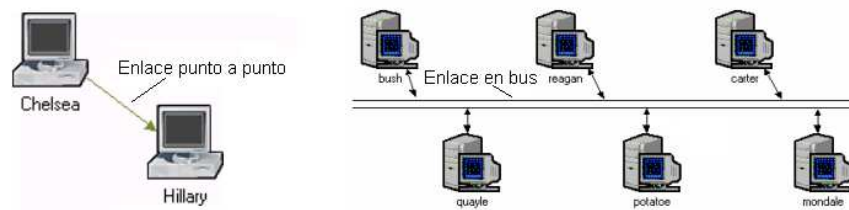


Fig. 2.15. Enlaces punto a punto y en bus.
Fuente: [1]

Mientras que el tipo general de conectividad proporcionado por estos enlaces está predefinido por Opnet, una arquitectura abierta es proporcionada para permitir a los desarrolladores especificar un comportamiento personalizado para cada enlace individualmente. Esta arquitectura se conoce como tubería transceptora porque provee un conducto que conecta un transmisor con uno o más receptores. La funcionalidad básica de una tubería transceptora es determinar cuándo un paquete llega al receptor y si se recibió exitosamente. De esta manera, Opnet provee una arquitectura abierta y modular para implementar el comportamiento del enlace.

b) Dominio de nodo (*Node Domain*):

Los modelos de redes en Opnet se construyen a partir de dos grandes clases de componentes: los nodos de comunicación y los enlaces de comunicación. La mayor parte de la estructura interna de estos objetos, no es visible a nivel de red.

El dominio de nodo provee los métodos para especificar la estructura interna y muchas otras capacidades de los nodos de comunicación. En la siguiente figura se muestra un ejemplo del modelo de nodo:

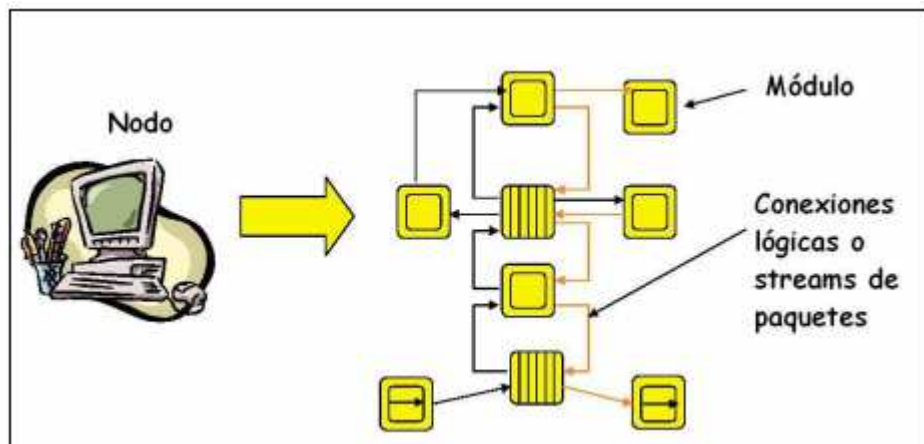


Fig. 2.16. Ejemplo de modelo de nodo.
Fuente: el autor

Tanto la complejidad estructural de los nodos de la red y el alcance de su actividad pueden variar mucho en función del sistema que se modela. En el nivel más bajo de complejidad y función, los terminales simples (conectados a un ordenador central vía conexión en serie) pueden considerarse nodos de una red en estrella. En el nivel más alto de complejidad, un centro de gestión de la red pública de paquetes puede también considerarse como un “*hub*” nodo. Algunos ejemplos típicos de modelos de nodos que pueden construirse con Opnet son:

Estaciones de trabajo (*workstations*): Pueden generar y recibir transferencia de archivos o paquetes muchas conexiones de red concurrentes.

- Conmutador de paquetes (*packet switch*): Soporta un amplio número de enlaces de datos entrantes y salientes y realiza el enrutamiento de paquetes a altas velocidades.
- Terminal satélite: Genera y recibe paquetes de acuerdo al protocolo de acceso al canal. Actúa como un multiplexor para muchos enlaces entrantes.
- Sensor remoto de datos: Actúa como una simple fuente de paquetes. Usualmente los transmite en ráfagas.

Como el objetivo de utilizar Opnet es construir modelos de simulación ejecutables de redes de comunicaciones, las funciones de los nodos internos deben modelarse suficiente bien, para que la simulación del comportamiento de la red sea precisa.

El sistema proporcionado por Opnet es lo suficientemente potente como para especificar prácticamente cualquier forma de nodo de comunicación.

c) Domino de proceso (*Process Domain*):

Los modelos de proceso se utilizan para especificar el comportamiento de los módulos *Processor* y *Queue* que existen en el Dominio de nodo. Los modelos de procesos de Opnet pueden utilizarse para implementar una amplia variedad de subsistemas de hardware y software, incluyendo protocolos de comunicación, algoritmos, recursos compartidos tales como discos o memoria, sistemas operativos, generadores de tráfico especializado, colectores personalizados de estadísticas y mucho más. En la siguiente figura se muestra un ejemplo de modelo de proceso:



Fig. 2.17. Ejemplo de modelo de proceso.
Fuente: el autor

Los objetos de tipo nodo en Opnet se utilizan para representar dispositivos que pueden conectarse en conjunto para formar redes. Los dispositivos pueden variar desde fuentes simples de tráfico, hasta switches complejos o computadoras procesando múltiples aplicaciones. Los nodos se crean como

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

instancias de los modelos de nodo, lo que significa que un modelo de nodo es el patrón para todos los nodos individuales de un tipo en particular. Los modelos de nodo se definen como una colección de módulos que representan funcionalidades distintas. Algunos módulos están limitados en el tipo de comportamiento que pueden representar, por ejemplo, los transmisores y receptores representan interfaces para enlaces definidos en el dominio de red. Tres tipos de módulos, *Processors*, *Queues*, y *Esys (External System)*, soportan modelado total del comportamiento. Estos módulos ofrecen esencialmente las mismas capacidades con respecto a su comportamiento general y la mayoría de los recursos físicos. Sin embargo, dos de ellos proveen capacidades adicionales:

El módulo *Queue* por ejemplo provee soporte especial para el almacenamiento de paquetes organizado, habilitando a los usuarios para definir subcolas internas de cómo los paquetes pueden insertarse, y cuáles paquetes pueden extraerse de acuerdo a un método definido por el usuario.

d) Dominio de sistema externo (*External System Domain*):

Un sistema externo es la representación en Opnet de un modelo cuyo comportamiento está determinado por un código externo al Opnet. Opnet pasa datos al sistema externo y recibe datos de él sin tener conocimiento explícito de cómo el código externo procesó los datos. Para el Opnet, el sistema externo es una caja negra.

Los sistemas externos pueden contener cualquier número de interfaces de sistemas externos, que tanto el Opnet como el código externo pueden acceder. El Editor del Sistema Externo (*External System Editor*) ofrece una manera para construir definiciones de sistemas externos (*ESDs, External System Definitions*), especificando la cantidad de interfaces de sistemas externos, cómo construir la cosimulación, y otros atributos.

Cuando se tiene diseñada la interfaz de sistema externo, se puede salvar como un archivo y entonces asignar esa interfaz a un módulo *esys* utilizando el

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

atributo “esd model” del módulo. El ESD sirve como interfaz entre el Opnet y el sistema externo.

Una cosimulación de Opnet está constituida de un modelo de Opnet, un código de cosimulación y un código externo. El modelo de Opnet contiene uno o más ESDs que especifican el sistema externo a ser utilizado. El código externo puede ser un simulador completo o mecanismos de comunicación (tales como un socket o memoria) para otros procesos. El código de cosimulación une al Opnet con el código externo tal como se muestra en la figura 2.18, mostrada a continuación.

El código de cosimulación utiliza el API *External Simulation Access (ESA, Acceso a Simulación Externa)*.

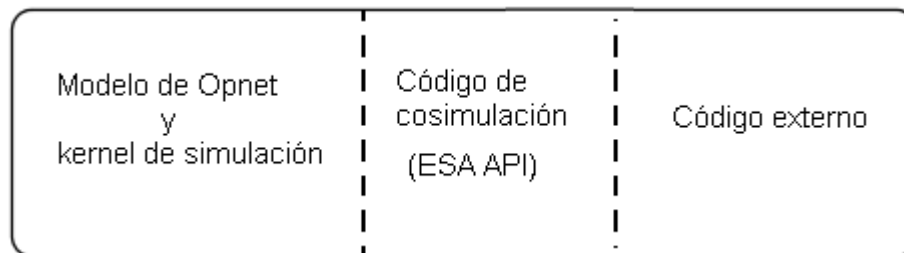


Fig. 2.18. Funcionamiento del código de cosimulación.

Fuente: [1]

Opnet interactúa con los sistemas externos en dos niveles: el nivel de nodo y el nivel de proceso. El posicionamiento de un módulo *esys* en el Editor de Nodo (*Node Editor*) define dónde el sistema externo se ajusta a la estructura del modelo de Opnet. El modelo del proceso del módulo *esys* dirige y transforma de ser necesario los datos entre el sistema externo y el resto del modelo de Opnet.

2.5.7 Mecanismos de comunicación

La mayoría de los modelos de Opnet pueden clasificarse como sistemas distribuidos compuestos por múltiples subsistemas que interactúan unos con otros. Las interacciones de los subsistemas disponen de recursos de comunicación para soportar cambios tales como comandos, consultas e información general. La comunicación pudiera ser requerida entre subsistemas que están físicamente

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

distantes, así como también entre aquellos que están separados lógicamente por sus áreas funcionales. Opnet proporciona varios mecanismos de comunicación para abordar la amplia gama de necesidades requeridas por las aplicaciones, incluso las más sofisticadas.

La forma de comunicación que prevalece en los modelos de Opnet está basada en la noción general de mensaje que puede llevar información y enviarse entre subsistemas. En el entorno de modelado de Opnet, estos mensajes se denominan paquetes, un término tomado del mismo campo de las redes de telecomunicaciones. Sin embargo, los paquetes se pueden utilizar en muchos tipos diferentes de aplicaciones para representar mensajes en general, o transferencia de información.

Los paquetes son estructuras de datos definidas por el entorno de Opnet para dar soporte a la comunicación orientada a mensaje. Cada paquete existe como una estructura individual que puede utilizarse para transferir información entre una entidad y otra. Los paquetes son tratados como objetos, los cuales pueden ser creados, modificados, examinados, copiados, enviados, recibidos y destruidos.

Los paquetes raramente se emplean como elementos de almacenamiento estático que se mantienen dentro de un contexto físico o lógico. Por el contrario, casi siempre se crean de forma dinámica para permitir la transferencia de información entre contextos. El kernel de simulación de Opnet ofrece varios mecanismos para soportar la comunicación basada en paquetes. Uno o más de estos mecanismos puede ser apropiado, dependiendo de la relación entre el contexto que inicia la transferencia y el contexto que recibe el paquete.

2.5.8 Modelado del tráfico de red

Una vez que se crea una topología de red en Opnet, ya sea manualmente o importada, el próximo paso es añadir tráfico a la red. Se puede crear tráfico en la red manualmente mediante la configuración de atributos en varios objetos de la red o mediante la importación de tráfico de archivos externos o programas.

En Opnet, se pueden modelar los siguientes tipos de tráfico: Tráfico explícito (Explicit Traffic) y Tráfico de fondo (Background Traffic)

a) Tráfico explícito (*Explicit Traffic*)

El “tráfico explícito” es el tráfico paquete por paquete, donde la simulación modela cada evento relacionado con el paquete (paquete creado, paquete en cola, paquete transmitido, etc.) que ocurre durante la simulación. El modelado del tráfico explícito provee los resultados más exactos pues éste modela todos los efectos de los protocolos. Por otra parte, esto trae como consecuencia simulaciones más largas y un uso elevado de memoria (porque la simulación asigna memoria a cada paquete individualmente).

Existen tres métodos generales para modelar el tráfico explícito en Opnet:

- Generación de paquetes: se pueden configurar ciertos nodos para generar un flujo de paquetes genéricos. Este es el método básico para añadir tráfico a una topología de red.
- Demandas de aplicación: se pueden crear demandas de aplicación para el flujo de datos entre dos nodos. El tráfico generado por la demanda de la aplicación puede ser puramente discreto (explícito), puramente analítico (de fondo), o una combinación de ellos (híbrido).
- Modelos de tráfico de aplicación: Opnet Modeler incluye un conjunto de modelos para generar tráfico basado en aplicaciones estándar tales como FTP, HTTP, voz y correo electrónico.

b) Tráfico de fondo (*Background traffic*)

El tráfico de fondo es un tráfico modelado analíticamente que afecta el desempeño del tráfico explícito mediante la inserción de demoras adicionales. A diferencia del tráfico explícito, el tráfico de fondo no sólo puede afectar las simulaciones discretas de evento, sino también el análisis de flujo. Las simulaciones discretas de evento que incluyen tráfico de fondo utilizan el modelo de simulación híbrida. Este modelo incluye los efectos del tráfico de fondo para calcular las colas que se generan en los dispositivos intermedios y las demoras basadas en la longitud de las colas, en cualquier momento durante

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

la simulación. Al utilizar tráfico de fondo se aumenta la velocidad de simulación considerablemente, ya que cada paquete que produce tráfico no está modelado explícitamente.

Los tráficos de fondo toman tres formas en el Opnet Modeler:

- Flujos de tráfico: describe un flujo de tráfico desde una fuente hacia uno o más nodos destinos. Se pueden crear flujos de tráfico manualmente, utilizando objetos de flujo de tráfico. Además, se pueden importar flujos de tráfico de archivos externos.
- Cargas de nivel básico: este tipo de tráfico representa el tráfico como una carga de fondo en objeto nodo o enlace. A diferencia del flujo de tráfico, el cual puede extenderse a múltiples enlaces y nodos, una carga de tráfico es estática y se aplica a un objeto.
- Demandas de aplicación: Se pueden utilizar las demandas de aplicación para representar el tráfico de fondo que fluye entre dos nodos. Además del tráfico de fondo, las demandas de aplicación se pueden configurar para ser puramente de tráfico explícito, o una combinación de ambos (*hybrid*).

2.5.9 Tipos salidas de simulación

Las simulaciones de Opnet pueden generar una variedad de tipos de salidas que son aplicables en diferentes situaciones. Dado que las simulaciones de Opnet pueden contener código definido por el usuario, es claramente posible definir nuevos tipos de salidas de simulación que pueden reportarse para post-procesamiento durante o después de las simulaciones.

Las formas de simulación que provee Opnet son:

- Vectores de salida: el historial de un sistema variable puede capturarse como en forma de vectores de salida (*output vector, OV*). Los vectores de

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

salida proporciona una forma de penetrar en la manera en que el sistema se desenvuelve y su respuesta a incidentes particulares durante una simulación. Cada vector de salida está constituido por una serie de valores y tiempos asociados. Una simulación puede configurarse para reunir simultáneamente múltiples vectores de salida independientes.

- Escalares de salida: Ciertos datos estadísticos de interés no varían con el tiempo. En cambio, no son un valor que es típico del desempeño del sistema sobre el curso de la simulación. (por ejemplo, la eficiencia de la red). Cada uno de estos datos estadísticos se denominan escalares de salida (*output scalars, OS*). Cada escalar de salida se graba solamente una vez por simulación.
- Animación: la visualización de la actividad del sistema puede ser una herramienta valiosa para ganar comprensión acerca del comportamiento y las interacciones entre componentes. Las simulaciones pueden generar animaciones mientras se están ejecutando o pueden salvarlas para reproducirlas posteriormente una vez que se haya completado su ejecución.
- Reportes propios y archivos: Como Opnet permite la inclusión de procesos y modelos de enlaces definidos por los usuarios, es posible generar reportes personalizados o archivos de salida al correr la simulación, o cuando se complete. Las facilidades del lenguaje C y del Sistema Operativo se utilizan para este propósito.

2.5.10 Requerimientos del sistema

Mínimos:

- Sistema Operativo: Windows NT, 2000, XP, Unix
- CPU: Intel Pentium III 500MHz
- RAM: 64 MB libres

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

- Almacenamiento: 100 MB de espacio libre en disco
- Resolución del display: SVGA
- 8 MB en memoria de video y tarjeta de sonido.

2.6 Packet Tracer

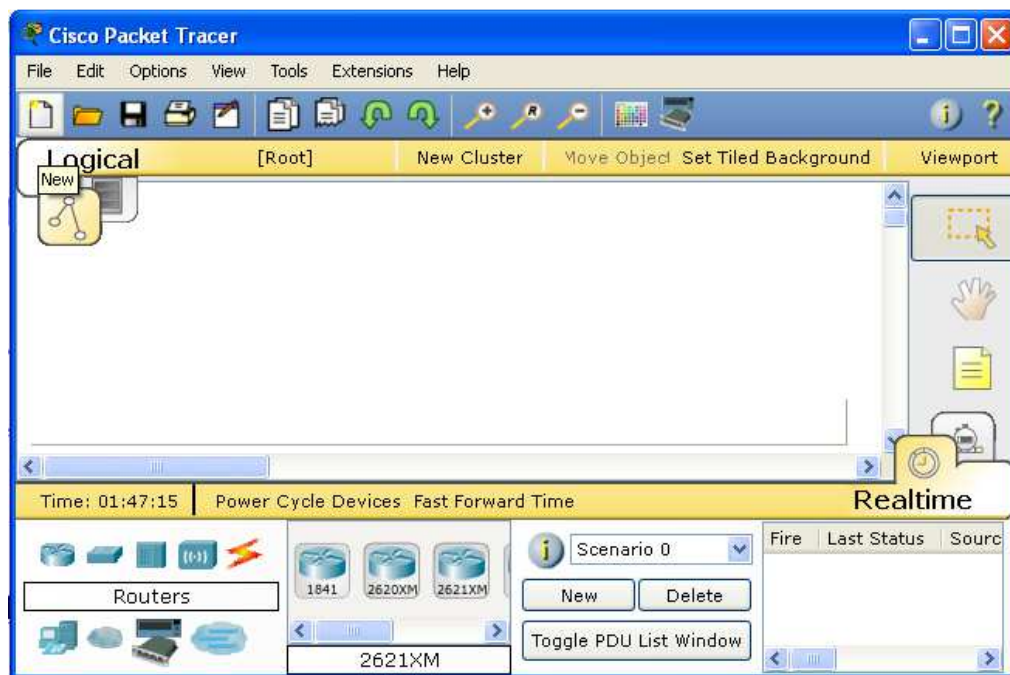


Fig. 2.19. Herramienta de simulación de redes Packet Tracer.
Fuente: el autor

Packet Tracer es una herramienta de simulación, visualización, colaboración y evaluación para la enseñanza de redes. Packet Tracer permite a los estudiantes construir sus modelos de red propios, obtener acceso a representaciones gráficas importantes de esas redes, animar el funcionamiento de la red mediante la inserción de paquetes de datos, responderse preguntas sobre esas redes, y finalmente, anotar y salvar sus diseños. El término “*packet tracing*” describe un modo animado donde el estudiante puede pasear a través de los eventos de red simulados, uno a la vez, para investigar la microgénesis del complejo fenómeno de *internetworking*, que normalmente ocurre en un periodo de miles de millones de eventos por segundo.

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

Un evento típico instructivo pudiera comenzar con un instructor planteando un problema de redes. Los estudiantes pueden utilizar Packet Tracer para arrastrar y soltar dispositivos tales como routers, switches y estaciones de trabajo dentro de un espacio lógico de topología (el workspace lógico). Ellos pueden especificar el tipo de interconexiones entre los dispositivos (enlaces) y configurar los dispositivos que ellos crearon. Una vez que han configurado una red de nodos y enlaces, ellos pueden lanzar paquetes de datos de muestra en la red, incluso en tiempo real, o en un modo de simulación controlada por el usuario. Los paquetes son mostrados gráficamente. Los estudiantes pueden seguir el paquete a través de la red y examinar las decisiones de procesamiento tomadas por los dispositivos de la red, cómo conmutan y enrutan el paquete hacia su destino.

Las redes, escenarios de paquetes y animaciones resultantes pueden ser anotadas, salvadas y compartidas. Muchas representaciones importantes para el conocimiento del dominio de redes están disponibles para que los estudiantes puedan desarrollar diversos modos de investigación. De particular interés para los instructores puede ser el Activity Wizard, el cual permite la creación de redes de respuestas para que los estudiantes puedan comparar su progreso. Además, de posible interés para los instructores es la función de multi-usuario de Packet Tracer, donde diferentes instancias de Packet Tracer pueden utilizarse para crear un Internet virtual, sobre una red real.

Packet Tracer se basa en tres principios de aprendizaje: el aprendizaje es activo, el aprendizaje es social y el aprendizaje es contextual. Por lo tanto, está destinado a facilitar la creación de materiales de enseñanza atractivos, colaborativos y localizados. Packet Tracer puede utilizarse de diversas maneras:

- Trabajo en grupo
- Trabajo en clase, tareas y educación a distancia.
- Evaluación formativa
- Refuerzo del laboratorio
- Clases magistrales
- Modelado y visualización de los algoritmos de dispositivos de red y protocolos de red.

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

- Estudio de caso.
- Actividades multiusuario cooperativas y competitivas.
- Competencias
- Actividades de resolución de problemas.
- Actividades de resolución de problemas en relación con la fomentación de conceptos, el desarrollo de habilidades, desafíos de diseño y manejo de problemas.

Packet Tracer permite la actividad de creación de aproximadamente el 80% de los temas y habilidades requeridas para la certificación CCNA, y tiene relevancia para CCNA-Security, CCNP, IT Essentials y cursos de TCP/IP además. Aunque el programa incluye algunas actividades, los estudiantes pueden crear sus propios ejemplos y compartirlos. Además, los cursos de Discovery y Exploration de CCNA vienen con actividades de Packet Tracer ya integradas.

2.6.1 Espacios de trabajo

Packet Tracer utiliza dos esquemas de representación para una red: el Espacio de Trabajo Lógico (*Logical Workspace*) y el Espacio de Trabajo Físico (*Physical Workspace*), los cuales se muestran en las figuras 2.20 y 2.21 respectivamente, mostradas a continuación.

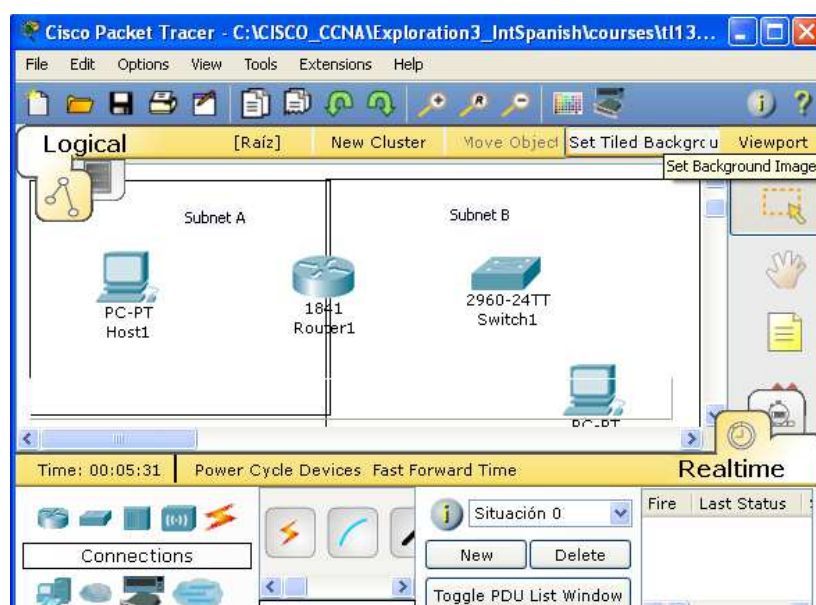


Fig. 2.20. Espacio de Trabajo Lógico (*Logical Workspace*).
Fuente: el autor

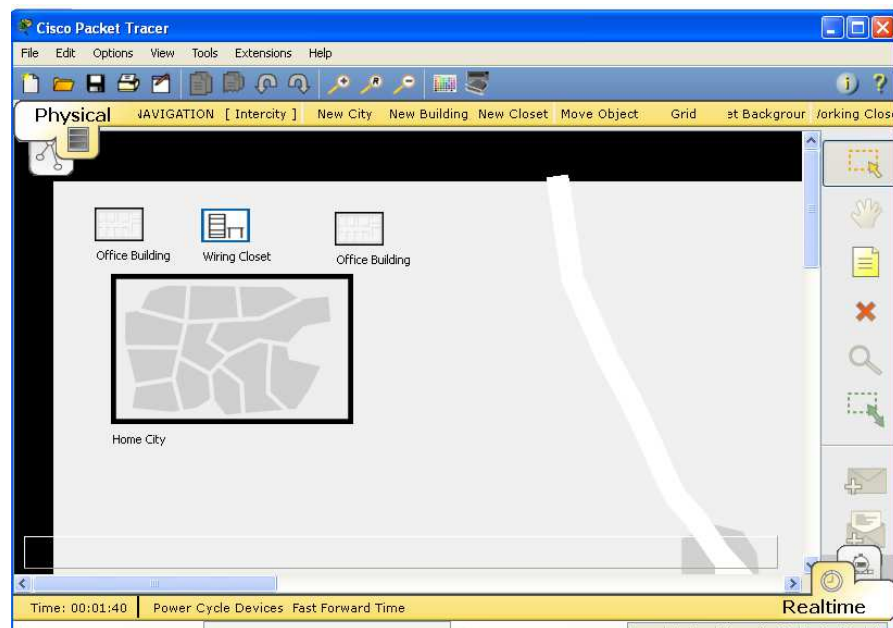


Fig. 2.21. Espacio de Trabajo Físico (*Physical Workspace*).
Fuente: el autor

El Espacio de Trabajo Lógico permite construir una topología lógica de red, sin tener en cuenta su dimensión y disposición físicas. El Espacio de Trabajo Físico permite ubicar dispositivos físicamente en ciudades, edificios y closets de comunicaciones. Si las conexiones inalámbricas se utilizan las distancias y otras magnitudes físicas afectarán el desempeño y otras características de la red. En Packet Tracer se construye primero la red lógica y luego se ubica la red en el Espacio de Trabajo Físico.

a) Espacio de Trabajo Lógico

En el Espacio de Trabajo Lógico es donde se trabaja la mayor parte del tiempo. Generalmente lo primero que se hace es crear dispositivos. Luego se puede hacer cualquiera de las siguientes tareas:

- Añadir módulos a los dispositivos para instalar interfaces adicionales.
- Conectar los dispositivos mediante la elección de los cables apropiados.
- Configurar los parámetros de los dispositivos.

- Realizar configuraciones avanzadas.

b) Espacio de Trabajo Físico

El propósito del Espacio de Trabajo Físico es asignar una dimensión física a la topología de red lógica. Esto da un sentido de dimensión y ubicación (cómo podría quedar la red en un entorno real).

El Espacio de Trabajo Físico se divide en cuatro capas para reflejar la dimensión física de cuatro entornos: Interciudad (*Intercity*), Ciudad (*City*), Edificio (*Building*) y closet de cableado (*Wiring Closet*). Interciudad es el entorno más grande. Este puede contener muchas ciudades. Cada ciudad puede contener muchos edificios. Por último, cada edificio puede contener muchos closets de cableado.

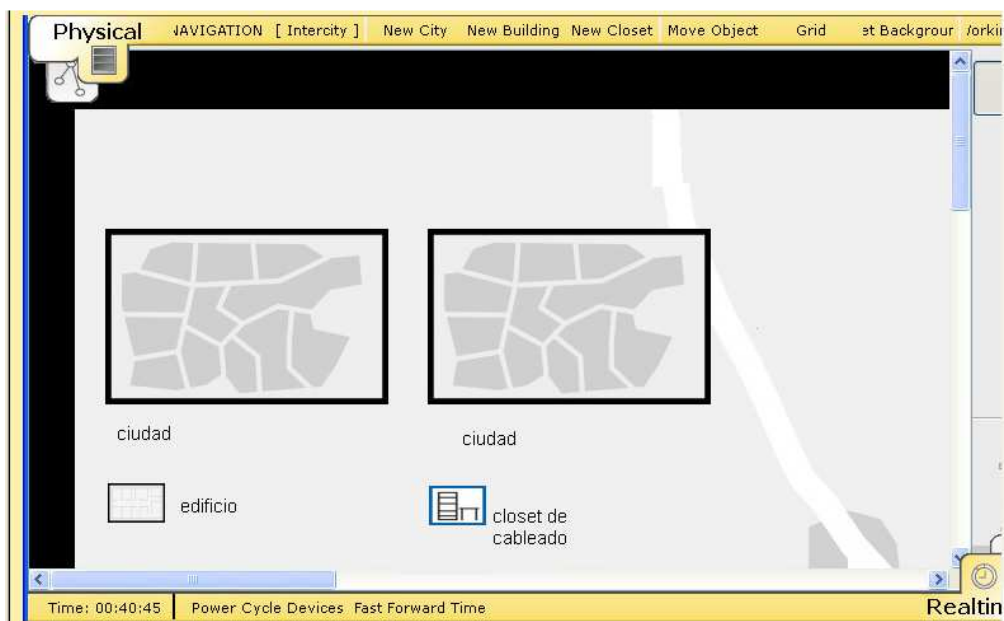


Fig. 2.22. Entornos Físicos: interciudad, ciudad, edificio y closet de cableado.
Fuente: el autor

El closet de cableado proporciona una vista que es diferente a las otras tres vistas. Es en esta vista donde se ven realmente los dispositivos que se crearon en el Espacio de Trabajo Lógico, colocados en racks, tal como se muestra en la figura 2.23, que aparece a continuación.

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

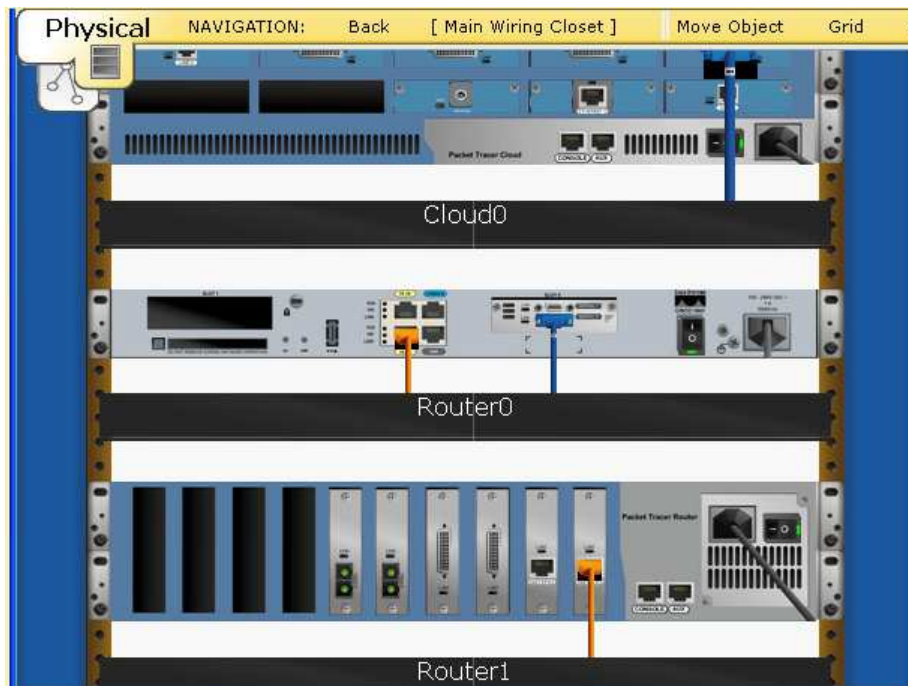


Fig. 2.23. Vista de los dispositivos ubicados en un closet de cableado.
Fuente: [2]

El Espacio de Trabajo Físico proporciona la dimensión de distancia para los dispositivos Ethernet e inalámbricos. Este parámetro de distancia es uno de los factores para determinar si un dispositivo está habilitado para conectarse o no a otro dispositivo.

Los puntos de acceso pueden establecer conexiones con dispositivos finales inalámbricos que están dentro de cierto rango de distancia. Este rango se indica por una malla gris alrededor del punto de acceso. En la figura 2.24, mostrada a continuación, se observa un ejemplo.

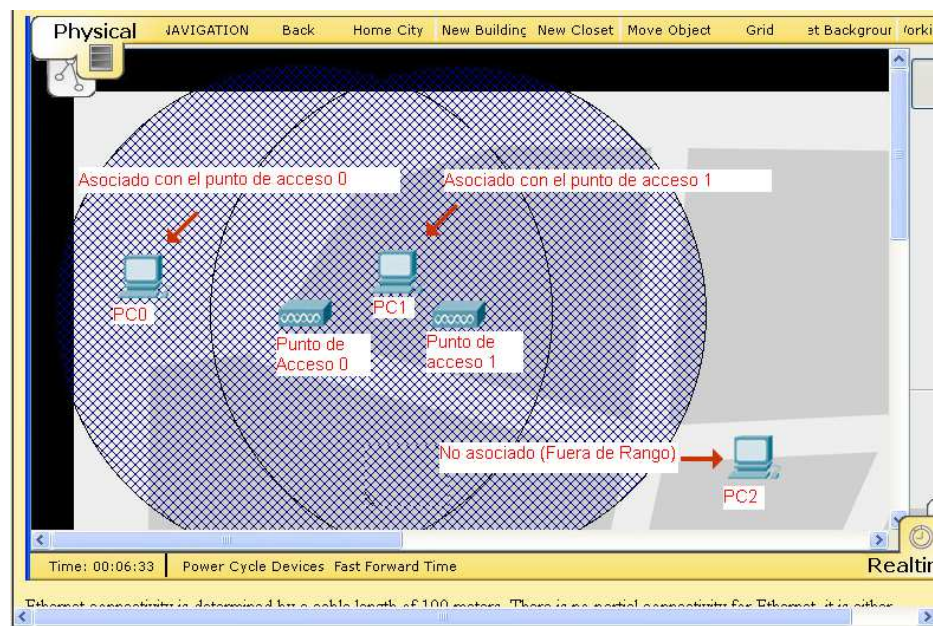


Fig. 2.24 Ejemplo del alcance de los puntos de acceso.
Fuente: el autor

En este ejemplo se crean tres PCs con tecnología inalámbrica y dos puntos de acceso. Notar lo siguiente:

- PC0 está dentro del rango inalámbrico del Punto de Acceso 0, así que se asocia con el Punto de Acceso 0.
- PC1 está dentro del rango de ambos puntos de acceso, pero por cercanía al Punto de Acceso 1 se asocia con el Punto de Acceso 1.
- PC2 no está en el rango de ningún punto de acceso, así que no tiene conectividad.

2.6.2 Modos de Operación

Los modos de operación de Packet Tracer reflejan el esquema de tiempo de la red. Éstos son: Modo tiempo real y Modo simulación.

a) Modo tiempo real

En modo tiempo real, la red ejecuta su funcionamiento en un modelo de tiempo real, dentro de los límites de los modelos de protocolos usados. La

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

red responde a las acciones inmediatamente como lo haría en un dispositivo real. Por ejemplo, tan pronto se haga una conexión a Ethernet, aparecerán las luces del enlace indicando el estado de conexión. Cada vez que se escriba un comando en la línea de comandos (por ejemplo un **ping** o **show**), el resultado de la respuesta se genera en tiempo real y se ve como tal. Toda la actividad de la red, particularmente el flujo de paquetes de datos a través de la red, ocurre en el modelo de Packet Tracer de tiempo real.

b) Modo simulación

En modo simulación, se puede “congelar” tiempo. El usuario tiene control directo sobre el tiempo en relación al flujo de los paquetes. Se puede ver el funcionamiento de la red paso a paso, o evento por evento, de la forma que uno prefiera, rápido o lento. Se pueden establecer escenarios, como el envío de un ping de un dispositivo a otro. Sin embargo, nada se ejecuta hasta que el usuario lo captura (la primera vez, como con un husmeador (*sniffer*) de protocolo) o lo reproduce (reproducir los eventos capturados como una animación). Cuando se captura o reproduce la simulación, se observan representaciones gráficas de paquetes viajando de un dispositivo a otro. La reproducción se puede detener, ir hacia adelante y hacia atrás en el tiempo e investigar muchos tipos de información de paquetes y dispositivos específicos en tiempos específicos. Sin embargo, otros aspectos de la red seguirán funcionando en tiempo real. Por ejemplo, si se apaga un puerto, la luz de este enlace responderá inmediatamente indicando el estado de apagado.

En el modo simulación, se pueden simular situaciones de redes complejas (escenarios). Cuando se cambia al modo simulación, el escenario por defecto es el 0. Un escenario es un conjunto de paquetes que se colocan en la red para ser enviados en un tiempo específico. Para una misma

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

topología lógica se pueden crear múltiples escenarios correspondientes a diferentes condiciones de prueba.

Packet Tracer permite además, el envío de paquetes personalizados. Se puede el puerto por el cual va a enviarse el paquete, o puede dejarse por defecto. Se puede cambiar el tipo de paquete seleccionando de una lista de aplicaciones. Dependiendo del tipo de aplicación y del dispositivo, los paquetes pueden tener los siguientes atributos: **Destination IP Address**, **Source IP Address**, **TTL (Time-To-Live)**, **TOS (Type of Service)**, **Source Port**, **Destination Port**, **Sequence Number**, y **Size**.

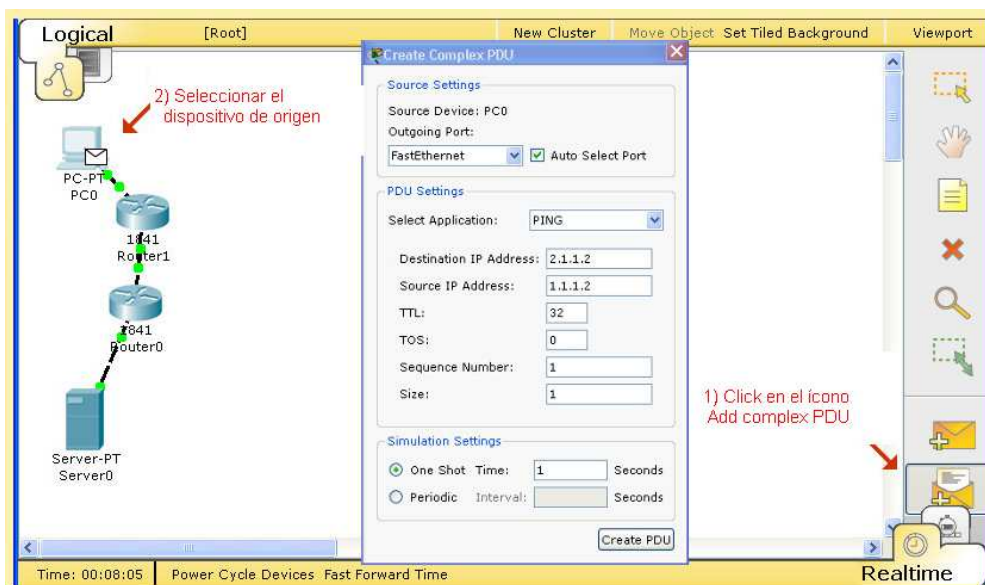


Fig. 2.25 Configuración del envío personalizado de paquetes.

Fuente: el autor

Packet Tracer soporta paquetes personalizados con puertos de fuente y destino correspondientes a los siguientes protocolos de aplicación: DNS, Finger, HTTP, HTTPS, IMAP, NetBIOS, Ping, POP3, SFTP, SMTP, SNMP, SSH, Telnet, TFTP, y otros.

2.6.3 Conexiones y enlaces

Packet Tracer soporta un amplio rango de conexiones de red. Cada tipo de cable puede sólo ser conectado entre determinadas interfaces:



Consola (*Console*): las conexiones de consola pueden realizarse entre PCs y routers o switches. Ciertas condiciones deben cumplirse para la sesión de consola de la PC para trabajar: la velocidad en ambos lados de la conexión debe ser la misma, los bits de datos deben ser 7 u 8 para ambos, la paridad debe ser la misma, los bits de parada deben ser 1 o 2 (no tienen que ser igual) y el control de flujo puede ser cualquiera por cualquier lado.



Cobre Straight-through (*Copper Straight-through*): este tipo de cable es el medio estándar Ethernet para la conexión entre dispositivos que operan en diferentes capas del modelo OSI (de hub a router, de switch a PC, de router a hub). Éste puede conectarse a los siguientes tipos de puertos: 10 Mbps Copper (Ethernet), 100 Mbps Copper (Fast Ethernet), y 1000 Mbps Copper (Gigabit Ethernet).



Cobre Cross-over (*Copper Cross-over*): este tipo de cable es el medio Ethernet para la conexión entre dispositivos que operan en la misma capa del modelo OSI (de hub a hub, de PC a PC, de PC a impresora). Éste puede conectarse a los siguientes tipos de puertos: 10 Mbps Copper (Ethernet), 100 Mbps Copper (Fast Ethernet), y 1000 Mbps Copper (Gigabit Ethernet).



Fibra (*Fiber*): la fibra es el medio utilizado para hacer conexiones entre los puertos de fibra (100 Mbps o 1000 Mbps).

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.



Línea telefónica (*Phone*): las conexiones de línea telefónica pueden realizarse solamente entre dispositivos con puertos de modem. La aplicación estándar para conexiones de módem es un dispositivo final (una PC por ejemplo) de marcación en una nube de red.



Coaxial (*Coaxial*): El cable coaxial es utilizado para hacer conexiones entre puertos coaxiales tal como un modem cableado conectado a una nube Packet Tracer.



DCE y DTE serie (Serial DCE and DTE): las conexiones series, frecuentemente se utilizan para enlaces WAN, deben conectarse entre puertos series.

Enlaces inalámbricos: Se pueden establecer enlaces inalámbricos entre puntos de acceso y dispositivos finales (PCs, servidores e impresores). Para establecer un enlace, simplemente se remueve el módulo existente del dispositivo final, se inserta un módulo inalámbrico y se enciende el dispositivo. El dispositivo tratará automáticamente de asociarse con el punto de acceso más cercano.

2.6.4 Modelado en Packet Tracer

Packet Tracer simula el comportamiento de las redes y dispositivos reales utilizando modelos. Como con todas las simulaciones, el programa está inherentemente limitado por las decisiones de modelado.

a) Modelos de la capa 1

Los modelos de la capa 1 están en basados las siguientes pautas:

¿Cómo los hubs procesan las tramas?,

¿Cómo los repetidores procesan la trama?

¿Cómo deben modelarse los enlaces inalámbricos?

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

b) Modelos de la capa 2

Los modelos de la capa 2 están basados en las siguientes pautas:

- ¿Cómo los switches procesan las tramas entrantes?
- ¿Cómo los switches envían las tramas?
- ¿Cómo debe modelarse CSMA/CD?
- ¿Cómo los switches utilizan el protocolo Spanning Tree Protocol (STP)?
- ¿Cómo funciona la seguridad en los puertos?
- ¿Cómo el protocolo DTP decide el modo del puerto?
- ¿Cómo los switches procesan Las tramas VTP entrantes?
- ¿Cuándo los switches envían tramas VTP?
- ¿Cómo funcionan los canales Ethernet?
- ¿Cómo funciona la conmutación multicapa?
- ¿Cómo funciona HDLC?
- ¿Cómo funciona el protocolo PPP?
- ¿Cómo funciona el cable DSL?
- ¿Cómo funciona el protocolo PPPoE?
- ¿Cómo funciona Frame Relay?

c) Modelos de la capa 3

Los modelos de la capa 3 están basados en las siguientes pautas:

- ¿Cómo los routers inician el proceso RIP?
- ¿Cuáles son las versiones de RIP?
- ¿Cómo un router envía actualizaciones de RIP?
- ¿Cómo un router procesa los paquetes RIP entrantes?
- ¿Cómo un router procesa los paquetes EIGRP entrantes?
- ¿Cómo un router procesa los paquetes OSPF entrantes?
- ¿Cómo un router procesa los paquetes entrantes?
- ¿Cómo funciona CEF?
- ¿Enrutamiento IPv6 y protocolos de enrutamiento?
- ¿Cómo un router procesa los paquetes BGP entrantes?

d) Modelos de la capa 4

Los modelos de la capa 4 están basados en las siguientes pautas:

¿Cómo los dispositivos procesan los segmentos UDP?

¿Cómo la conexión TCP maneja los datos salientes?

¿Cómo la conexión TCP maneja los segmentos entrantes?

e) Modelos de la capa 5

Los modelos de la capa 5 están basados en las siguientes pautas:

¿Cómo los routers y las PCs manejan los segmentos ISAKMP?

f) Modelos de la capa 7

¿Cómo los servidores y clientes DHCP procesan los paquetes entrantes?

¿Cómo los servidores y clientes DHCP para IPv6 procesan los paquetes entrantes?

¿Cómo funcionan los procesos de los clientes y servidores FTP?

¿Cómo funcionan los procesos de los clientes y servidores TELNET?

¿Cómo funcionan los procesos de los clientes y servidores SSH?

¿Cómo funciona HTTP?

¿Cómo funciona SMTP?

¿Cómo funciona POP3?

¿Cómo funciona un servidor DNS?

¿Cómo funciona VoIP?

Otras

2.6.5 Activity Wizard

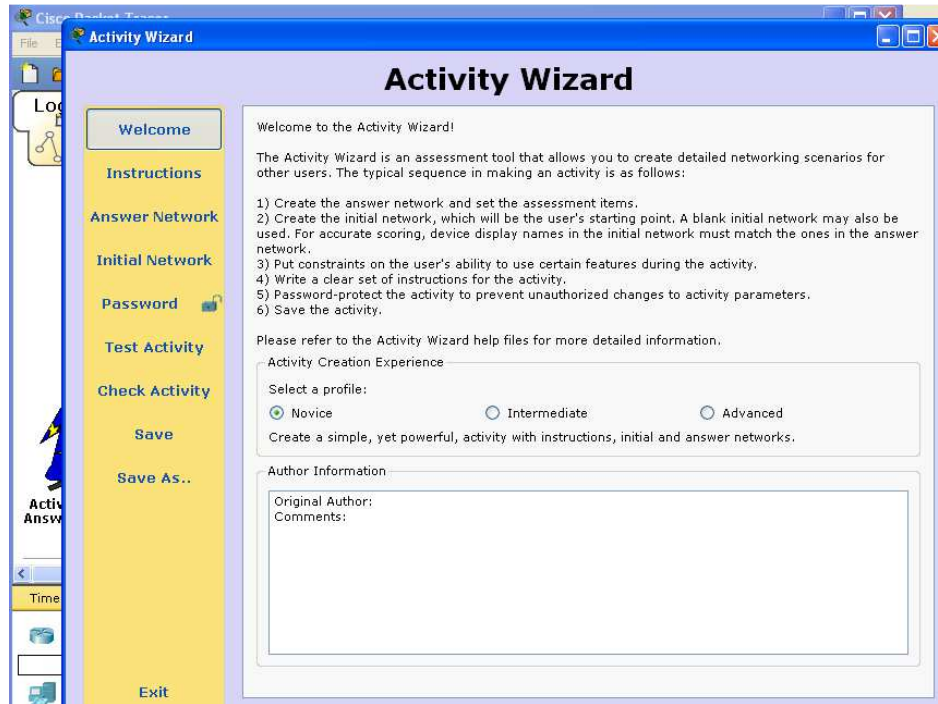


Fig 2.26. El Activity Wizard de Packet Tracer.

Fuente: el autor

El Activity Wizard es una herramienta de evaluación que permite la creación de escenarios de red altamente específicos para otros usuarios. Esta herramienta es particularmente útil para que los instructores puedan crear actividades para sus estudiantes. Cuando un estudiante inicia una actividad, se le presenta un diseño de red inicial y un conjunto de instrucciones. Los estudiantes siguen las instrucciones para completar la actividad. Ellos finalmente pueden comparar sus resultados con las respuestas del instructor. El instructor tiene control total sobre todos los aspectos de la actividad.

2.6.6 Multiusuario

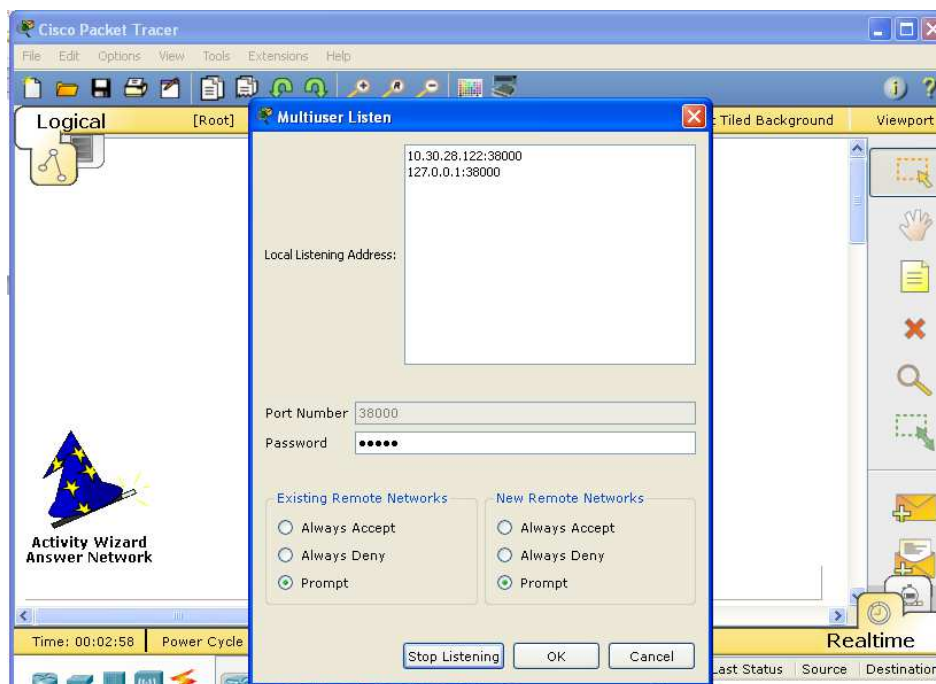


Fig. 2.27. Opción multiusuario (*multiuser*) de Packet Tracer.
Fuente: el autor

Las comunicaciones multiusuario permiten múltiples conexiones punto a punto entre múltiples instancias de Packet Tracer. Al permitir la comunicación entre múltiples instancias de Packet Tracer se abren las puertas a un entorno de aprendizaje divertido, interactivo, social, colaborativo y competitivo. Para los instructores está ahora disponible la creación de una gran variedad de actividades para que los estudiantes aprendan en grupo. Los estudiantes estarán beneficiados con este entorno puesto que trabajarán juntos en la solución de problemas y comparten ideas. Ambos, estudiantes y profesores deberían tomar las ventajas que les ofrece para el aprendizaje la opción Multiusuario de Packet Tracer.

2.6.7 IPC

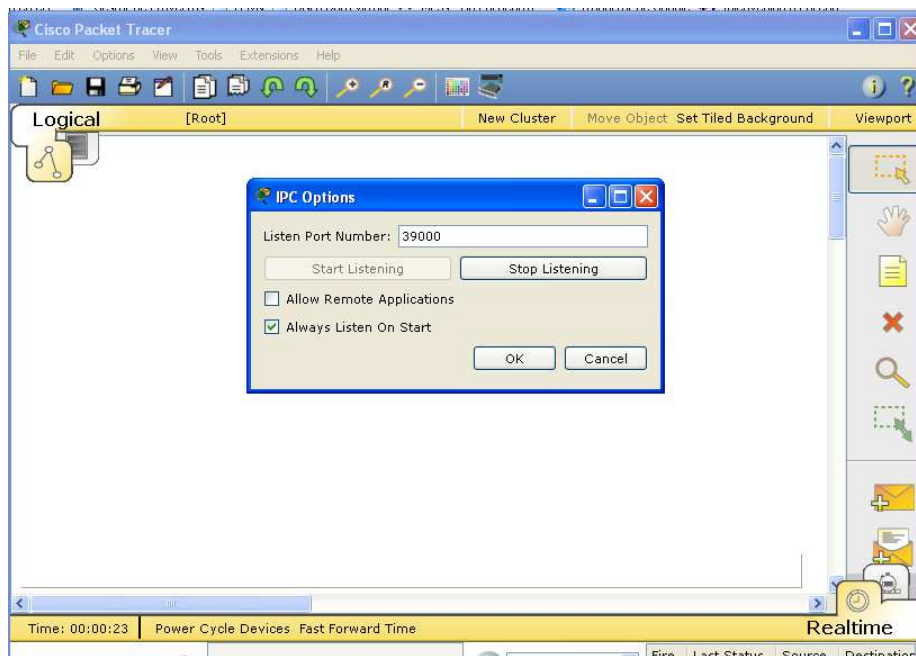


Fig. 2.28. Opción IPC de Packet Tracer.
Fuente: el autor

La opción IPC (*Inter-Process Communication*, Comunicación entre procesos) controla la función de una extensión de programación externa para Packet Tracer, que permite que aplicaciones externas (*ExApps*) puedan añadirse a Packet Tracer para extender su funcionalidad. Las aplicaciones externas están siendo creadas por Cisco y otros grupos interesados para su utilización en el Programa de la Academia. Un ejemplo de un caso de uso de una aplicación externa podría incluir un programa que pueda automatizar la tarea de envío y recibimiento remoto de las actividades de evaluación entre los estudiantes y los instructores. Debido al control que IPC ofrece en relación con las aplicaciones externas, esto amplía enormemente las capacidades que Packet Tracer puede proporcionar sin tener que esperar a que sea mejorado el núcleo actual del programa.

2.6.8 Patrones de diseño de actividades

Existen cuatro tipos principales de patrones de diseño de actividades en Packet Tracer: fomentadores de conceptos, fomentadores de habilidades, desafíos de diseño y manejo de problemas.

- a) Fomentadores de conceptos: los fomentadores conceptuales se basan en indagaciones en la construcción de modelos e investigaciones que guían a los estudiantes a explicaciones y animaciones de conceptos de redes, especialmente algoritmos de dispositivos y protocolos de red. Uno de los usos para los que fue previsto Packet Tracer es para que los estudiantes construyan sus propios modelos o redes virtuales, obtengan acceso a representaciones gráficas importantes de estas redes, animar estas redes mediante la adición de sus propios paquetes de datos, responder preguntas sobre estas redes, y finalmente, anotar y salvar sus creaciones. El término “packet tracing” describe un modo animado donde los estudiantes pueden pasear a través de los eventos de red simulados, uno a la vez, para investigar la microgénesis de los fenómenos que normalmente ocurren a velocidades de millones de eventos por segundo. Un ejemplo de constructores conceptuales pudiera ser ilustrar el proceso de reenvío de los hubs. Otro ejemplo pudieran incluir el diseño de una red que permitiera comparar el contraste del comportamiento entre los hubs y los switches.
- b) Fomentadores de habilidades: los fomentadores de habilidades apoyan la resolución de problemas algorítmicos en apoyo al desarrollo del conocimiento de procedimientos de redes. Por ejemplo, los problemas para fomentar habilidades simples pueden incluir que los estudiantes desarrollen laboratorios prácticos antes de trabajar con el equipo real, después de trabajar con el equipo real, o sólo para practicar. Esto posibilita que los estudiantes pueden probar sus configuraciones antes de asistir a clase.
- c) Desafíos de diseño: los desafíos de diseño son problemas basados en restricciones con múltiples soluciones correctas. Estas pueden variar desde muy simples, a intermedios o complejos. Algunos instructores

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

proponen a los alumnos utilizar Packet Tracer para verificar la funcionalidad de los esquemas de direccionamiento IP que ellos han diseñado.

- d)** Manejo de problemas: las actividades de manejo de problemas incluyen diagnóstico, aislamiento y reparación de la red simulada, teniendo como fuente un ejemplo erróneo. Las actividades de manejo de problemas puede variar desde simple (velocidad de Ethernet y desajustes dúplex, direcciones IP en la subred equivocada, incorrecta elección de cables, u olvidar los ajustes del reloj en las interfaces serie) a complejas (actualizaciones de enrutamiento incorrectas, múltiples fallas de red interactuando). Otro ejemplo incluso pudiera ser en una red muy compleja con miles de componentes potencialmente configurables, pudiera introducirse un simple error, y la actividad consistiría en que los estudiantes diagnostiquen, aíslen y reparen el error.

2.6.9 Requerimientos del sistema

Mínimo:

- CPU: Intel Pentium III 500 MHz o equivalente
- Sistema Operativo: Microsoft Windows XP, Microsoft Windows Vista, Microsoft Windows 7, Fedora 11, o Ubuntu 8.04 LTS
- RAM: 250 MB libres
- Almacenamiento: 250 MB de espacio libre en disco
- Resolución del display: 800x600
- Adobe Flash Player
- Los últimos controladores de la tarjeta de video y actualizaciones del sistema operativo.

Recomendado:

- CPU: Intel Pentium III 1.0 GHz o mejor
- RAM: 512 o más.

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

- Almacenamiento: 300MB de espacio libre en disco
- Resolución del display: 1024 x 768
- Tarjeta de sonidos y altavoces.
- Conectividad a Internet(si se utiliza la característica de Multiusuario)

2.7 FLAN (F- Links And Nodes)

FLAN es un software desarrollado con el lenguaje de programación Java y se distribuye con licencia pública GNU. Se considera que pertenece al grupo de los simuladores de propósito general ya que por medio de Java se pueden crear y configurar nuevos dispositivos, aplicaciones o protocolos de red, aun si no están incluidos dentro de las librerías del programa, inclusive se pueden realizar modificaciones al código fuente de FLAN [11].

2.7.1 Características generales

FLAN es una herramienta de simulación que permite el diseño, la construcción, y la prueba de una red de comunicaciones en un ambiente simulado. El programa hace el análisis de las redes asociando su estructura basada en nodos y enlaces, con bloques simples, por medio de los cuales se puede entender el funcionamiento especialmente de los protocolos de enrutamiento que maneja la capa de red.

2.7.2 Interfaz gráfica de usuario

El área de trabajo de este simulador consta de tres módulos bien identificados: en primer lugar una ventana principal, después una ventana de consola y finalmente una interfaz de salida gráfica.

- a) Ventana principal: en éste módulo se encuentran todos los elementos necesarios para crear y manipular una topología de red. Dentro de esta ventana se incluyen la barra de menú, la barra de herramientas, y la hoja de dibujo, es decir el área de trabajo. Se puede acceder a archivos que se hayan guardado y simulado con anterioridad, utilizando las opciones de la barra de menú, además, la mayoría de los elementos

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

necesarios para simular se encuentran disponibles en la barra de herramientas en forma de botones que se complementan con la representación gráfica de la acción o dispositivo que representan.

- b) Ventana de consola: este módulo, muestra al usuario la información de la red y proporciona información sobre las acciones que se están ejecutando durante la simulación, es decir, mediante este módulo el usuario tiene la capacidad de ver no solamente los eventos ocurridos en la hoja de dibujo, sino también analizar y seguir las acciones que esos eventos producen en la red.

- c) Interfaz de salida gráfica: La interfaz de salida permite al usuario analizar lo sucedido durante la simulación en forma gráfica, mediante un eje de coordenadas, donde las ordenadas (eje y), representan el número de paquetes y las abscisas representan el tiempo.

2.7.3 Requerimientos del sistema

Para instalar el simulador FLAN, es necesario tener previamente el Kit de Desarrollo de Java J2SE (*Java 2 Platform Standard Edition*). Este kit incluye una JVM (*Java Virtual Machine*, Máquina Virtual de Java), una API (*Application Programming Interfaces*, Interface de Programación de Aplicaciones) y un compilador que se necesita para desarrollar y compilar en FLAN.

- CPU: Intel Pentium 250 MHz o equivalente
- Sistema Operativo: Microsoft Windows 98/ 2000, Linux, UNIX, Mac OS X.
- RAM: 64 MB libres
- Almacenamiento: 20 MB de espacio libre en disco

2.7.4 Ventajas

- Este es un software multiplataforma, es decir que puede ser implementado sobre cualquier sistema operativo que soporte la máquina virtual de Java.

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

- El programa contiene además, unas herramientas llamadas manejadores, que son protocolos específicos que ayudan a determinar cómo es recibida la información, cómo procesarla y además cómo dirigir la simulación.
- Los manejadores podrían incluir Protocolo IP por ejemplo, que conduce la simulación hacia el mundo del IP. Esto incluiría tomar datos abstractos tales como entradas y direcciones de la tabla de encaminamiento, y el proceso de ellas según el estándar del IP. Los manejadores también incluyen paquetes de datos para distintos tipos de datos.

2.7.5 Desventajas

- Flan está diseñado para la prueba de protocolos en redes pequeñas, es decir, que tengan menos de 100 nodos. Aunque el usuario puede tener tantos nodos como desee, el funcionamiento se verá afectado mientras se agreguen más y más nodos.
- El programa permite que se trabaje con la interfaz gráfica, sin embargo es necesario tener conocimientos básicos sobre el lenguaje de programación Java, para poder hacer más configuraciones con el software y dar solución a problemas que se presenten al momento de definir características o parámetros de los dispositivos, protocolos y/o aplicaciones.
- Por otra parte pueden presentarse problemas al compilar en FLAN si no se tiene la versión apropiada del JDK de Java.

2.8 Comnet III

COMNET III es una herramienta comercial orientada al diseño, configuración y estudio de las redes de comunicaciones, desarrollado por CACI Products Inc, haciendo uso del lenguaje de programación MODSIM II. Por medio de este programa es posible crear topologías de redes complejas, configurar varias tecnologías, protocolos y dispositivos de red, para hacer un análisis detallado del

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

funcionamiento y del rendimiento de redes tipo LAN, MAN y WAN, utilizando una interfaz gráfica en un ambiente de ventanas [11].

2.8.1 Características generales

Este software es de tipo gráfico, que permite analizar y predecir el funcionamiento de redes informáticas, desde topologías básicas de interconexión hasta esquemas mucho más complejos de simulación con múltiples redes interconectadas con diversos protocolos y tecnologías como Ethernet, ATM, Satelitales, Frame Relay, X25, etc. Dentro del área de trabajo del programa, se hace la descripción gráfica del modelo de red, se asocian las fuentes generadoras de tráfico en la red, se configuran los parámetros y las características de los dispositivos de acuerdo a la aplicación que se desea implementar, luego se pone en marcha la simulación y finalmente, se analizan los resultados estadísticos sobre el desempeño de la red, los cuales son programados antes de iniciar la simulación y que se generan automáticamente cuando se concluye la simulación. Entre los parámetros que se pueden incluir dentro de los informes de la red están: la ocupación de enlaces o nodos, la cantidad de mensajes generados, el número de colisiones, entre otros. [16]

Este programa contiene una gran variedad de dispositivos de red como: hosts, hubs, switches, routers, puntos de acceso, satélites, entre otros; los cuales pueden ser interconectados con enlaces y tecnologías como: Ethernet, FDDI, punto a punto, Frame relay, Aloha, PVC, CSMA, entre otros; a la vez que permite implementar gran variedad de protocolos; es decir COMNET III presenta características muy completas e interesantes, en cuanto a las interfaces que soporta para su uso. Sin embargo cabe mencionar que el máximo desempeño de este simulador se alcanza al utilizar las librerías para los diferentes tipos de dispositivos de redes con sus diferentes parámetros.

Además, esta herramienta es muy útil para fines didácticos en el área de las telecomunicaciones ya que adentra al usuario al mundo de las redes de forma amena, obligándolo a familiarizarse con los términos reales de los estándares

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

existentes en las redes de comunicaciones, independientemente de cuál sea la aplicación.

El simulador es capaz de soportar cualquier tipo de redes de comunicaciones, aunque se necesita un panorama muy completo en cuanto a lo que existe en el mercado y la implementación de redes en la práctica.

2.8.2 Interfaz gráfica de usuario



Fig. 2.29 Interfaz gráfica de usuario de Comnet III.
Fuente: [16]

COMNET III utiliza un ambiente gráfico de ventanas, el cual tiene una serie de menús y barras de herramientas que permiten crear el modelo de la red que se va a simular. Esta característica hace de COMNET una herramienta ideal para la academia, ya que los tiempos de aprendizaje y de implementación de una simulación son cortos, si se tiene en cuenta las tecnologías y protocolos que soporta. [16]

2.8.3 Requerimientos del sistema

- CPU: Pentium a 250 MHz o equivalente

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

- Sistema Operativo: Microsoft Windows 95 en adelante. Aunque también existe una versión para Solaris 2.5 o SunOS 5.5, HP- UX 10.2, SGI - IRIX 6.3
- RAM: 32 MB libres
- Almacenamiento: 25 MB de espacio libre en disco

2.8.4 Ventajas

- El programa ofrece la posibilidad de simular una gran cantidad de protocolos y tecnologías de red, y ofrece la posibilidad de crear protocolos a medida que se van necesitando.
- Permite configurar y observar una gran cantidad de parámetros durante la simulación como: colisiones, capacidad de los buffers de entrada y salida de los dispositivos, utilización del canal, anchos de banda, etc.
- Ofrece la posibilidad de ver el intercambio de mensajes entre los nodos de la red de manera gráfica, según avanza la simulación.
- Permite obtener gráficos y/o archivos de texto con las estadísticas de la simulación.
- Se pueden diseñar, configurar y simular redes complejas, que incluyan planes de contingencia, seguridad e implementación de tecnologías de superposición como LAN Emulada (LANE).

2.8.5 Desventajas

- A pesar de ser COMNET III un software muy poderoso, en la edición universitaria presenta algunas limitaciones ya que no se pueden realizar las simulaciones que involucren más de 20 nodos.
- Es un software propietario.
- Por ser una de las herramientas de simulación más completas del mercado, la programación de los parámetros de los dispositivos y enlaces de la red tiende a ser compleja.

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

- Además de los conocimientos sobre el manejo y el diseño de redes de comunicaciones, se requieren conocimientos en otras áreas como por ejemplo la estadística.

2.9 OMNET++

Es un programa orientado a simular objetos y a modular eventos discretos en redes de comunicaciones, posee una gran cantidad de herramientas y una interfaz que puede ser manejada en plataformas Windows y en distribuciones tipo Unix; haciendo uso de varios compiladores de C++. OMNET ++ es una versión libre, para fines académicos de la versión comercial OMNEST, desarrollada por Omnest Global, Inc. OMNET++, así como las interfaces y las herramientas, se pueden ejecutar perfectamente sobre sistemas operativos Windows y sobre algunas versiones de UNIX y Linux, usando varios compiladores de C++. [11]

2.9.1 Características generales.

OMNET++ es una herramienta eficiente, enfocada al área académica y desarrollada para modelar y simular eventos discretos en redes de comunicaciones. Básicamente este simulador de redes recrea dichos eventos discretos por medio de módulos orientados a objetos, puede ser utilizado para modelar el tráfico de información sobre las redes, los protocolos de red, las redes de colas, multiprocesadores y otros sistemas de hardware distribuido. Además, para validar arquitecturas de hardware y evaluar el rendimiento de sistemas complejos. Este simulador, utiliza el lenguaje de programación NED, que se basa en el lenguaje C++, como herramienta para modelar topologías de red. Este lenguaje facilita la descripción modular de una red, es decir, un modelo en OMNET ++ se construye con módulos jerárquicos mediante el lenguaje NED, dichos módulos pueden contener estructuras complejas de datos y tienen sus propios parámetros usados para personalizar el envío de paquetes a los destinos a través de rutas, compuertas y conexiones. Los módulos de más bajo nivel son llamados “*simplemodules*” y son programados en C++ usando la librería de

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

simulación. Básicamente, con el lenguaje NED se definen tres módulos: módulos simples, módulos compuestos y de redes, dentro de los cuales se encuentran los componentes y especificaciones de la descripción de una red de comunicaciones. Con el fin de facilitar el diseño de redes y la simulación de eventos sobre las mismas, OMNET ++, permite al usuario trabajar gráficamente empleando el editor del lenguaje NED (GNED). Este editor es la interfaz gráfica que permite crear, programar, configurar y simular redes de comunicaciones, sin necesidad de hacerlo utilizando la codificación del lenguaje NED, ya que automáticamente, GNED se encarga de generar el código del lenguaje, de acuerdo al diseño y configuración que realiza el usuario en forma gráfica. Además GNED, permite acceder fácilmente a dicho código. [13]

2.9.2 Interfaz de usuario

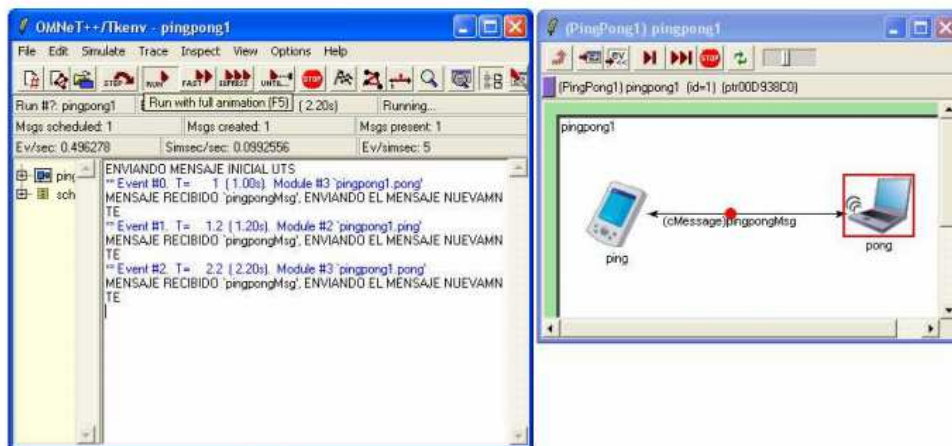


Fig. 2.30 Interfaz gráfica de usuario de Omnet++.
Fuente: [11]

Las simulaciones en OMNET++ pueden utilizar varias interfaces de usuario, dependiendo del propósito. La interfaz más avanzada permite visualizar el modelo, controlar la ejecución de la simulación y cambiar variables/objetos del modelo. Esto facilita la demostración del funcionamiento de un modelo. Para la interfaz de usuario, se pueden generar dos tipos de archivos ejecutables: interfaz de usuario gráfico e interfaz de consola.

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

- a) Interfaz de usuario gráfico: esta interfaz es útil para depurar y comprender, los procesos y configuraciones que se aplican a las redes. A esta interfaz gráfica se accede con el editor GNED. GNED es la herramienta que simplifica el desarrollo de las simulaciones con OMNET++, ya que permite trabajar sin necesidad de programar.
- b) Interfaz de consola: esta interfaz es más eficaz para realizar las simulaciones por lotes. OMNET++ contiene unas clases programadas en C++, diseñadas para recoger y exhibir datos estadísticos, de los resultados de la simulación.

2.9.3 Ventajas

- OMNET++ es gratuito solamente para propósitos académicos, lo que facilita su utilización en universidades y grupos de investigación.
- Gracias a la programación por módulos, es posible simular procesos paralelos y distribuidos, los cuales pueden utilizar varios mecanismos para comunicarse entre sí.
- Es multiplataforma.

2.9.4 Desventajas

- Para fines de investigación y desarrollo, es necesario saber programar en lenguaje NED, ya que el trabajo con el editor gráfico, es un poco más rígido.
- Por ser un software de aplicación en áreas comerciales y para efectos de investigación y desarrollo, tiene un alto grado de complejidad en su manejo.

2.10 NS

El Network Simulator más conocido como NS, es un software orientado a simular eventos discretos. Se desarrolló con base a dos lenguajes de programación: uno de ellos es un simulador escrito en C++ y el otro es una extensión de TCL19,

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

orientada a objetos. Este programa ha sido diseñado especialmente para el área de la investigación de redes telemáticas.

2.10.1 Características generales

NS es una herramienta con un amplio rango de uso y que continuamente sirve como base para el desarrollo de otros programas de simulación. Además este software soporta una gran cantidad de protocolos de las capas de aplicación y transporte, además de otros utilizados para el enrutamiento de los datos, entre los cuales están: HTTP, FTP CBR, TCP, UDP, RTP, SRM, entre otros; los cuales pueden ser implementados tanto en redes cableadas, como inalámbricas locales o vía satélite, y que son aplicables a grandes redes con topologías complejas y con un gran número de generadores de tráfico. NS depende de algunos componentes externos como: Tcl/TK, Otcl, TclCL20 que hacen parte del compilador de NS para Linux, además del xgraph, que es un componente opcional sólo para cuando se necesite evaluar series. [20][16]

Como se observa en la figura mostrada a continuación, NS2 es un intérprete de scripts del lenguaje TCL orientado a objetos, el cual tiene un planificador de eventos de simulación y librerías de objetos de componentes de red y librerías de módulos de instalación de red. Esto quiere decir que la simulación se debe programar en el lenguaje de scripts OTCL.

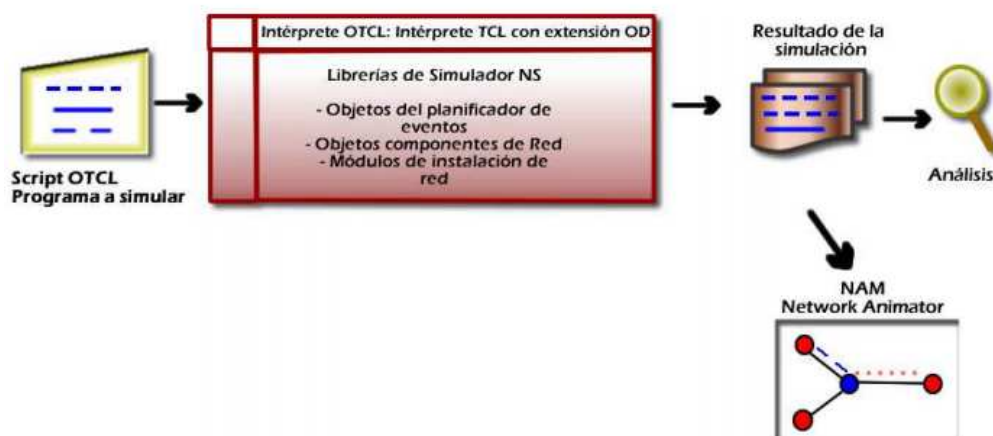


Fig. 2.31 Esquema de simulación en NS2.
Fuente: [16]

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

NS2 proporciona unas herramientas que facilitan la interpretación de los resultados de una manera gráfica. Estas herramientas son NAM (*Network Animator*) y XGraph que crean una representación a partir de los ficheros de traza generados por NS2. Los ficheros de traza reflejan todos los eventos que han sucedido durante la simulación de la red. Cada una de las trazas incluidas en estos ficheros contiene información sobre el instante de tiempo en que ha ocurrido, sobre los nodos implicados, información sobre la trama MAC, sobre el paquete IP, e información adicional que depende del evento que se ha producido o del tipo de paquete del que se trata.

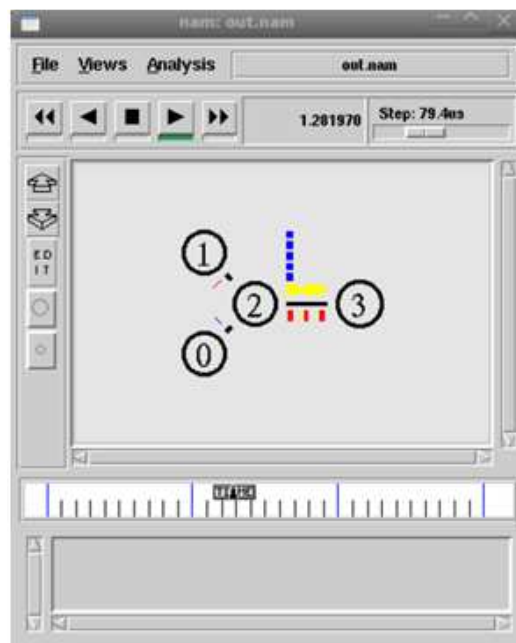


Fig. 2.32 Herramienta NAM.
Fuente: [11]

2.10.2 Interfaz de usuario

NS2 tiene un editor de topología por código, con el cual se diseña y se configuran las redes, los protocolos y las aplicaciones de red que se desean simular. También cuenta con una herramienta llamada Simulador de red automatizado (*Automated Network Simulation*), este asistente automáticamente carga las tareas que se ejecutan más frecuentemente en los dispositivos de la red. [20]

2.10.3 Requerimientos del sistema

NS2 es un paquete compuesto por un conjunto de componentes requeridos y otros tantos opcionales. Este paquete contiene un script de instalación para configurar, compilar e instalar estos componentes. Para instalar este software se requiere cumplir con las siguientes especificaciones:

- CPU: Procesador PentiumII de 200 MHz o equivalente.
- Sistema Operativo: Plataformas Unix (FreeBSD, Linux, SunOS, Solaris), Windows desde la versión 95
- RAM: 32 MB de memoria RAM
- Almacenamiento: 320 MB de espacio libre en disco
- Software: Para plataformas tipo UNIX Tcl release 8.4.5, Tk release 8.4.5, Otcl release 1.9, TclCL release 1.16, Ns release 2.28, otros componentes opcionales: Nam release 1.11, Xgraph version 12, CWeb version 3.4g, SGB versión 1.0. En sistemas Windows es necesario MS Visual C++ 5.0 (o superior). Otra forma de instalarlo, es a través de un programa de emulación de Linux, tal como Cygwin.

2.10.4 Ventajas

- Este programa contiene módulos que cubren un extenso grupo de aplicaciones, protocolos de enrutamiento, transporte, diferentes tipos de enlaces, estrategias y mecanismos de entrenamientos, entre otros. Algunos de estos son: HTTP, TcpApp, Telnet, CBR (Constat BitRate), TCP, RTP, algoritmos de enrutamiento jerárquico y enrutamiento manual.
- Por ser uno de las más antiguas herramientas de simulación, el NS se ha convertido en un estándar de su área, esto ha llevado a que sea ampliamente utilizado, ya que se encuentran en Internet un gran número de ayudas y proyectos realizados sobre NS.

2.10.5 Desventajas

- La configuración de las simulaciones a través de código, hace que sea mayor el tiempo de desarrollo.
- Tiempo elevado para el aprendizaje.
- NS requiere la instalación de varios componentes adicionales para su correcto funcionamiento.

2.11 NCTUns

NCTUns (*National Chiao Tung University, Network Simulator*) es un simulador y emulador de redes y sistemas de telecomunicaciones avanzado. NCTUns es software libre y se ejecuta sobre Linux. Además utiliza una metodología de simulación que entra y modifica el Kernel de Linux, lo cual hace que el programa tenga ventajas únicas en comparación con otros simuladores y emuladores de redes de comunicaciones. [3]

NCTUns ha recibido varios reconocimientos a nivel internacional, debido a las prestaciones que ofrece y al desarrollo del programa. Algunos de estos reconocimientos son: MobiCom 2002 y2003, Reporte especial en el revista de la IEEE – Julio de 2003, IEEE MASCOTS 2004, IEEE vehicular technology society, IEEE INFOCOM 2005, etc. Esto evidencia el impacto que ha causado este programa en el ambiente de la simulación de redes de comunicaciones. Este simulador permite desarrollar, evaluar y diagnosticar el desempeño de protocolos y aplicaciones en diferentes tipos de redes (LAN, MAN, WAN). Las simulaciones hechas con esta herramienta, cuentan con características muy especiales, ya que NCTUns simula en tiempo real y con una interfaz similar a la de los sistemas reales, lo cual permite familiarizar más al usuario con el manejo del diseño, configuración e implementación de aplicaciones en redes de comunicaciones.[3][11]

2.11.1 Características generales

NCTUns utiliza una sintaxis sencilla pero muy efectiva para describir la topología, los parámetros y la configuración de una simulación, esta descripción se genera a partir de la interfaz gráfica del usuario.

NCTUns fue desarrollado basado en el simulador NS, de ahí su nombre, sólo que incluye una interfaz más amigable para la implementación de los modelos de red que se simulan. Este programa permite la simulación de arquitecturas de redes sencillas, sin embargo, su mayor potencial está en la simulación de redes tan complejas como las redes GPRS, satelitales y ópticas. El NCTUns también puede ser utilizado como emulador, especialmente para redes móviles e inalámbricas; para dichas aplicaciones provee recursos para manejo y estudio de sistemas de radiofrecuencia y permite obtener mediciones para establecer niveles de calidad de servicio (*QoS*) de las señales irradiadas.

El hecho de que el simulador permita definir obstáculos, trayectorias de movimiento y que los terminales móviles (como celulares GPRS y portátiles) se puedan desplazar siguiendo dicha trayectoria, al mismo tiempo en que se hacen mediciones de atenuación, interferencia y de ancho de banda, dan cuenta de las sobresalientes características del NCTUns y justifican los diferentes reconocimientos que ha obtenido a nivel mundial. Adicionalmente, permite simular redes ópticas y como si fuera poco, puede usarse fácilmente como un emulador, cuando se desee desarrollar funciones de desempeño de un host real y ver cómo se comportaría bajo diferentes tipos de condiciones de red sin modificar su protocolo interno. Esto quiere decir que NCTUns tiene la posibilidad de emular un dispositivo de red del mundo real en su entorno gráfico e interconectarlo con dispositivos simulados o virtuales, para intercambiar paquetes. También posee una característica importante, la cual, sumado a lo anteriormente expuesto, hacen de NCTUns uno de los más poderosos simuladores de redes de telecomunicaciones. La arquitectura de sistema abierto, en la cual la GUI y el motor de simulación son elementos separados que utilizan un modelo cliente servidor, permite ejecutar simulaciones remotas, paralelas, distribuidas y concurrentes, lo cual permite entre muchas otras cosas, correr simulaciones simultáneamente en diferentes nodos de una red y cuyos resultados individuales

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

sirven para el análisis de un sistema único. Esto quiere decir, que un usuario, puede enviar su proyecto de simulación a un servidor remoto que esté ejecutando el motor de simulación, utilizando su propia GUI y además correr múltiples simulaciones concurrentes en diferentes hosts conectados a dicho servidor.[11]

2.11.2 Interfaz gráfica de usuario

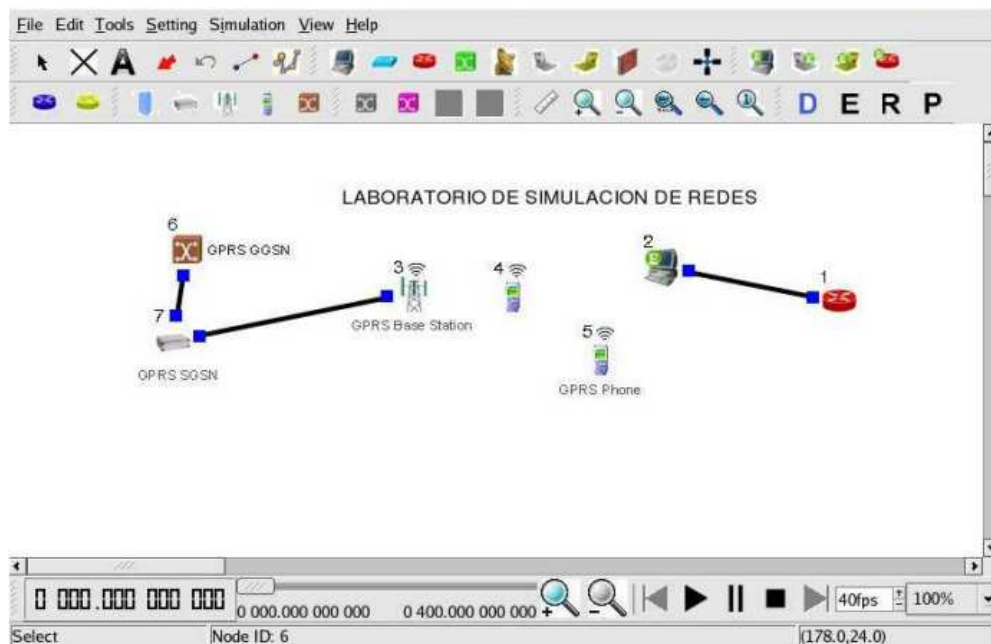


Fig. 2.33 Interfaz gráfica de usuario de NCTUns.
Fuente: [11]

NCTUns provee una GUI (Interfaz Gráfica de Usuario) profesional y de alta integración, en la cual el usuario diseña y edita la topología de la red, configura los módulos de protocolos que manejará cada nodo de la red, asigna valores y define parámetros específicos de cada dispositivo. Desde la interfaz de usuario, se programa y se visualiza la animación de la transferencia de paquetes durante el proceso de simulación, el desplazamiento; de los terminales móviles y la presentación de resultados mediante gráficos estadísticos. Además el usuario puede consultar el desarrollo de los procesos que se está ejecutando en determinado dispositivo durante la simulación, sin necesidad de pararla o cancelarla.

2.11.3 Requerimientos del sistema

- CPU: Pentium de 1 GHz
- Sistema Operativo: Linux Red Hat, Fedoracore 2.0 y core 3.0. Otras distribuciones de Linux como Debian.
- RAM: 256 MB libres
- Almacenamiento: 200 MB de espacio libre en disco

Se recomienda utilizar un procesador con mayor velocidad, es decir mayor a 1GHz y en lo posible disponer de más de 200 MB de espacio en el disco. Además, a nivel software, es importante tener instalado un compilador gcc, el sistema Xwindows Gnome o Kde, que el usuario tenga los privilegios del administrador o root para el manejo del programa y que se asigne el shell Bash/tcsh al usuario para el manejo por comandos en el modo consola.

2.11.4 Ventajas

- Es un software libre, con distribución de código abierto.
- Utiliza directamente el conjunto de protocolos TCP/IP de Linux, por consiguiente se generan resultados de simulación de alta fidelidad y permite que la configuración y el uso de una red simulada, sea exactamente igual a los usados en redes IP del mundo real. Puede ser utilizado como emulador. Esto permite que un host externo conectado a una red del mundo real, pueda intercambiar paquetes con nodos (por ejemplo: host, enrutadores o estaciones móviles celulares) en una red simulada en NCTUns.
- Puede utilizar cualquier aplicación de UNIX existente en la vida real, como un generador de tráfico, además, puede utilizar las herramientas de configuración y monitoreo de UNIX.
- Puede simular redes fijas, inalámbricas, redes celulares, redes GPRS y redes ópticas.

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

- Puede simular una gran variedad de dispositivos de red, como: hubs, switches, enrutadores, estaciones móviles, puntos de acceso de WLANs, teléfonos GPRS, etc, así como obstáculos para las señales inalámbricas, además ofrece alta velocidad de simulación.
- Simula varios protocolos de redes como: IEEE 802.3, IEEE 802.11, IP, IPMobile, Diffserv, RIP, OSPF, UDP TCP, RTP/RTCP, SDP, FTP, etc.

2.11.5 Desventajas

- Solamente funciona en sistemas Fedora core 3, para otras distribuciones de Linux es necesario hacer pruebas y configuraciones adicionales.
- Existe muy poca información sobre el funcionamiento y configuración del software. El anterior punto lleva a que sea mayor el tiempo de aprendizaje del simulador.
- El servicio de soporte proporcionado por los autores del proyecto NCTUns es deficiente y en algunas ocasiones no funciona.

2.12 KIVA

Es un simulador de redes basado en Java que permite especificar diferentes esquemas de redes de datos y simular el encaminamiento de paquetes a través de dichas redes. [3][12]

2.12.1 Características generales

Kiva es una herramienta software orientada principalmente a simular el comportamiento del protocolo IP, y especialmente para el estudio del tratamiento de los datagramas y el encaminamiento de los mismos por una red. También al utilizarlo, se puede estudiar el funcionamiento de los protocolos auxiliares ARP e ICMP y emular el funcionamiento básico de tecnologías de enlace como Ethernet. Con esta herramienta, se puede diseñar una topología de red con la interfaz

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

gráfica, configurar el direccionamiento y las tablas de encaminamiento para los dispositivos y simular el envío de paquetes de un equipo a otro.

La principal aplicación del programa es en la enseñanza de los fundamentos sobre el funcionamiento de redes de datos, pero este entorno, también puede ser muy útil para el diseño y comprobación del encaminamiento en redes de datos a nivel comercial.

El objetivo principal de este programa, es ayudar a diseñar y comprender el funcionamiento de redes de datos y en especial el encaminamiento de paquetes en la arquitectura TCP/IP, sin necesidad de una infraestructura real y de herramientas de análisis de tráfico. Este programa, también es capaz de simular distintos tipos de errores en el funcionamiento de las redes, como la pérdida de paquetes o fallos en tablas de encaminamiento.

El programa es multiplataforma, dado que todo su entorno fue desarrollado con el programa de simulación Java, además Kiva ofrece un API que permite usar las funciones de simulación desde otras aplicaciones de Java.

2.12.2 Interfaz gráfica de usuario

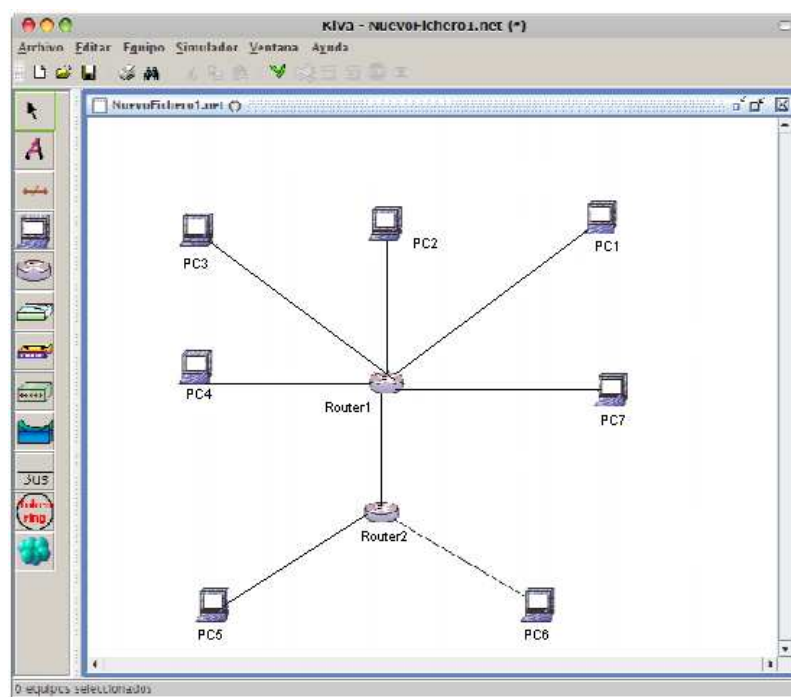


Fig. 2.34. Interfaz gráfica de KIVA.
Fuente: [12]

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

En la versión actual, la interfaz de usuario está implementada con un conjunto de clases que se deben descargar y ejecutar en el equipo del usuario, cada vez que se desee trabajar con el programa.

Kiva se compone de dos partes, totalmente implementadas con el lenguaje de programación Java. La primera es un API (*Application Programming Interface*, Interface de Programación de Aplicaciones), que ofrece un motor de simulación de redes a otras aplicaciones; este API está formado por cuatro bloques: el primero de estos es el bloque de gestión de eventos discretos, el segundo es el de los objetos que representan las redes de datos, el tercer bloque es el de los objetos que representan los equipos finales o de interconexión y finalmente, aparece un cuarto bloque con la pila de comunicaciones. Los APIs son modulares y extensibles, de forma que se puedan ir incorporando fácilmente a éstos, nuevos tipos de redes y de equipos.

La segunda es propiamente la interfaz gráfica, la cual, también hace uso del API de simulación. La interfaz gráfica permite especificar las topologías de las redes de datos, mediante un editor gráfico; además permite la configuración del direccionamiento de los equipos de la red, el encaminamiento de la información y el acceso a las características que ofrece el API de simulación de una forma sencilla, sin necesidad de programar.

2.12.3 Requerimientos del sistema

- CPU: Pentium de 250 MHz o equivalente
- Sistema Operativo: Microsoft Windows y Linux.
- RAM: 32 MB libres
- Almacenamiento: 20MB de espacio libre en el disco

2.12.4 Ventajas

- El programa se distribuye con software libre y además es multiplataforma.

Capítulo II: Estudio de algunas herramientas software para la simulación de redes.

- Permite el estudio de las redes IP y especialmente el seguimiento y análisis del funcionamiento, el envío, el tratamiento y la recepción de los datagramas a través de arquitecturas TCP/IP.
- Su orientación académica, hacen que sirva de ayuda para el diseño y comprensión del funcionamiento de redes de datos.
- Sirve como complemento de los fundamentos teóricos sobre arquitecturas por niveles, protocolos de enlace y arquitecturas TCP/IP.

2.12.5 Desventajas

- En la versión actual, la interfaz de usuario está implementada con un conjunto de clases, las cuales deben ejecutarse en el equipo del usuario, cada vez que se desee trabajar con éste programa.
- Se deben descargar varios archivos para poder instalar el programa. Además se debe tener especial cuidado en descargar las versiones que se especifican ya que otras versiones de dichos paquetes, no permitirán que se complete la instalación.
- Para el diseño y comprobación del encaminamiento en redes de datos a nivel comercial o para fines de investigación y desarrollo, se debe programar en Java.

Capítulo III: Desarrollo de ejercicios prácticos para comparar el desempeño de los software Opnet Modeler y Cisco Packet Tracer.

Capítulo III: Desarrollo de ejercicios prácticos para comparar el desempeño de los software Opnet Modeler y Cisco Packet Tracer.

3.1 Introducción

Debido a la importancia que tiene para la sociedad de la información y el conocimiento las redes de comunicaciones, resulta, no sólo interesante sino también necesario, aprovechar las bondades que brindan los software de simulación, en la enseñanza de los fundamentos teóricos de las redes, en la investigación de nuevas arquitecturas, y en el desarrollo de nuevos protocolos y aplicaciones de red.

El objetivo fundamental de los ejercicios prácticos que se describen a continuación, es analizar las principales características de los software utilizados y mostrar las posibilidades que cada uno por su lado ofrece a los usuarios, y finalmente, llevar a cabo una comparación entre ellos.

3.2 Ejercicio práctico en Opnet Modeler

Para el desarrollo de este ejercicio práctico se escogió un ejemplo de una topología de red frame relay, y se definió un escenario (escenario1) compuesto por los siguientes elementos de red del Opnet:

- 1 estación de trabajo Ethernet (modelo ethernet_wkstn, nombre cliente1).
- 1 servidor Ethernet (modelo ethernet_server, nombre Servidor1).
- 2 routers Ethernet-Frame Relay (modelo fr4_ethernet2_gtway, nombres router1 y router2)
- 1 nube Frame Relay, totalmente transparente (modelo fr32_cloud, nombre Red FR).
- 2 enlaces 100 BaseT (de router1 a cliente1 y de router2 a Servidor1).
- 2 enlaces FR E1(de Red FR a router1 y de Red FR a router2)
- 1 circuito virtual permanente Frame Relay (modelo fr_pvc, de router1 a router2)

Capítulo III: Desarrollo de ejercicios prácticos para comparar el desempeño de los software Opnet Modeler y Cisco Packet Tracer.

- Los nodos de aplicaciones, perfiles y la configuración de PVCs

La topología lógica es la mostrada en la figura 3.2, que aparece a continuación.

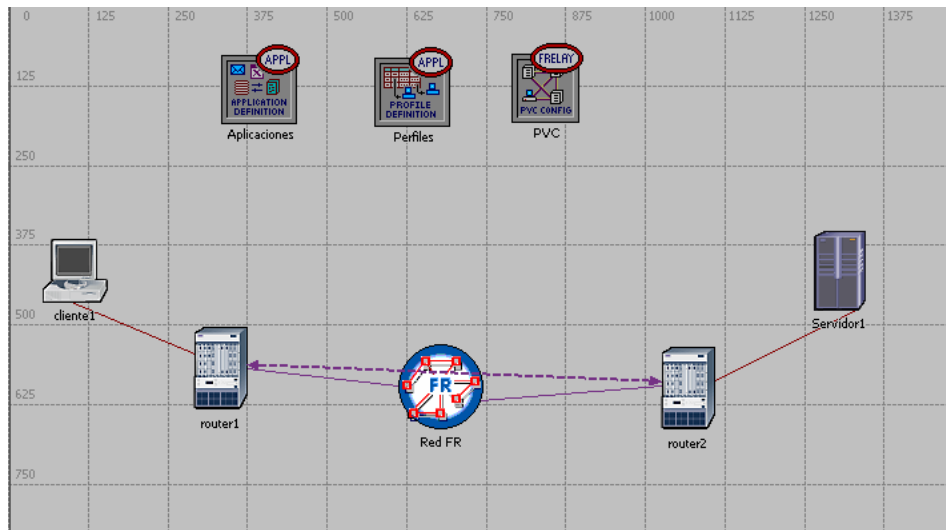


Fig. 3.1. Topología lógica de la actividad práctica desarrollada en Opnet.
Fuente: el autor

La aplicación escogida para la simulación es de base de datos, por lo que se debe configurar la aplicación y el perfil y los nodos respectivos (Aplicaciones y Perfiles), tal como se muestra en las siguientes figuras:

Capítulo III: Desarrollo de ejercicios prácticos para comparar el desempeño de los software Opnet Modeler y Cisco Packet Tracer.

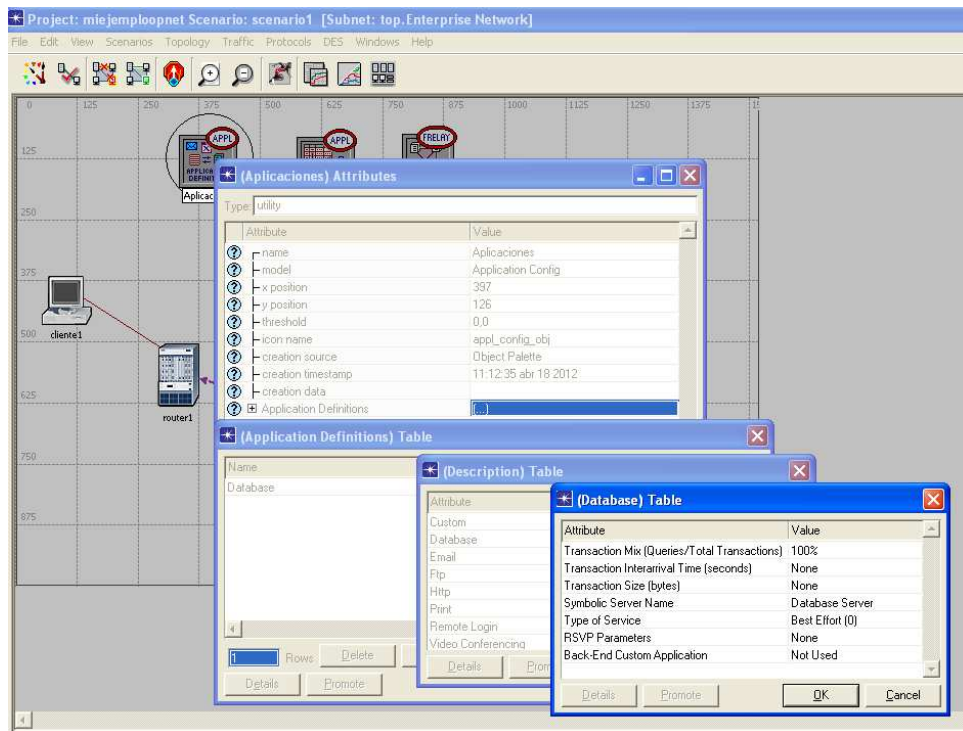


Fig. 3.2. Configuración de la aplicación de base de datos para la simulación.

Fuente: el autor

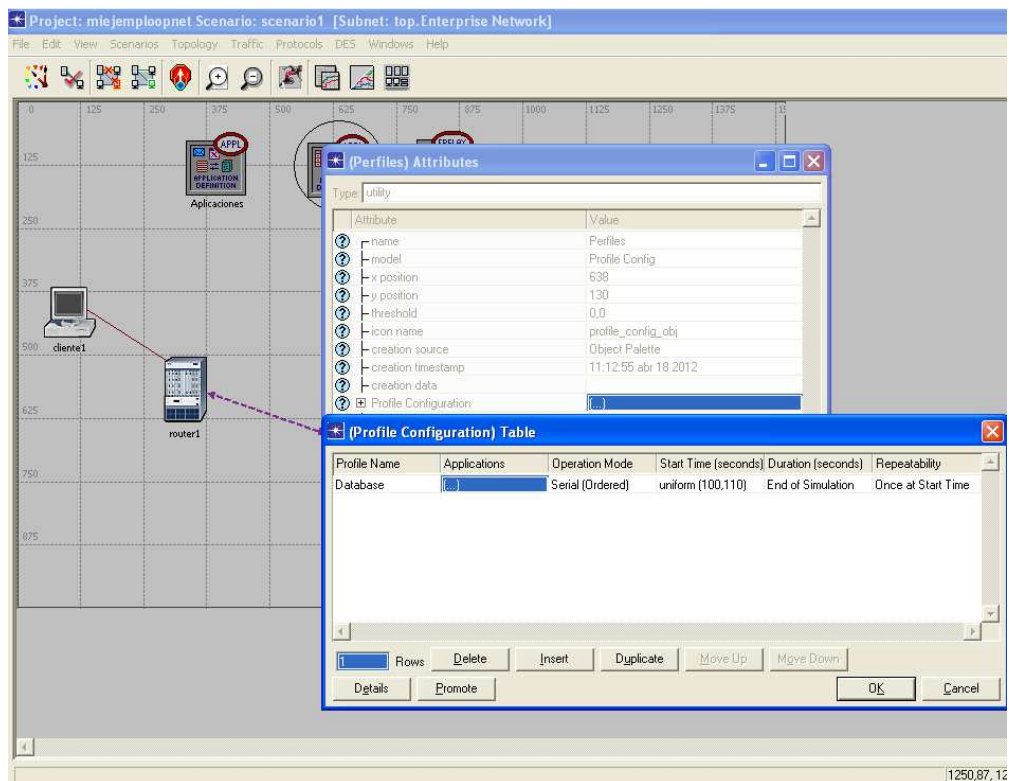


Fig. 3.3. Configuración del perfil para la simulación.

Fuente: el autor

Capítulo III: Desarrollo de ejercicios prácticos para comparar el desempeño de los software Opnet Modeler y Cisco Packet Tracer.

Posteriormente se configura un circuito virtual permanente PVC entre los routers 1 y 2, tal como se muestra en la figura siguiente:

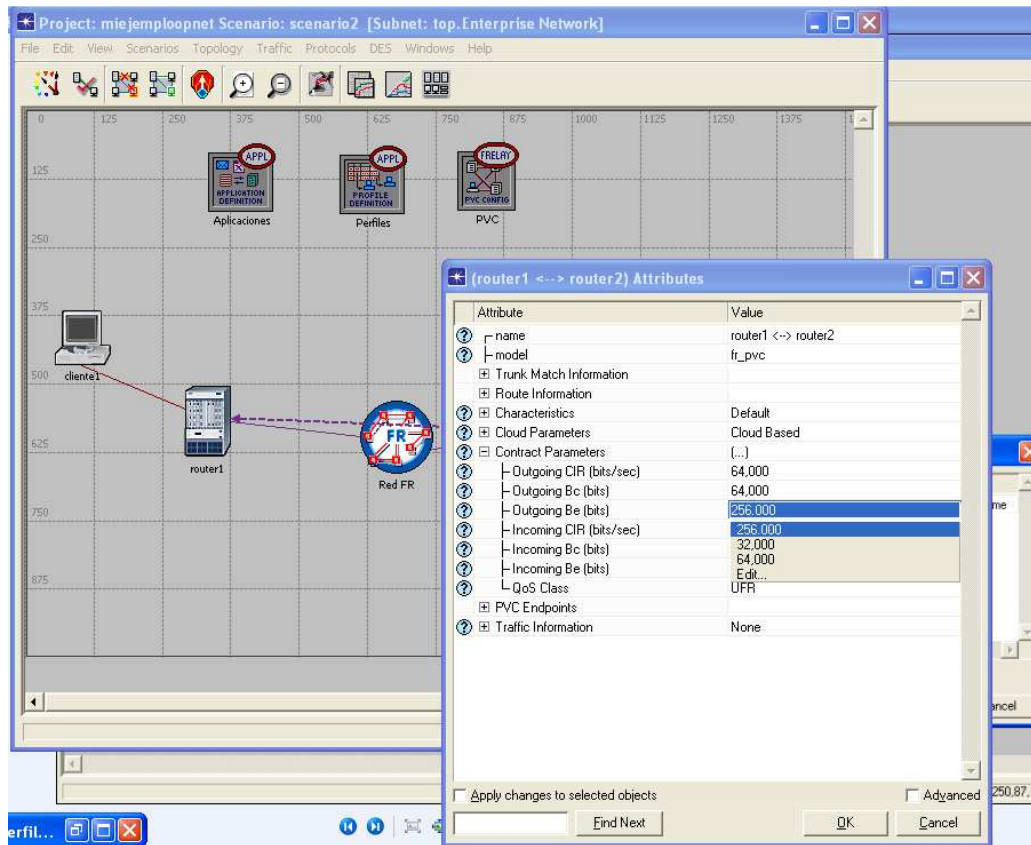


Fig. 3.4. Configuración del circuito virtual permanente (PVC).

Fuente: el autor

Luego se configura cliente1 y Servidor1 para que soporten la aplicación de base de datos, tal como se muestra en las figuras 3.5 y 3.6, que aparecen a continuación:

Capítulo III: Desarrollo de ejercicios prácticos para comparar el desempeño de los software Opnet Modeler y Cisco Packet Tracer.

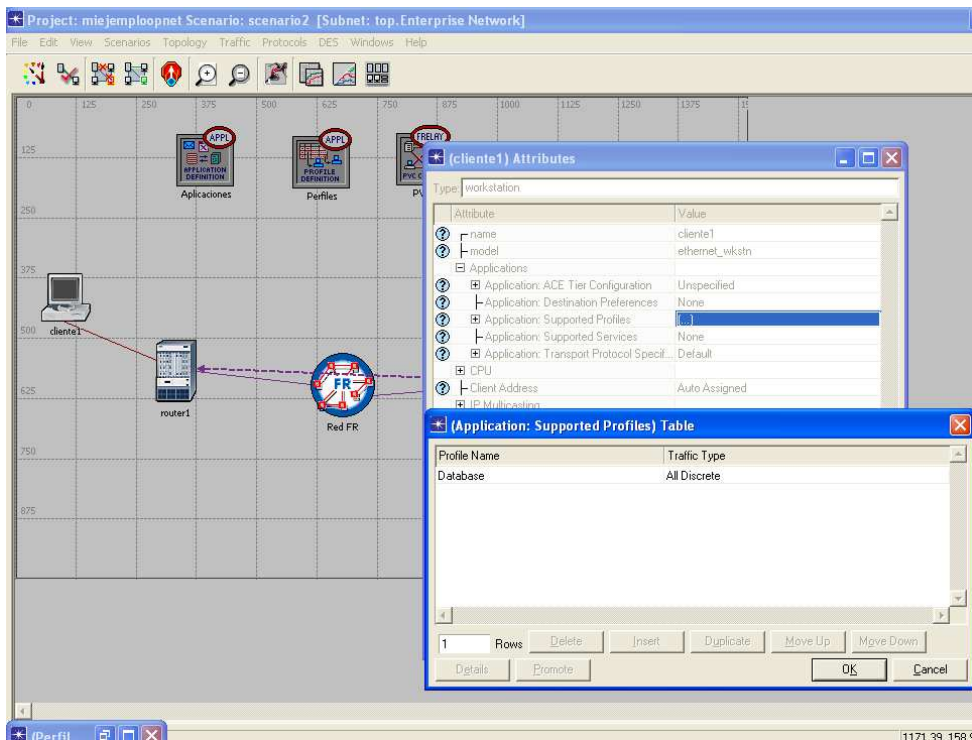


Fig. 3.5 Configuración cliente1.
Fuente: el autor

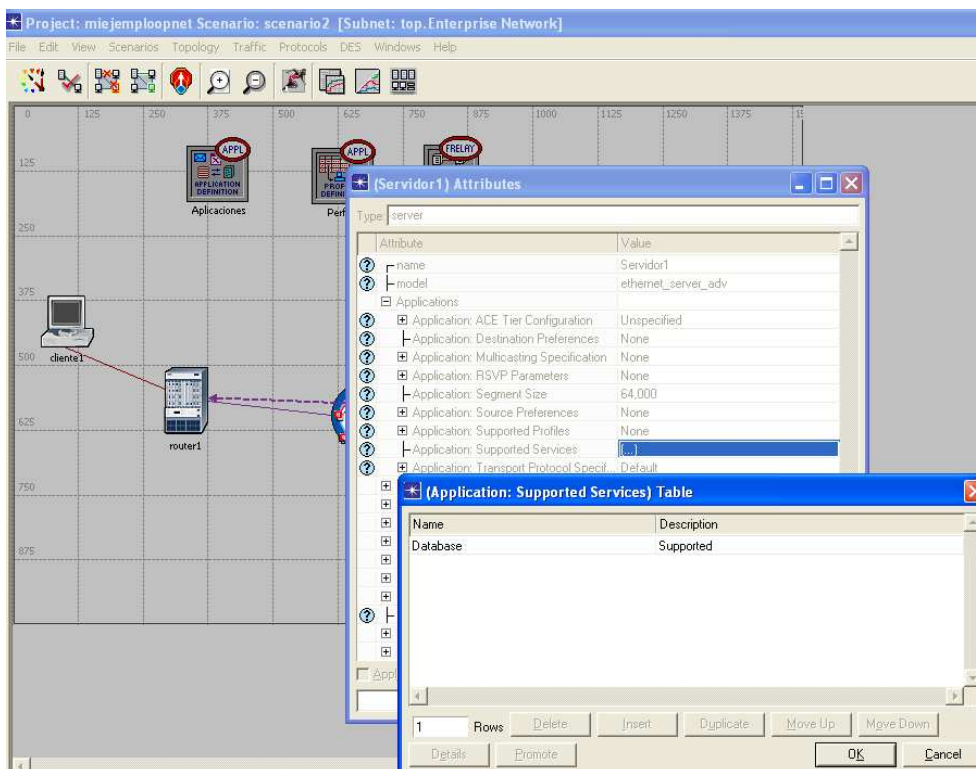


Fig. 3.6. Configuración Servidor1.
Fuente: el autor

Capítulo III: Desarrollo de ejercicios prácticos para comparar el desempeño de los software Opnet Modeler y Cisco Packet Tracer.

Una vez configurados todos los dispositivos necesarios, se procede a seleccionar las estadísticas de la simulación, algunas de las cuales se muestran en las figuras:

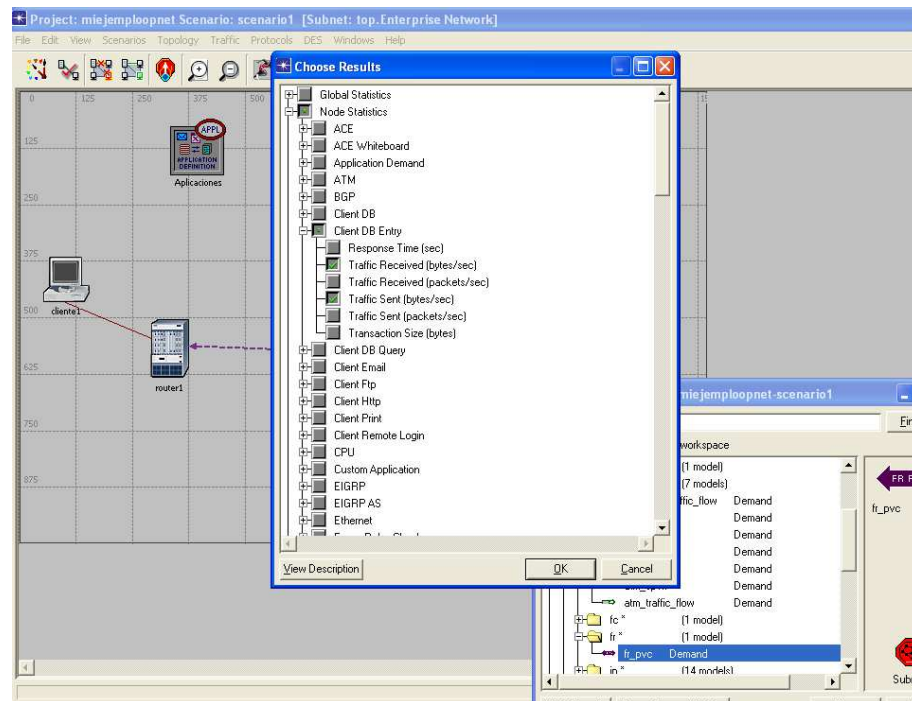


Fig. 3.7. Selección de estadísticas para la simulación (tráfico enviado y recibido por el cliente1).

Fuente: el autor

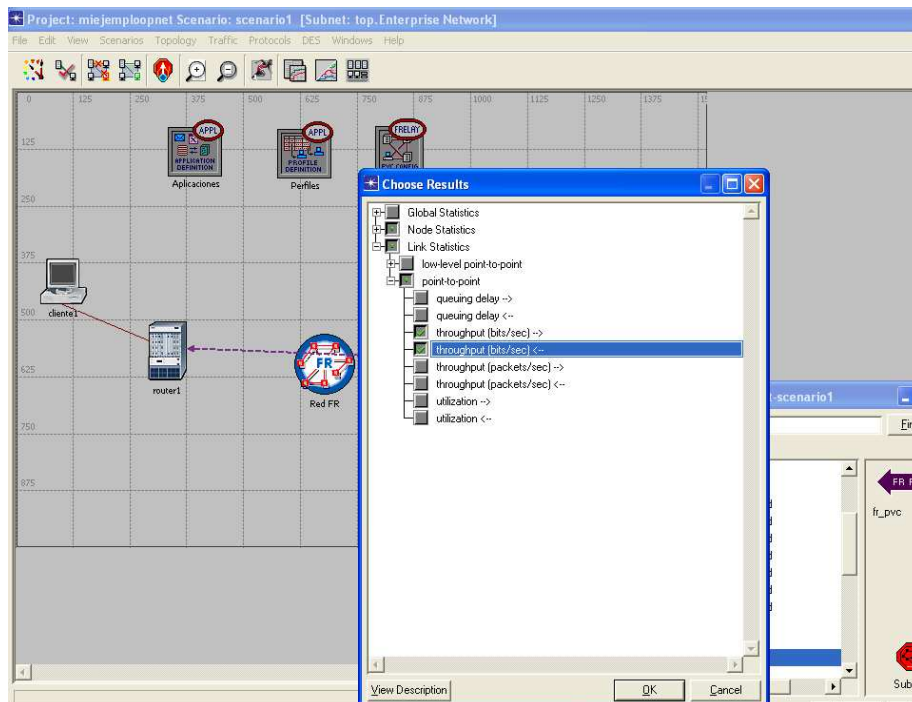


Fig. 3.8. Selección de estadísticas para la simulación (cargabilidad de la red).

Fuente: el autor

Capítulo III: Desarrollo de ejercicios prácticos para comparar el desempeño de los software Opnet Modeler y Cisco Packet Tracer.

Visualización de los resultados

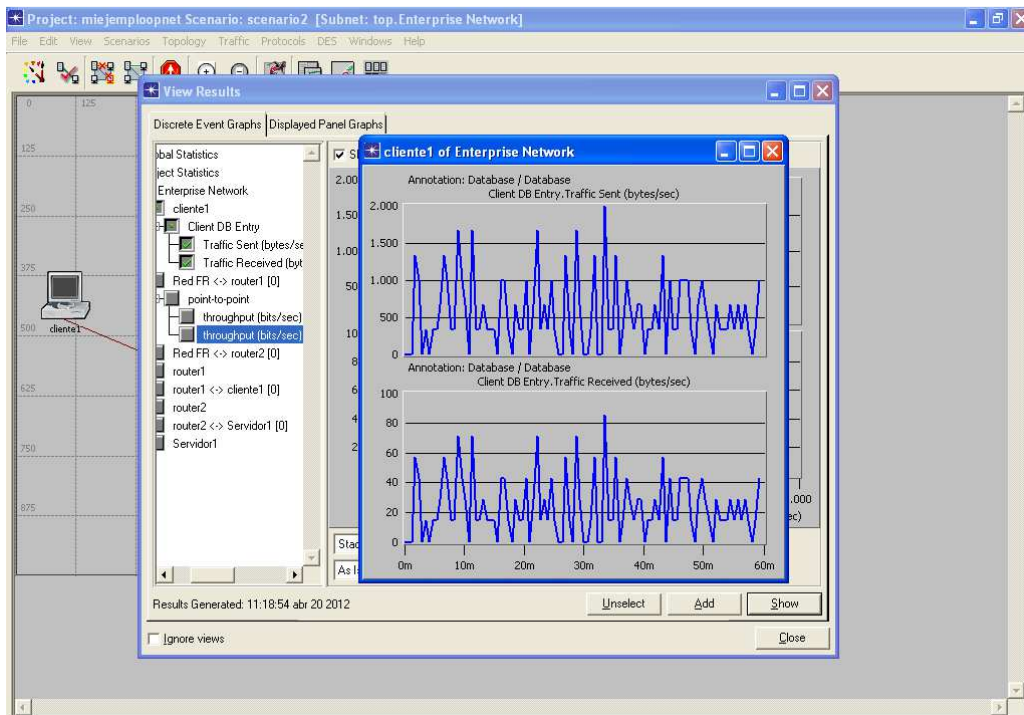


Fig. 3.9. Gráfico del tráfico enviado y recibido por el cliente1.

Fuente: el autor

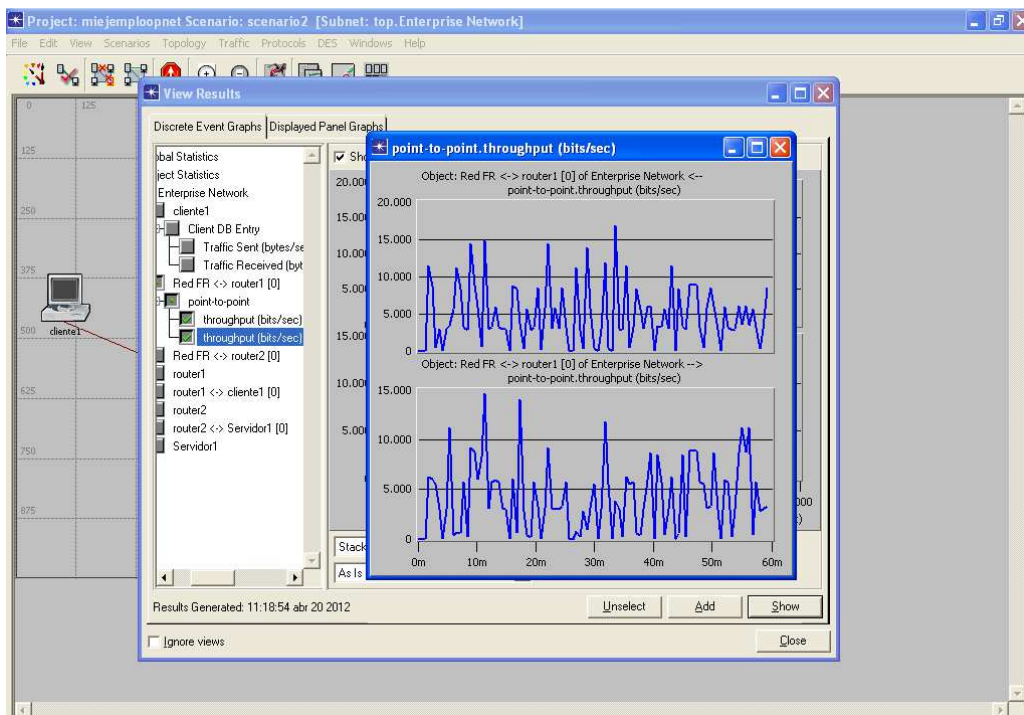


Fig. 3.10 Gráfico del tráfico enviado y recibido por el cliente1.

Fuente: el autor

3.3 Ejercicio práctico en Packet Tracer

Esta actividad se tomó del curso CCNA Exploration de Cisco y se enfoca en la creación de una topología jerárquica, desde el núcleo hasta las capas de distribución y de acceso, partiendo de una topología inicial, brindada por la propia actividad, la cual se muestra en la figura 3.11, mostrada a continuación:

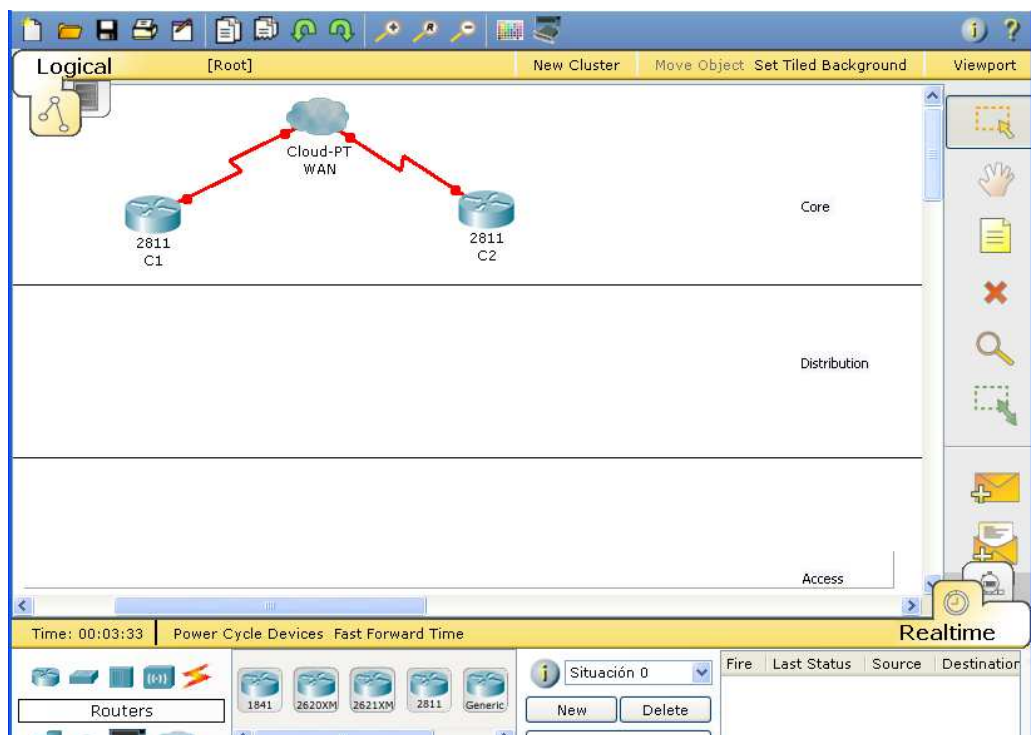


Fig. 3.11 Topología lógica a completar en la actividad propuesta.

Fuente: el autor

El primer paso orientado en la actividad es agregar los routers faltantes en la capa de distribución y configurarles sus nombres. Éstos son:

- 2 routers modelo 1841, nombres R1 y R3.
- 1 router modelo 2621XM, nombre R3.

Capítulo III: Desarrollo de ejercicios prácticos para comparar el desempeño de los software Opnet Modeler y Cisco Packet Tracer.

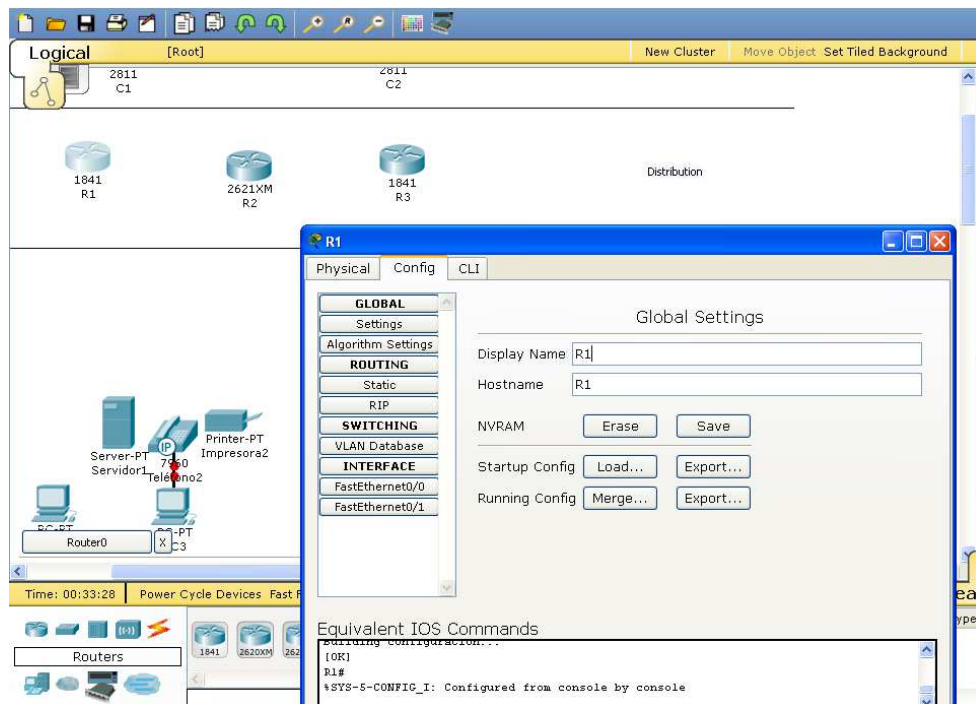


Fig. 3.12 Configuración del nombre de los routers de la capa de distribución.

Fuente: el autor

El próximo paso es agregar los switches de la capa de acceso, tal como se muestra en la figura. Éstos son:

- 9 switches modelo 2960-24TT, nombres S1, S2.....S9.

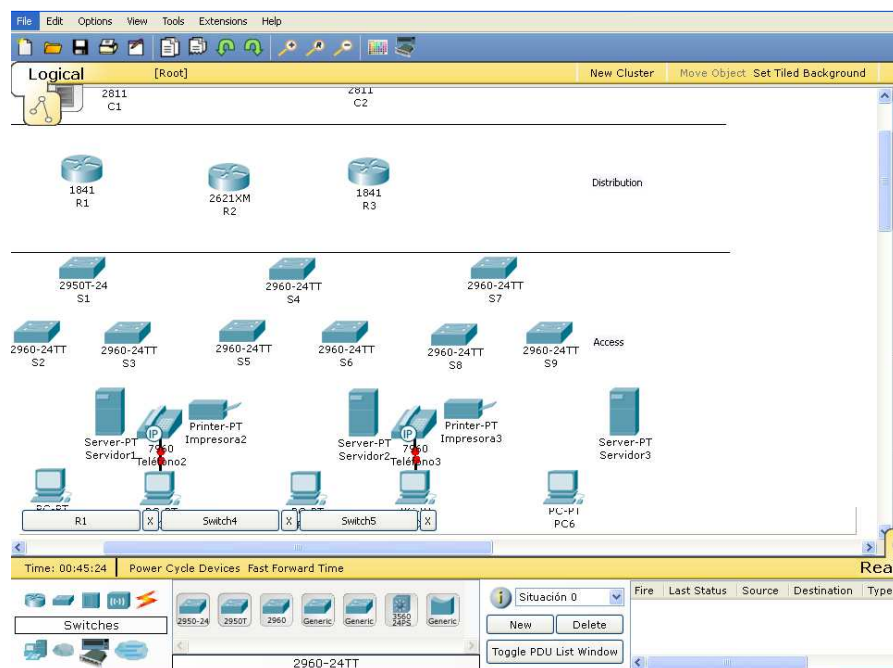


Fig. 3.13 Switches agregados en la capa de acceso.

Fuente: el autor

Capítulo III: Desarrollo de ejercicios prácticos para comparar el desempeño de los software Opnet Modeler y Cisco Packet Tracer.

Luego se van realizando las conexiones con los cables recomendados (fig. 3.14):

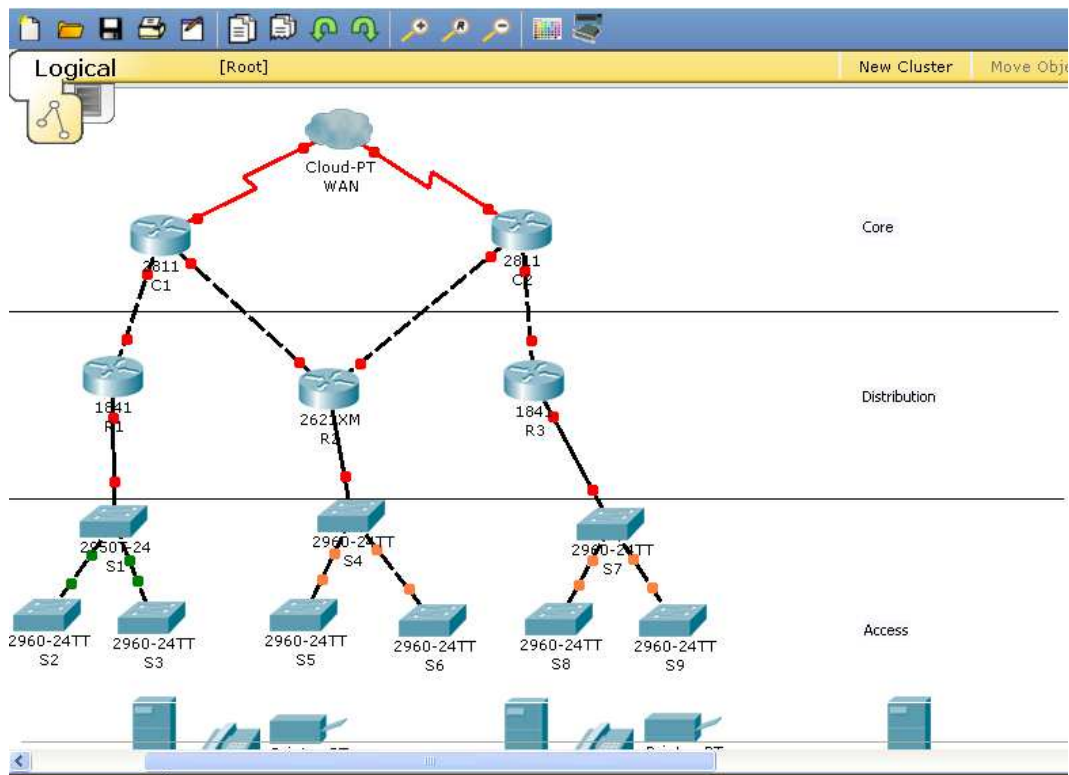


Fig. 3.14. Conexiones realizadas en la capa de distribución y de acceso.

Fuente: el autor

Finalmente, al conectar los dispositivos finales, la topología de la red queda como se muestra en la figura 3.15:

Capítulo III: Desarrollo de ejercicios prácticos para comparar el desempeño de los software Opnet Modeler y Cisco Packet Tracer.

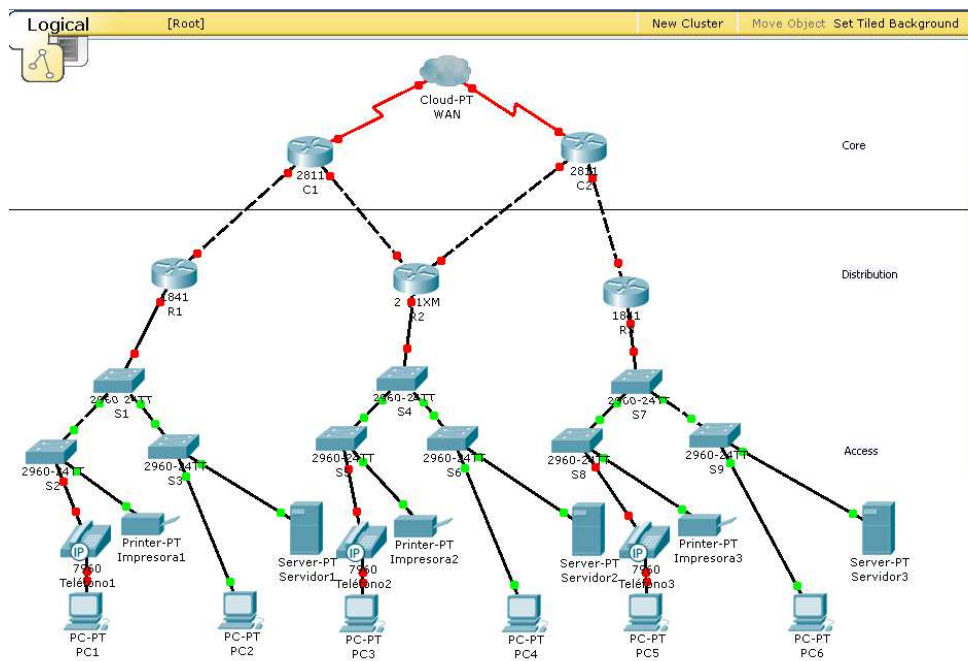


Fig. 3.15 Configuración final de la red jerárquica.

Fuente: el autor

Una vez realizada la actividad, se verifican los resultados en una interfaz como la mostrada en la figura 3.16:

Assessment Items		Status	Points	Component(s)	Feedback
Network					
C1					
Ports					
FastEthernet0/0					
Link to R1					
Connects to FastEther...	Correct	0	Other		
Type	Correct	0	Other		
FastEthernet0/1					
Link to R2					
Connects to FastEther...	Correct	0	Other		
Type	Correct	0	Other		
C2					
Ports					
FastEthernet0/0					
Link to R2					
Connects to FastEther...	Correct	0	Other		
Type	Correct	0	Other		
FastEthernet0/1					
Link to R3					
Connects to FastEther...	Correct	0	Other		
Type	Correct	0	Other		
R1					
Host Name	Correct	0	Other		
Ports					
FastEthernet0/0					
Link to C1					
Connects to FastEther...	Correct	0	Other		
Type	Correct	0	Other		
FastEthernet0/1					
Link to S1					
Connects to FastEther...	Correct	0	Other		

Fig. 3.16 Verificación de los resultados.

Fuente: el autor

Capítulo III: Desarrollo de ejercicios prácticos para comparar el desempeño de los software Opnet Modeler y Cisco Packet Tracer.

3.4 Comparación de los software Opnet Modeler y Packet Tracer.

	Packet Tracer	Opnet Modeler
Interfaz gráfica de usuario	<p>Ofrece una interfaz basada en ventanas que le brinda al usuario muchas facilidades para el modelado, la descripción, la configuración y la simulación de redes.</p> <p>El programa posee una interfaz de usuario muy fácil de manejar, e incluye documentación y tutoriales sobre el manejo del mismo.</p> <p>Interfaz más amigable que Opnet Modeler</p>	<p>Ofrece una interfaz basada en una serie de editores organizados jerárquicamente, los cuales permiten diseñar y configurar los modelos de red, de nodos y procesos en las topologías que se van a simular con este programa. Los editores trabajan en forma directa y paralela a la estructura real de la red, los equipos y los protocolos.</p> <p>Interfaz menos intuitiva que la del Packet Tracer</p>
Aplicabilidad, Escalabilidad, Configurabilidad, y Flexibilidad	<p>Permite ver el desarrollo por capas del proceso de transmisión y recepción de paquetes de datos de acuerdo con el modelo de referencia OSI. Permite la simulación del protocolo de enrutamiento RIP V2 y la ejecución del protocolo STP y el protocolo SNMP para realizar diagnósticos básicos a las conexiones entre dispositivos del modelo de la red.</p> <p>Por otra parte :</p> <p>Solo permite modelar redes en términos de filtrado y retransmisión de paquetes No permite crear topologías de red que involucren la implementación de tecnologías diferentes a Ethernet.</p> <p>Es decir, con este programa no se pueden implementar simulaciones con tecnologías de red como Frame Relay, ATM, XDSL, Satelitales, telefonía celular entre otras.</p>	<p>Opnet incluye las librerías para acceder a un extenso grupo de aplicaciones y protocolos como: HTTP, TCP, IP, OSPF, BGP, EIGRP, RIP, SVP, Frame Relay, FDDI, Ethernet, ATM, LANs 802.11 (Wireless), aplicaciones de voz, MPLS, PNNI, DOCSIS, UMTS, IP Multicast, Circuit Switch, MANET, IP Móvil, VoIP, IPv6 entre otras.</p> <p>Tiene interfaces para visualización del modelo en 3D. Los APIs de simulación permiten acceder libremente al código fuente, lo cual facilita la programación de nuevos protocolos de red. Las librerías de modelos de red estándar incluyen dispositivos de red tanto comerciales como genéricos. Modelos de red jerárquicos. Maneja topologías de red complejas con subredes anidadas ilimitadas. Permite mostrar el tráfico por la red a través de una animación, durante y después de la simulación. Los resultados se exhiben mediante gráficos estadísticos.</p> <p>Es mucho más aplicables, escalable, configurable, y flexible que Packet Tracer.</p>
Tiempo de aprendizaje	El tiempo de aprendizaje es relativamente corto en comparación con Opnet.	El tiempo de aprendizaje es elevado, ya que su alcance es mayor.
Tiempo de desarrollo	Pequeño	Un poco mayor que Packet Tracer, ya que posee más opciones de configuración Packet Tracer.
Tiempo de ejecución	Pequeño	Medio

Capítulo III: Desarrollo de ejercicios prácticos para comparar el desempeño de los software Opnet Modeler y Cisco Packet Tracer.

Utilidad en ambientes educativos	Muchas posibilidades, ya que el Packet Tracer está enfocado fundamentalmente a la enseñanza.	Tiene grandes posibilidades, aunque menos que Packet Tracer, ya que Opnet no está enfocado a la enseñanza.
Requerimientos	<p>Mínimo:</p> <ul style="list-style-type: none"> • CPU: Intel Pentium III 500 MHz o equivalente • Sistema Operativo: Microsoft Windows XP, Microsoft Windows Vista, Microsoft Windows 7, Fedora 11, o Ubuntu 8.04 LTS • RAM: 250 MB libres • Almacenamiento: 250 MB de espacio libre en disco • Resolución del display: 800x600 • Adobe Flash Player • Los últimos controladores de la tarjeta de video y actualizaciones del sistema operativo. 	<p>Mínimo:</p> <ul style="list-style-type: none"> • Sistema Operativo: Windows NT, 2000, XP, Unix • CPU: Intel Pentium III 500MHz • RAM: 64 MB libres • Almacenamiento: 100 MB de espacio libre en disco • Resolución del display: SVGA • 8 MB en memoria de video y tarjeta de sonido.
Disponibilidad	Es software propietario de Cisco, por lo que hay que adquirir licencias para su instalación.	Es software propietario por el que hay que adquirir licencias para su instalación, Aunque, para entornos universitarios existen licencias gratuitas limitadas, pero si se requiere soporte técnico o algún módulo adicional, se debe adquirir la correspondiente licencia de pago. Los modelos que utiliza OPNET son de código abierto ya que tienes acceso al código para implementar nuevas funciones. Las fuentes del kernel del simulador, en cambio, no son públicas.

Tabla 3.1. Comparación Opnet Modeler y Packet Tracer.

Fuente: el autor

Conclusiones

Para dar cumplimiento al objetivo planteado en este trabajo:

- Se realizó un profundo estudio de los conceptos importantes de la simulación de sistemas.
- Se estudiaron las características fundamentales de los simuladores de redes.
- Se realizó un estudio y caracterización de algunos de los simuladores de redes más conocidos actualmente en el mercado del software, haciendo especial énfasis en los software Opnet Modeler y Packet Tracer, ya que sus características resultaron relevantes durante los estudios realizados.
- Se desarrolló un ejercicio práctico con el software Opnet Modeler para evaluar su desempeño.
- Se desarrolló un ejercicio práctico con el software Packet Tracer para evaluar su desempeño.
- Finalmente, se realizó un estudio comparativo entre los software utilizados en los ejercicios: Opnet Modeler y Packet tracer, analizando, mediante una tabla, las ventajas y desventajas de uno con respecto al otro.

Recomendaciones

Se recomienda que:

- Se utilice este trabajo como herramienta de apoyo y consulta por los docentes de la asignatura Telemática I, de manera que ellos puedan escoger, entre los software analizados, el más adecuado para realizar las prácticas de laboratorio en su asignatura.
- Se evalúe fundamentalmente, el uso de Opnet Modeler o Packet Tracer como software de apoyo para la asignatura Telemática.
- Previo a la selección de uno de estos software, se realicen actividades prácticas de prueba con estudiantes, de manera que después se les pueda realizar una encuesta que permita evaluar realmente cuál de estas herramientas sería la más adecuada.

Bibliografía

- [1] Ayuda de Opnet Modeler
- [2] Ayuda de Packet Tracer
- [3] Borrero Pardo, Diego Fernando “Simulación de algoritmo handoff vertical basado en preferencias de usuario mediante la herramienta software libre NCTuns.2009”, Disponible en <http://bdigital.uao.edu.co/bitstream/10614/332/1/T0003563.pdf>,>. [Consulta: diciembre 2012]
- [4] Cabrero Sergio, Pañeda Xabiel G., García Roberto, Melendi David, Orea Rafael, “Herramienta educativa para el diseño y configuración de redes de comunicaciones”. Disponible en <http://www.romulo.det.uvigo.es/revista/RITA/site/201202/uploads/IEE-E-RITA.2010.V5.N1.A3.pdf>> [Consulta: abril 2012]
- [5] Castellanos Hernández Wilder Eduardo, Chacón Osorio Mónica Edith, Utilización de herramientas software para el modelado y simulación de redes de comunicaciones, Disponible en <http://scienti.colciencias.gov.co:8084/publindex/docs/articulos/1657-8236/5/7.pdf>>. , [Consulta: febrero 2012].
- [6] Di Lorenzo, P. OPNET mobility simulation models [En línea]. Disponible en: <http://labreti.ing.uniroma1.it/wine/papers/mms.pdf> > [Consulta: 21 noviembre 2011]
- [7] Ergen, M. UC Berkeley. OPNET. [En línea] Disponible en: <http://www.eecs.berkeley.edu/~ergen/docs/OPNET.pdf> [Consulta: 21 noviembre 2011]
- [8] Graham Robert, Gray Clifford, “Business Games Handbook”, American Management Association, 1969. [Consulta: noviembre 2011]
- [9] Griffiths, A. OPNET Tutorials [En línea] Disponible en: [http://www.soc.staffs.ac.uk/alg1/2004_5/Semester_1/Network%20Systems%20&%20Technologies,%20NSTs1%20\(CE00227-M\)/NST/msc_tutorial.html](http://www.soc.staffs.ac.uk/alg1/2004_5/Semester_1/Network%20Systems%20&%20Technologies,%20NSTs1%20(CE00227-M)/NST/msc_tutorial.html)>[Consulta: noviembre 2011]
- [10] Gómez Márquez Andrés “Protocolos de Enrutamiento Simulador de Tráfico de Redes.2008”, Disponible en <http://www.monografías.com/trabajos-pdf/enrutamiento-trafico-redes/enrutamiento-trafico-redes.pdf>>[Consulta: abril 2012]
- [11] “Herramientas de simulación”. Disponible en <http://es.scribd.com/doc/49911247/herramientas-de-Simulacion> [Consulta: enero 2012].

- [12] Information Sciences Institute, “The network simulator: ns-2”, Disponible en http://nslam.isi.edu/nslam/index.php/Main_Page, [Consulta: octubre 2011].
- [13] Integración en OMNeT++ de un estimador eficiente de medias como mecanismo de control del fin de la simulación. Disponible en http://e-archivo.uc3m.es/bitstream/1016/6970/2/PFC_Ivan_garcia_moreno.pdf >[Consulta: febrero 2012].
- [14] L. Joyanes Aguilar. La Carrera Mundial por el Conocimiento. Una Visión desde la Nueva Economía. Separata de Corintios XIII, núm. 96, Madrid-España, 2000.
- [15] Nesmachnow Sergio, Robledo Franco, Hartmam Raúl , “Relevamiento de simuladores 3G – UMTS.2008”, Disponible en <http://www.fing.uy/inco/pedeciba/bibliote/reptec/TR0813>>. [Consulta: marzo 2012].
- [16] Pellicer Corbacho, Rafael, “Estudio comparativo de la implementación de los protocolos RTP/RTCP en los simuladores Opnet Modeler y Network Simulator 2”. Disponible en <http://riunet.upv.es/bitstream/handle/10251/9917/PFC.pdf> [Consulta: diciembre 2011].
- [17] Pulgarín Velez Marcela, “KIVA-NS Modulo RIP v1” Disponible en <http://bdigital.eafit.edu.co/Proyecto/P004.65CDV436/marcoteorico.pdf> [Consulta:enero 2012].
- [18] Ruiz López Luis Daniel, Oliveras Pla Marc, “Proyecto Fin de Carrera: Redes mesh basadas en puntos de acceso inteligentes 802.11 open source (III).2006”, Disponible en <http://www.upcommons.upc.edu/pfc/bitstream/2099.1/3797/1/53876-1.pdf>. >[Consulta: marzo 2012].
- [19] Tanenbaum A. S. Redes de Computadoras. Prentice Hall Hispanoamericana S. A., México-México, 1997.
- [20] “The ns-3 network simulator”, Disponible en <http://www.nslam.org/>>, [Consulta: marzo 2012].
- [21] “Ventajas y desventajas de la simulación” Disponible en <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r20742.DOC>.> [Consulta: diciembre 2011].
- [22] Vera Castellón Isabel “Simulación de Redes de Computadores aplicado a docencia”. Disponible en <http://nsl.csie.nctu.edu.tw/NCTUnsReferences/memoria.pdf> > [Consulta: octubre 2011].

Bibliografía

- [23] W. Stallings. Data and Computer Communications -Fifth Edition.
Prentice Hall, NJ-USA, 1997.

Anexo 1. Routers modelados en Packet Tracer



Fig. A.1: Router 1841 de Cisco
Fuente: [2]



Fig A.2: Router 2620XM de Cisco
Fuente: [2]



Fig A.3: Router 2621XM de Cisco
Fuente: [2]

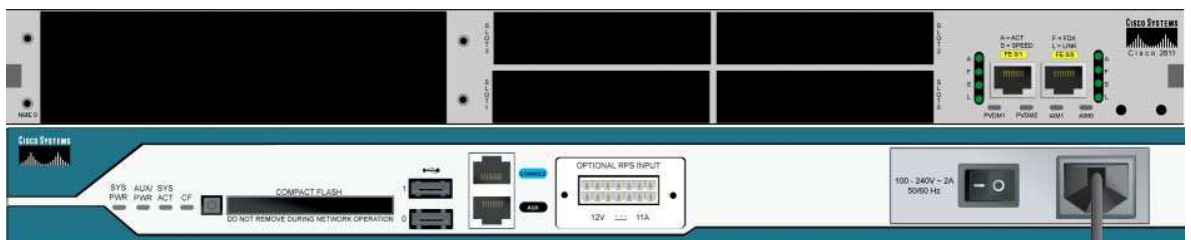


Fig. A.4: Router 2811 de Cisco
Fuente: [2]

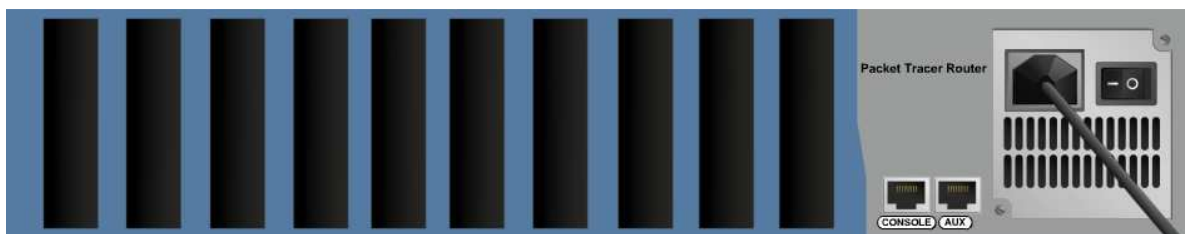


Fig. A.5: Router PT genérico
Fuente: [2]

Anexo 2. Switches modelados en Packet Tracer

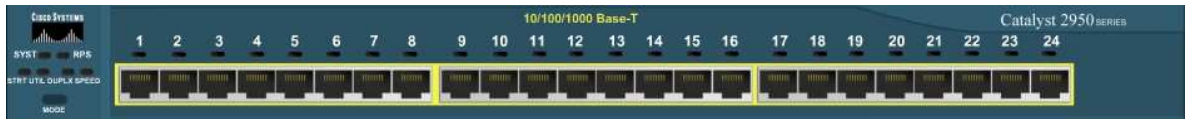


Fig A.6: Switch 2950-24 Cisco Catalyst
Fuente: [2]

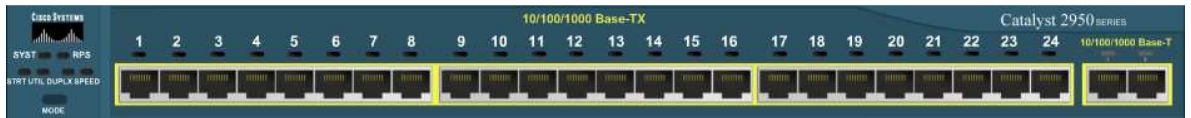


Fig A.7: Switch 2950T-24 Cisco Catalyst
Fuente: [2]

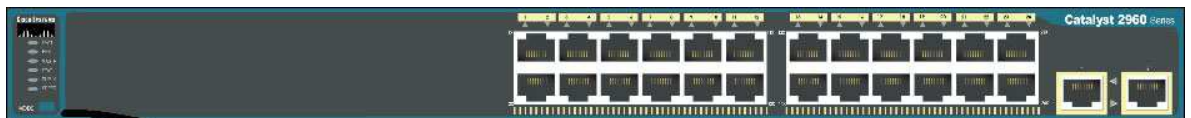


Fig A.8: Switch 2960-24TT Cisco Catalyst
Fuente: [2]



Fig. A.9: Switch PT genérico
Fuente: [2]

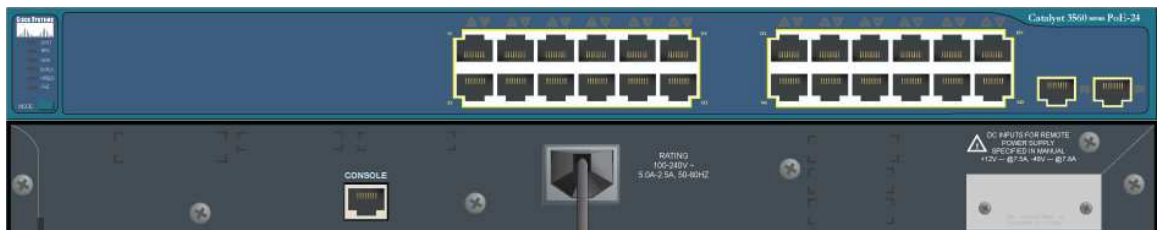


Fig A.10: Switch 3560-24PS
Fuente: [2]

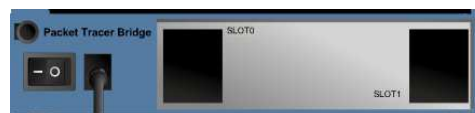


Fig. A.11: Bridge-PT genérico
Fuente: [2]

Anexo 3. Dispositivos finales modelados en Packet Tracer



Fig. A.12: PC-PT genérico
Fuente: [2]



Fig. A.13: Laptop-PT genérico
Fuente: [2]

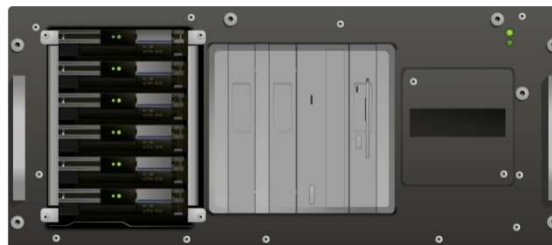


Fig. A.14: Server-PT genérico
Fuente: [2]

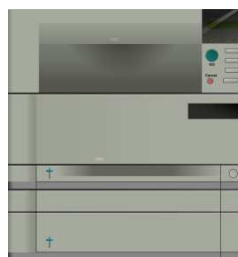


Fig. A.15: Printer-PT genérico
Fuente: [2]



Fig. A.16: Teléfono IP 7960
Fuente: [2]

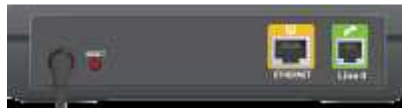


Fig A.17: Home-VoIP-PT
Fuente: [2]



Fig A.18: Analog-Phone-PT
Fuente: [2]



Fig A.19: TV-PT
Fuente: [2]



Fig. A.20: TabletPC-PT
Fuente: [2]



Fig. A.21: PDA-PT
Fuente: [2]



Fig A.22: WirelessEndDevice-PT
Fuente: [2]



Fig A.23: WiredDevice-PT
Fuente: [2]

Anexo 4. Otros dispositivos modelados en Packet Tracer



Fig. A.24: Hub-PT
Fuente: [2]



Fig. A.25: Repeater-PT
Fuente: [2]



Fig A.26: CoAxialSplitter-PT
Fuente: [2]



Fig A.27: AccessPoint-PT
Fuente: [2]



Fig A.28: AccessPoint-PT-A
Fuente: [2]



Fig A.29: AccessPoint-PT-N
Fuente: [2]



Fig A.30: Linksys-WRT300N
Fuente: [2]

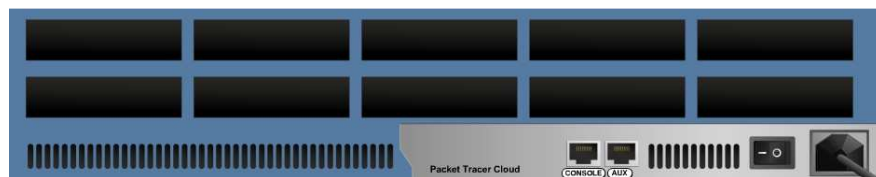


Fig. A.31: Cloud-PT
Fuente: [2]



Fig. A.32: DSL-Modem-PT
Fuente: [2]



Fig. A.33: DSL-Modem-PT
Fuente: [2]