

## CAPITULO 3

### OMNET++

A fin de realizar la simulación de la red de comunicación establecida para analizar el desempeño del IDS diseñado, se empleó el programa OMNET++. Este capítulo presenta a este programa como una herramienta para la simulación no sólo de una red ante un ambiente hostil, sino también para simular la función que desempeñará el IDS en la red presentada para dicha simulación.

#### 3.1 Descripción de OMNET++

OMNET++ se caracteriza por ser un simulador modular de eventos discretos de redes orientado a objetos, es decir las variables de estado cambian sólo en un conjunto de puntos en el tiempo, usado habitualmente para modelar el tráfico de redes de telecomunicaciones, protocolos, sistemas multiprocesadores y distribuidos, validación de arquitecturas hardware, evaluación del rendimiento de sistemas software y, en general, modelar cualquier sistema que pueda simularse con eventos discretos.

Cuando se habla de la característica modular, se refiere a que cada elemento que se agregue a la red diseñada funciona como módulo, el cual contiene submódulos que se denominan módulos compuestos. Existen también los módulos simples (elementos básicos) que contienen los algoritmos que permiten el funcionamiento del modelo. Ambos tipos de módulos se implementan en el lenguaje C++ usando la biblioteca de simulación de OMNET++ [16].

Esta herramienta de simulación esta disponible tanto para sistemas operativos basados en UNIX como para Windows y se distribuye bajo la Licencia Pública Académica. Su versión comercial, denominada OMNEST ha sido desarrollada por Simulcraft Inc. El desarrollo de este software se enfocó en el objetivo de ser un programa que permitiera al usuario simular modelos de redes de comunicación a gran escala con periodos de procesamiento rápidos y con recursos no limitados para la ejecución de distintos eventos que se puedan simular en la red [16].

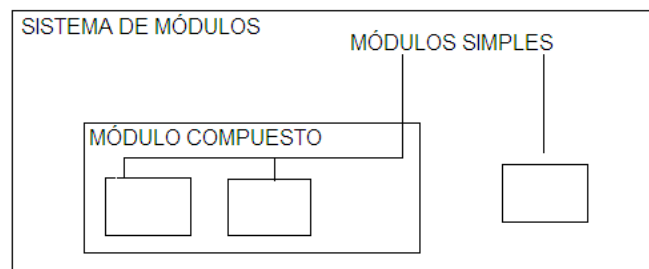
### 3.1.1 Elementos de simulación de OMNET++

Con el propósito de ofrecer un ambiente que facilite el desarrollo de las simulaciones de redes de comunicaciones, OMNET++ cuenta con una interfaz gráfica la cual se compone de los siguientes elementos:

- Librerías de Simulación.
- Un Editor Gráfico de Redes (*GNED*) que emplea el lenguaje NED.
- Compilador para el lenguaje Editor de Redes, *NED (Network Editor)*.
- Interfaz de Comando por Línea (*Cmdenv*) empleado para la ejecución de la simulación.
- Una Interfaz Gráfica de Usuario, *GUI (Graphical User Interface)*, para la ejecución de la simulación de nombre *Tkenv*.
- Herramienta para la graficación de vectores, conocida como *Plove*.
- Herramienta para la graficación de escalares, conocida como *Scalars*.
- Herramienta para la documentación del código.
- Creación de un archivo “*Make*”, entre otros elementos [16].

Un modelo de simulación en OMNET++ consiste de una serie de módulos que se comunican a través del paso de mensajes y en el que los *módulos activos* son módulos simples y cuya implementación es a través de la programación en C++ empleando la clase de la librería de simulación (véase la Figura 3.1). Por otra parte los módulos simples pueden ser agrupados en módulos compuestos, finalmente el usuario crea un sistema de módulos como una instancia previamente definida en el tipo de módulo, es decir el tipo de elemento que compone a la red.

Cuando un módulo es usado como un bloque compuesto por otros módulos, no existe distinción entre ellos lo que permite la transparencia en la comunicación durante la simulación [16].



**Figura 3.1:** Estructura de modelado en OMNET++ [16].

Durante la simulación, los módulos se comunican entre sí mediante el envío de mensajes los cuales pueden tener distintos niveles de complejidad establecidos de acuerdo a las necesidades del usuario. Se pueden mandar estos mensajes de forma directa a un módulo o a través de conexiones y compuertas ya que los módulos poseen sus propios parámetros que determinan el comportamiento del mismo.

Con el propósito de facilitar la construcción o diseño de una simulación OMNET++ cuenta con diversas simulaciones como ejemplos, una vez descargado el programa dichas simulaciones se encuentran en una plataforma llamada *INET* lo que permite al usuario emplear las librerías incluidas en esta plataforma como guías para el desarrollo de una simulación. Brevemente se presentará el lenguaje *NED* y su ambiente gráfico *GNED* utilizado para la construcción gráfica de la red de la simulación para este trabajo de Tesis.

### 3.1.2 Lenguaje NED y Ambiente GNED

El usuario define la estructura del modelo (a través de los módulos y sus conexiones) por medio del lenguaje NED, en OMNET++. Básicamente los elementos para este lenguaje son la declaración de los módulos y la definición de las características de la red. La interfaz de los módulos se compone de compuertas y parámetros. Mientras que la definición de un módulo compuesto se compone de la declaración de la interface externa de módulos con sus respectivos parámetros y compuertas, así como la definición de las conexiones.

El lenguaje NED y OMNET++ fueron diseñados para construir modelos a gran escala, lo que quiere decir que se pueden reutilizar elementos de modelado vía módulo, así como la disponibilidad de distintos elementos para desarrollar una simulación [16].

*NED* establece el comportamiento para los módulos declarados en la red. Habiendo hecho esto y después de haber creado un archivo “.ini” para la configuración de la simulación, se presenta el *GUI (Tkenv)* donde se corre la simulación y se observa el comportamiento de la red. Por otra parte se cuenta con las herramientas de graficación utilizadas para graficar variables escalares y vectoriales que emplea OMNET++ para registrar el desempeño de la red [16].

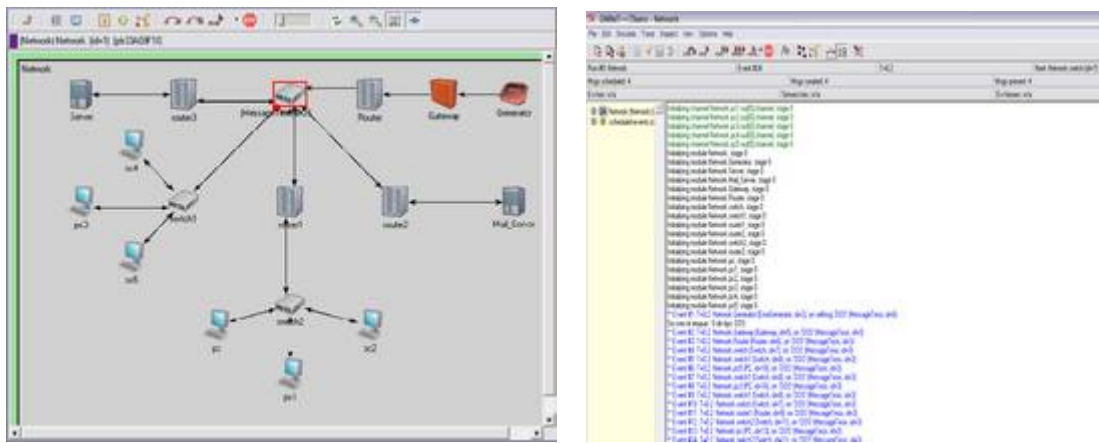
### 3.1.3 Lenguaje C++

Dado que los módulos simples son considerados en OMNeT++ como los componentes activos de la red, son los únicos elementos programados en C++ a través de archivos .cpp o .cc, mediante la librería de OMNeT++ (omnetpp.h), lugar donde ocurren los eventos para ejecutar una simulación [16].

Mediante el uso del lenguaje C++ se programarán los módulos o elementos que conformarán la red de comunicación para la simulación tales como los IDS. Cabe mencionar que dicha programación se basa en la plataforma de Visual Studio, desarrollado por Microsoft, la cual se compone de un entorno de desarrollo integrado (IDE) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic.NET. En el apéndice A, el cual contiene el programa de la simulación contenido en un CD en programación C++

### 3.1.4 Ambiente de Simulación TKENV

Para llevarse acabo la ejecución de la simulación se requiere de un ambiente que lo permita en el programa el cual es conocido como *TKENV* y que se compone de la programación en *NED* y en C++, que posteriormente permitirán observar los resultados de la simulación a través de las herramientas mencionadas en la sección 4.1.1. A continuación se presentan los ambientes para (a) la interfaz gráfica y (b) la ventana de información del progreso de la simulación (Véase la Figura 3.2) [16].



(a)

(b)

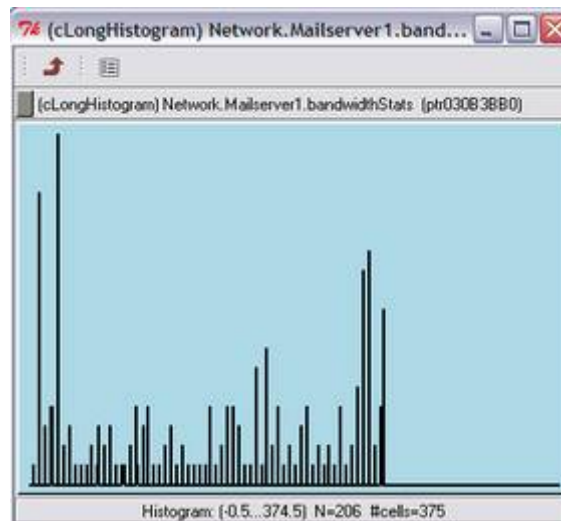
**Figura 3.2:** (a) Interfaz gráfica de OMNET++, (b) Ventana informativa de la simulación de *TKENV* [17].

Por otra parte tal como se muestra en la Figura 3.2 (b) se cuenta con los botones que permiten la ejecución y el progreso de una simulación, ambos en función del tiempo con el objetivo de generar resultados para posteriormente analizarlos e interpretarlos.

### 3.1.5 Herramientas de Graficación *Histogram*, *Vector* y *Scalars*.

Una vez ejecutada la simulación, se generan mensajes que viajan a través de la red diseñada, los cuales se componen de variables declaradas en la programación C++, dichas variables se almacenan en diversos archivos de graficación, de acuerdo a los eventos ocurridos. Al concluir la simulación OMNET++, se muestran estos archivos en imágenes o gráficas, las cuales varían dependiendo del tipo de variables almacenadas, ya sean vectores (*herramienta Vector*) o valores escalares (*scalars*).

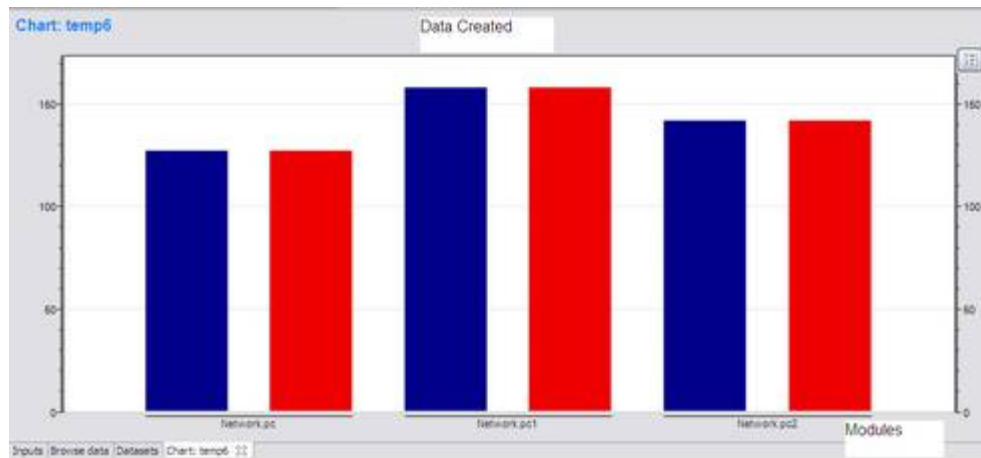
Posteriormente se muestra la herramienta *Histogram* (véase la Figura 3.3), la cual se basa en la traza de barras que indican el comportamiento del ancho de banda (*BW*, *Band Width*) en un elemento de la red o submódulo elegido, por la que pasan los mensajes o paquetes programados en la simulación, el nombre del archivo se conoce como *CLong Histogram*.



**Figura 3.3.** Gráfica *Histogram*. Muestra el comportamiento del BW a través de uno de los elementos de la red, para este caso el Mail Server1.

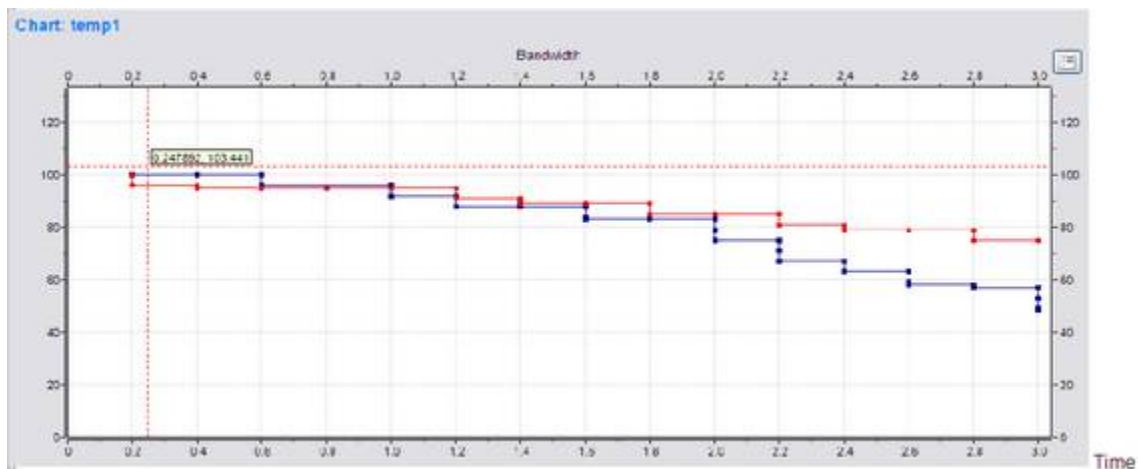
También se muestra la herramienta *Scalars* (véase la Figura 3.4). Esta herramienta da acceso al archivo de variables escalares, teniendo almacenadas varias corridas de simulación en el mismo archivo. Se genera un archivo por OMNET++ conocido como “*omnetpp.sca*” y muestra el módulo donde se almacenó la variable en el archivo, al igual que se muestra el nombre de la variable y su valor [16].

*Scalars* permite la aplicación de filtros para poder graficar la corrida que se desea, las variables que se desean y el módulo que se desea.



**Figura 3.4:** Herramienta *Scalars*.

Otra de las herramientas de graficación de este programa es *omnetpp.vec* (véase la Figura 3.5), el cual realiza el trazado de vectores en función del tiempo, en base al progreso o avance de la simulación [16]. En la siguiente figura se muestra el trazado del ancho de banda en función del tiempo empleando dicha herramienta.



**Figura 3.5:** Herramienta *Vector*.

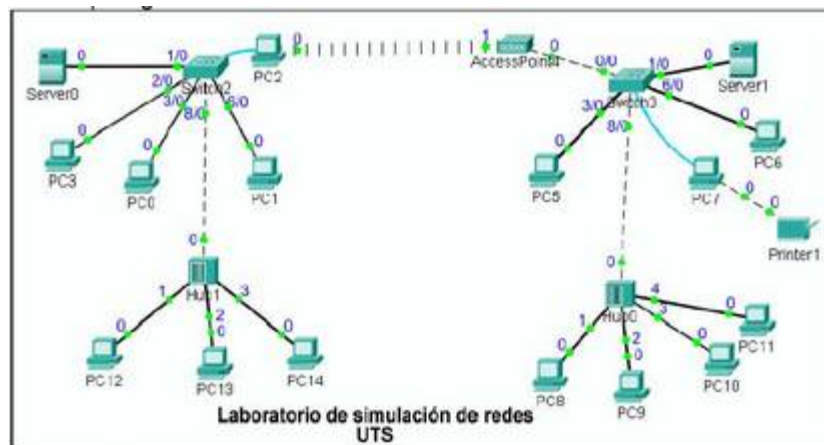
### 3.2 Programas Alternativos para la Simulación de Redes

#### 3.2.1 Packet Tracer

Uno de los programas que también funciona como un simulador gráfico de redes es *Packet Tracer* (véase la Figura 3.6), desarrollado y utilizado por Cisco como herramienta de entrenamiento para obtener la certificación CCNA<sup>4</sup>. *Packet Tracer* es un simulador de entorno de redes de comunicaciones de fidelidad media, que permite crear topologías de red mediante la selección de los dispositivos y su respectiva ubicación en un área de trabajo utilizando una interfaz gráfica.

En cuanto a sus características generales *Packet Tracer* es un simulador que permite realizar el diseño de topologías, la configuración de dispositivos de red, así como la detección y corrección de errores en sistemas de comunicaciones. Sin embargo como desventaja tiene que sólo 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; es decir, que con este programa no se pueden implementar simulaciones con tecnologías de red como Frame Relay, ATM (*Asynchronous Transfer Mode*), DSL (*Digital Subscriber Line*), Satelitales, telefonía celular entre otras. Mientras que con OMNET++ se pueden simular estas redes, en función del tiempo sincronizado con el progreso de la misma simulación [18].



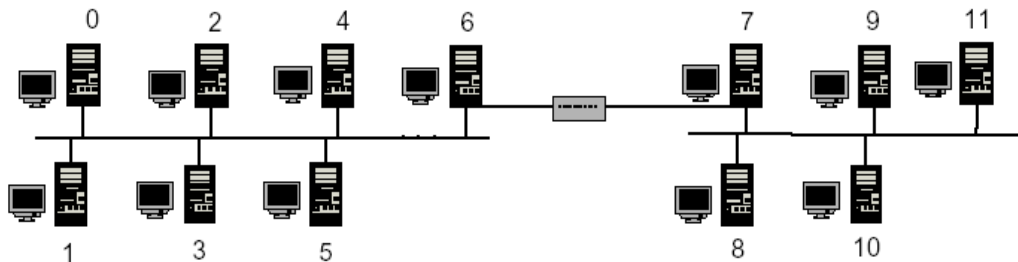
**Figura 3.6:** Topología de una Red con Packet Tracer [18].

<sup>4</sup>CCNA: Certificado Cisco de Redes Asociadas (*Cisco Certified Network Associate*).

### 3.2.2 NCTUns 2.0 Network Simulator/Emulator

Desarrollado por el profesor S.Y. Wang en *Harvard University*. Actualmente es un simulador de Redes de Computadores aplicado a docencia que permite analizar el tráfico de la red y medir su desempeño [18].

Esta herramienta es tanto un emulador como un simulador. Para realizar las simulaciones usa el mismo protocolo TCP/IP que se encuentre en el computador donde se ejecute el programa. Entre los tipos de redes que pueden simular se cuentan redes estructuradas con host fijos, LAN (*Local Area Network*) wireless, redes OBS (*Optical Burst Switching*), entre otros. Entre los dispositivos de red se pueden contar hubs ethernet, switches, router, host, estaciones, puntos de acceso wireless IEEE 802.11(b), estaciones base GPRS (*General Packet Radio Service*) y switches ópticos (véase la Figura 3.7) [18].



**Figura 3.7:** Topología de una Red con NCTUns 2.0 Network Simulator/Emulator [18].

En la Figura 3.7 se muestra una red de topología BUS para simular la transferencia de datos entre varios ordenadores y posteriormente obtener la tasa de transferencia respectiva. El elemento que permite la medición de esta transferencia está ubicado entre los nodos 6 y 7 de la red diseñada.

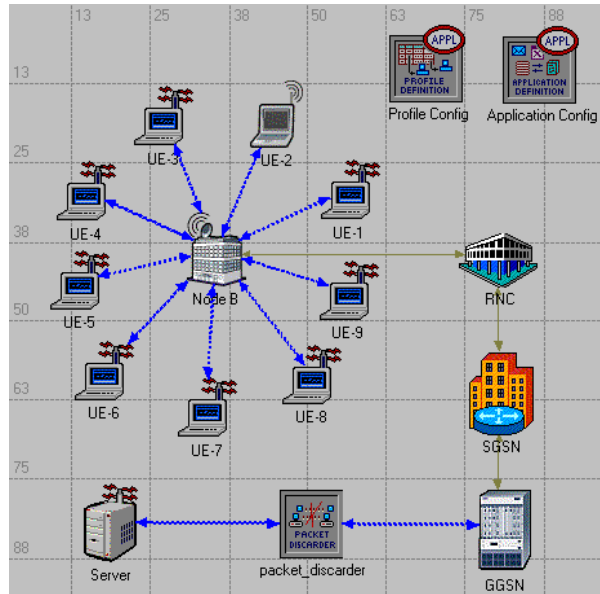
### 3.3.3 OPNET

OPNET es un programa simulador de redes desarrollado por el MIT (*Massachusetts Institute of Technology*) e introducido al mercado en 1987 como el primer simulador comercial. Se caracteriza por ser un programa ampliamente utilizado en la industria para modelar y simular



sistemas de comunicaciones; permite diseñar y estudiar redes, dispositivos, protocolos y aplicaciones brindando escalabilidad y flexibilidad (véase la Figura 3.8), cualidades que le permiten ofrecer a sus usuarios trabajar en procesos de investigación y desarrollo.

El software está orientado a simular objetos mediante un editor gráfico que permite diseñar una topología de red, soporta un amplio rango de tecnologías tipo LAN (*Local Area Network*), MAN (*Metropolitan Area Network*) y WAN (*Wide Area Network*).



**Figura 3.8:** Topología de una Red con OPNET.

La Figura 3.8 muestra una red inalámbrica en la que se calcula la pérdida de paquetes por medio del *Packet Discarder*, entre los nodos que establecieron una comunicación. En este programa se debe establecer inicialmente un perfil de simulación en el que se especifica la tecnología de comunicación utilizada

OMNET++ se caracteriza por ser un software de programación para simular el comportamiento de la red. Además de contar con librerías y comandos para la generación de ataques y así poder realizar el diseño del IDS.

El comportamiento de la red se puede generar en tan sólo unas cuantas líneas y con unos cuantos comandos. En OMNET++ existen también librerías que pueden ser usadas para la simulación de una red de comunicaciones. Sin embargo si uno desea añadir o modificar

ciertas características el usuario debe tener conocimiento de la programación en C++ para poder hacerlo.

### **3.4 Discusión**

En este capítulo se describe a OMNET++ como el programa para el diseño y la simulación del IDS, dando una breve explicación sobre los atributos o herramientas con las que cuenta este software para el modelado de una red y posteriormente obtener resultados al ser ejecutada alguna simulación. La plataforma de programación para este programa es C++.

Por otra parte se mencionan y describen a *Packet Tracer*, *NCTUns 2.0 Network Simulator/Emulator* y *OPNET* como simuladores para redes de comunicaciones, los cuales también pueden generar simulaciones para generar ataques y estudiar el tráfico y desempeño de la red diseñada sobre estos programas.