



SISTEMA DE POSGRADO
MAESTRÍA EN TELECOMUNICACIONES

TÍTULO DE LA TESIS:

Laboratorios virtuales para cursos de transmisión de datos

Previa la obtención del Grado Académico de Magíster en
Telecomunicaciones

ELABORADO POR:

ING. JORGE LUIS VELOZ ZAMBRANO

DIRIGIDO POR:

MSc. LUIS CÓRDOVA RIVADENEIRA

Guayaquil, Mayo de 2014



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

SISTEMA DE POSGRADO

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el Ing. Jorge Luis Veloz Zambrano como requerimiento parcial para la obtención del Grado Académico de Magíster en Telecomunicaciones.

Guayaquil, Mayo de 2014

DIRECTOR DE TESIS

Ing. Luis Córdova Rivadeneira, MSc.

REVISORES:

Ing. Orlando Philco Asqui, MSc.

Ing. Edwin Palacios Meléndez, MSc.

DIRECTOR DEL PROGRAMA

Ing. Manuel Romero Paz, MSc.



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

SISTEMA DE POSGRADO

DECLARACIÓN DE RESPONSABILIDAD

YO, JORGE LUIS VELOZ ZAMBRANO

DECLARO QUE:

La tesis “Laboratorios virtuales para cursos de transmisión de datos”, previa a la obtención del grado Académico de Magíster, ha sido desarrollada en base a una investigación exhaustiva, respetando derechos intelectuales de terceros conforme las citas que constan al pie de las páginas correspondientes. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico de la tesis del Grado Académico en mención.

Guayaquil, Mayo de 2014

EL AUTOR

ING. JORGE LUIS VELOZ ZAMBRANO



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

SISTEMA DE POSGRADO

AUTORIZACIÓN

YO, ING. JORGE LUIS VELOZ ZAMBRANO

Autorizo a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución de la Tesis de Maestría titulada: “Laboratorios virtuales para cursos de transmisión de datos”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, Mayo de 2014

EL AUTOR

ING. JORGE LUIS VELOZ ZAMBRANO

DEDICATORIA

A **mis hijas** quienes dan alegría a mi vida y son el incentivo para seguir adelante y alcanzar nuevas metas.

A **mis padres**, quienes con amor, dedicación y esfuerzo me enseñaron a vivir con responsabilidad en todas las acciones de mi vida.

A **mi Naña** (q.e.p.d.), quien fue mi segunda madre y que desde el cielo vigila cada uno de mis actos.

A **mi Andreita** el amor de mi vida quien ha sido mi punto de apoyo, y que con paciencia y amor ha sabido soportarme.

AGRADECIMIENTO

Al Sistema de Posgrado de la Universidad Católica Santiago de Guayaquil por permitirme obtener la formación científica en el área de las telecomunicaciones dentro de sus aulas.

Al MsC. Manuel Romero, Director de la Maestría en Telecomunicaciones de la Universidad Católica Santiago de Guayaquil por el apoyo brindado en el desarrollo y obtención de mi título.

Al MsC. Luis Córdova, por su colaboración y asesoría en la realización de esta investigación.

A todas aquellas personas que contribuyeron en la consecución de esta meta

RESUMEN

Los avances en materia de las Tecnologías de la Información y las Comunicaciones (TICs) involucran todos los sectores de la sociedad contemporánea. Las mejoras que éstas implantan en el proceso de aprendizaje instituyen oportuno su utilización. Tal es el caso de programas como; GNS3 y eNSP, los que permiten simular dispositivos de redes con un alto grado de similitud con los reales, siendo ambos programas libres, con entornos amistosos, flexibles e intuitivos y resultan fáciles de instalar y utilizar. En el presente trabajo se diseñaron prácticas de laboratorio utilizando el simulador de redes de datos eNSP y el GNS3 como emulador de dispositivos, estas serán utilizadas en los cursos de transmisión de datos. Para lograr este objetivo, primeramente se realizó una revisión detallada sobre los principales conceptos y características de los programas utilizados y posteriormente se simularon varios escenarios con diferentes tecnologías de redes, los cuales pueden ser utilizados en las prácticas de laboratorio.

Palabras Claves: GNS3, eNSP, laboratorios virtuales.

ABSTRACT

Advances in the field of Information Technology and Communications (ICT) involve all sectors of contemporary society. The advantages that they introduce into the teaching-learning process does appropriate it use. Such is the case of GNS3 and eNSP programs, to simulate networks very similar to the real ones, both free programs with environment friendly, flexible, intuitive and easy to install. In this paper virtual labs were designed using the data network simulator eNSP and GNS3 as the device emulator, these practices will be used in data transmission courses. To achieve this goal, a review of the main concepts and features of the software used was made, furthermore, simulations of several scenarios with different network technologies was made too, all of it to be used in the laboratory practices.

Keywords: GNS3, eNSP, virtual laboratories.

ÍNDICE GENERAL

<i>DEDICATORIA</i>	V
<i>AGRADECIMIENTO</i>	VI
<i>RESUMEN</i>	VII
<i>ABSTRACT</i>	VIII
<i>ÍNDICE GENERAL</i>	IX
<i>INTRODUCCIÓN</i>	13
<i>CAPÍTULO 1. LOS SIMULADORES DE REDES GNS3 Y ENSP</i>	15
<i>1.1 Simuladores de redes</i>	18
<i>1.2 Arquitectura del emulador GNS3</i>	19
<i>1.3 Dynamips</i>	21
<i>1.3.1 IDLE-PC</i>	22
<i>1.3.2 Instrumentales para optimizar uso de memoria</i>	22
<i>1.4 Dynagen</i>	23
<i>1.4.1 El archivo de la red</i>	24
<i>1.5 Sistema operativo IOS de Cisco</i>	24
<i>1.6 Formas de actuación de terminales de red</i>	26
<i>1.7 Arquitectura de los terminales de red</i>	26
<i>1.8 Conexión entre en terminales de red</i>	27
<i>1.8.1 Conexión de dispositivos de redes reales</i>	27
<i>1.9 Herramientas y protocolos adicionales usados en este trabajo</i>	28
<i>1.10 Requerimientos del sistema para trabajar con el GNS3</i>	31
<i>1.10.1 Requerimientos del sistema en Windows</i>	32
<i>1.10.2 Requerimientos del sistema en Linux</i>	32
<i>1.11 Empleo del GNS3</i>	33
<i>1.11.1 Emulación de enrutadores CISCO</i>	33
<i>1.11.2 Emulación de conmutadores Ethernet</i>	36
<i>1.11.3 Simulación de PCs</i>	38
<i>1.11.4 Elaboración de enlaces a módulos reales</i>	39

1.11.5 Alojamiento de referencia de red	41
1.11.6 Capacidad de captar datos.....	43
1.11.7 Trabajo en forma hypervisors	45
1.12 Arquitectura del simulador eNSP.....	48
1.13 Requisitos de configuración del sistema	49
1.14 Uso del eNSP.....	50
1.14.1 Configuración de enrutadores Huawei	50
1.14.2 Crear una topología en el eNSP.....	52
1.14.3 Captura de datos	54
1.14.4 Configuración del servidor y el cliente	55
1.14.5 Importación o exportación de un archivo de configuración.....	57
CAPÍTULO 2. LABORATORIOS VIRTUALES USANDO EL GNS3 Y EL ENSP	60
2.1 Aplicación del GNS3	60
2.1.1 Escenario del GNS3 con 2 enrutadores y 2 PCs.....	60
2.1.2 Escenario del GNS3 usando el protocolo VRRP.....	64
2.1.3 Escenario del GNS3 con una red IP/MPLS	67
2.2 Aplicación del eNSP.....	68
2.2.1 Escenario del eNSP con 2 enrutadores y 2 PCs	68
2.2.2 Escenario del eNSP con una red Frame Relay	71
2.2.3 Escenario del eNSP usando el protocolo Spanning Tree.....	73
2.2.4 Escenario del eNSP con una red IP/MPLS	75
VALORACIÓN ECONÓMICA	77
CONCLUSIONES	79
RECOMENDACIONES	81
BIBLIOGRAFÍA	82

ÍNDICE DE FIGURAS

Capítulo 1

<i>Figura 1. 1</i>	<i>Plataforma GNS3.....</i>	<i>19</i>
<i>Figura 1. 2</i>	<i>Comando que muestra la imagen y versión de un enrutador.....</i>	<i>25</i>
<i>Figura 1. 3</i>	<i>Ventana principal de GNS3.....</i>	<i>34</i>
<i>Figura 1. 4</i>	<i>Menú de opciones de un enrutador.....</i>	<i>35</i>
<i>Figura 1. 5</i>	<i>Ventana de configuración del nodo.....</i>	<i>35</i>
<i>Figura 1. 6</i>	<i>Ventana de IDLE PC.....</i>	<i>36</i>
<i>Figura 1. 7</i>	<i>Ventana de configuración del nodo.....</i>	<i>37</i>
<i>Figura 1. 8</i>	<i>Menú de opciones de una nube IP.....</i>	<i>40</i>
<i>Figura 1. 9</i>	<i>Ventana de configuración de nodo.....</i>	<i>41</i>
<i>Figura 1. 10</i>	<i>Ventana del nuevo proyecto.....</i>	<i>42</i>
<i>Figura 1. 11</i>	<i>Ventana de almacenamiento de configuraciones.....</i>	<i>43</i>
<i>Figura 1. 12</i>	<i>Menú de opciones.....</i>	<i>44</i>
<i>Figura 1. 13</i>	<i>Ventana de almacenamiento de configuraciones.....</i>	<i>44</i>
<i>Figura 1. 14</i>	<i>Enlace Ethernet entre enrutadores.....</i>	<i>45</i>
<i>Figura 1. 15</i>	<i>Ventana de captura.....</i>	<i>45</i>
<i>Figura 1. 16</i>	<i>Ventana de configuración de hypervisors.....</i>	<i>46</i>
<i>Figura 1. 17</i>	<i>Creación de un hypervisors.....</i>	<i>47</i>
<i>Figura 1. 18</i>	<i>Ventana principal del eNSP.....</i>	<i>51</i>
<i>Figura 1. 19</i>	<i>Menú de opciones de un enrutador.....</i>	<i>51</i>
<i>Figura 1. 20</i>	<i>Ventana de configuración del nodo.....</i>	<i>52</i>
<i>Figura 1. 21</i>	<i>Escenario del eNSP con un conmutador Ethernet y 2 PC.....</i>	<i>53</i>
<i>Figura 1. 22</i>	<i>Escenario del eNSP con un conmutador Ethernet y 2 PC.....</i>	<i>55</i>
<i>Figura 1. 23</i>	<i>Ventana de configuración del nodo.....</i>	<i>57</i>
<i>Figura 1. 24</i>	<i>Menú de opciones del enrutador.....</i>	<i>58</i>
<i>Figura 1. 25</i>	<i>Menú de opciones del enrutador.....</i>	<i>58</i>

Capítulo 2

<i>Figura 2. 1</i>	<i>Escenario del GNS3 con 2 enrutadores y 2 PCs.....</i>	<i>61</i>
<i>Figura 2. 2</i>	<i>Configuración del enrutador R1 visto con el cliente Telnet de SecureCRT.....</i>	<i>62</i>

Figura 2. 3 Uso del Multiservidor Paessler (izquierda) y una máquina virtual VirtualBox de Sun.....	63
Figura 2. 4 Muestra de captura con el Wireshark al realizar un ping a 192.168.2.2.....	64
Figura 2. 5 Escenario del GNS3 usando el protocolo VRRP.....	65
Figura 2. 6 Muestra de captura con el Wireshark al realizar un ping a 192.168.2.2.....	66
Figura 2. 7 Muestra de captura con el Wireshark al realizar un tracer a 192.168.1.1.....	66
Figura 2. 8 Escenario del GNS3 con una red IP/MPLS.....	67
Figura 2. 9 Captura de tráfico con el Wireshark en el escenario IP/MPLS. ..	68
Figura 2. 10 Escenario del eNSP con 2 enrutadores y 2 PCs.....	69
Figura 2. 11 Muestra de captura con el Wireshark al realizar un ping a 192.168.2.2.....	70
Figura 2. 12 Muestra de captura con el Wireshark al realizar un tracert a 192.168.3.2.....	70
Figura 2. 13 Escenario del eNSP con una red Frame Relay.....	71
Figura 2. 14 Muestra de captura con el Wireshark al realizar un tracert a 192.168.2.2.....	72
Figura 2. 15 Muestra de captura con el Wireshark al realizar un ping a 192.168.1.2.....	72
Figura 2. 16 Escenario del eNSP con una red Spanning Tree.....	73
Figura 2. 17 Muestra de captura con el Wireshark al realizar un ping a 192.168.1.30.....	74
Figura 2. 18 Muestra de captura con el Wireshark al realizar un tracert a 192.168.1.20.....	75
Figura 2. 19 Escenario del eNSP con una red IP/MPLS.....	76
Figura 2. 20 Captura de tráfico con el Wireshark en el escenario IP/MPLS.	77

ÍNDICE DE TABLAS

Tabla 1 Requisitos de configuración del sistema.....	50
Tabla 2 Valoración económica de los precios aproximados de los dispositivos empleados.....	78

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC) demuestran un progreso presuroso en la sociedad actual. Coyunturalmente con ello, en el aspecto tecnológico, este sector se identifica por el progreso incesante de eventos innovadores, sean de tipo radical, o de tipo incremental, el proceso o producto tecnológico es el resultado de transferencias de conocimientos basados en las TIC.

Es por ello que las organizaciones de telecomunicaciones demandan destinar cada año un alto presupuesto a la adiestramiento de su capital humano, con el intención de actualizarlo en las nuevas tecnologías, de forma que les permita su introducción y operación correcta. De ahí que una efectiva capacitación, convierta los gastos en este sentido en una inversión, al elevar la competitividad, posición respecto al mercado y satisfacción del cliente. (Avila, 2013) y (Hernández R. , 2008)

Las empresas de telecomunicaciones juegan un rol decisivo en el desarrollo de la informatización de la sociedad. En estas, a partir de la determinación de las necesidades de capacitación, se identifican oportunamente aquellas temáticas que requieren ser tratadas, siendo la rama de la transmisión de datos una de las de mayor prioridad (Avila, 2013).

Esto está dado, por el auge que esta rama ha experimentado a nivel mundial, con su consecuente introducción y despliegue gradual de las nuevas tecnologías en todo el territorio nacional, que se evidencia en el incremento de la técnica instalada y en servicios. (Boney, 2005)

En la actualidad existe una variada gama de tecnologías en explotación, así como diversos escenarios de aplicación y operación. Con el fin de atender esta tendencia del incremento de servicios de datos en nuestro país, se ha incrementado la formación y adiestramiento del personal que atiende estas funciones.

Para ello se requiere elaborar los materiales didácticos a utilizar en esas acciones de capacitación, que contribuyan a la apropiación adecuada por los estudiantes de los contenidos impartidos y al desarrollo de habilidades y competencias, entre los que se

incluyen aquellos avances en materia de las TIC, como apoyo a la docencia (Boney, 2005) y (Avila, 2013)

Una de las formas de aplicación de las TIC, son las llamadas salas virtuales, que tienen un entorno de laboratorio para efectuar simulación, con computadores, instrumentos virtuales y programas, que deriva en una opción para la sustitución de las computadoras, equipos y herramientas convencionales. (Lorandi, Hermida, Hernández, & Guevara, 2011)

Entre las importantes ventajas que los laboratorios virtuales implantan al proceso enseñanza-aprendizaje, se obtiene un entorno económico en la ejecución de las acciones llamadas tradicionalmente como prácticas de laboratorio, debido al superior costo de las infraestructuras experimentales, resultan útiles de emplear en situaciones que pueden implicar riesgos o demorar un tiempo prolongado, que de otra forma tendrían escasas posibilidades de realizarlas.

Contribuyen a resolver la obsolescencia de los equipos convencionales de laboratorio, al mejorar fácilmente las prestaciones de los programas y avances tecnológicos en este campo, con la instalación y/o actualización de nuevas versiones puestas a disposición del mercado. Permiten la enseñanza personalizada, donde los estudiantes lleven su propio ritmo de aprendizaje y puedan repetir los eventos o fenómenos cuantas veces quieran, de forma que se enfrenten de modo individual al proceso de elaboración de sus propias conclusiones, con relación a los fenómenos que van a simular y sobre la base de sus propios errores.

Puede conseguir un grado de detalle tal que se logre una visión mucho más desarrollada de aquellos fenómenos en que su contraparte, los laboratorios convencionales, no pueden ser observados con la suficiente claridad gráfica. (Avila, 2013) y (Lorandi, Hermida, Hernández, & Guevara, 2011)

En la actualidad, los programas de simulación disponibles, con ambientes amistosos, flexibles e intuitivos, ofrecen un grado de seguridad aceptable, lo que ha hecho posible que la opción de laboratorios virtuales implique cada vez más empleada en la enseñanza de la ingeniería. (Avila, 2013) y (Lorandi, Hermida, Hernández, &

Guevara, 2011). Específicamente, en la rama de las redes de datos ganan muchos adeptos programas como el OPNET y Cisco Packet Tracer.

Recientemente, han aparecido nuevos programas como el GNS3 y el eNSP. Entre sus ventajas se pueden mencionar el alto grado de similitud con los dispositivos de redes reales y encontrarse de forma gratuita en Internet, lo anterior sugiere su estudio y de ser posible, su introducción y aplicación práctica.

El GNS3 es un emulador de redes que incluye dispositivos del fabricante Cisco, mientras que el eNSP es un simulador de redes que emplea equipos del fabricante Huawei. Ambos disponen un entorno gráfico de diseño y permiten virtualizar enrutadores, conmutadores y PCs y de esta forma, crear escenarios complejos de redes de datos.

Tradicionalmente, las redes de comunicaciones se han diseñado sobre la base de métodos empíricos, los cuales involucran riesgos de fallo que atentan contra su buen desempeño, lo que no es conveniente, tanto para usuarios como para proveedores. Con la utilización de los programas de simulación se evitarían esos riesgos.

Los programas GNS3 y eNSP se conocen últimamente como “*testbed*” (banco de pruebas), cuyo uso es cada vez más frecuente en las empresas de diseño o en aquellas que necesiten probar a priori nuevos diseños, funcionalidades o topologías en sus redes. (Boney, 2005)

Antecedentes del problema de Investigación

En la actualidad los equipos de redes, como enrutadores, conmutadores, servidores y concentradores han alcanzado un alto costo en el mercado, debido a esto las empresas de telecomunicaciones no disponen de maquetas con dispositivos de redes reales que permitan a los estudiantes una mejor visión del funcionamiento de la transmisión de datos en los mismos.

Problema de Investigación

Necesidad de incorporar un componente práctico en los cursos de transmisión de datos que se imparten en las empresas de telecomunicaciones.

Objeto: Cursos de transmisión de datos.

Objetivo general

Diseñar prácticas de laboratorio utilizando simuladores de redes para usarse en los cursos de transmisión de datos que se imparten en las empresas de telecomunicaciones.

Objetivos específicos

- Realizar un estudio de los simuladores de redes GNS3 y eNSP.
- Demostrar que con el programa GNS3 se pueden emular plataformas de hardware de varios modelos de enrutadores reales de CISCO, como las plataformas 1700, 2600, 2691, 3600, 3700 y 7200.
- Demostrar que con el programa eNSP se pueden simular redes que empleen equipos del fabricante Huawei.
- Comprobar que la integración en el GNS3 de varios programas de emulación, permite virtualizar en una única PC múltiples escenarios de prueba, redundando en un mejor aprovechamiento del tiempo en el desarrollo de las actividades planificadas

Hipótesis

Si se utilizaran simuladores de redes para diseñar prácticas de laboratorio, se podría contar con herramientas que desde el punto de vista práctico contribuyan al desarrollo de los cursos de transmisión de datos que se imparten en las empresas de telecomunicaciones.

Metodología del proyecto

Esta investigación para diseñar prácticas de laboratorio utilizando simuladores de redes para usarse en los cursos de transmisión de datos que se imparten en las

empresas de telecomunicaciones se desarrolla con un perfil explicativo y se aplica el paradigma Empírico-Analítico.

La investigación se realiza con un enfoque cualitativo al recolectar datos sin medición numérica para probar la validez de la hipótesis y un diseño no experimental en razón de que no se alterarán las variables durante el estudio.

CAPÍTULO 1. LOS SIMULADORES DE REDES GNS3 Y ENSP

En este capítulo se realiza una revisión detallada sobre los principales conceptos y características del GNS3 y eNSP. Se expone notoriamente cómo establecer el uso de los dos programas, especificando las acciones que hay que cumplir para la simulación de los disímiles dispositivos soportados, Además se analizan las características básicas necesarias para la instalación de estas herramientas.

1.1 Simuladores de redes

Con el desarrollo de las redes, se ha hecho necesaria la creación de herramientas capaces de mostrar el comportamiento de las diferentes configuraciones adoptadas por los dispositivos de red de forma virtual sin necesidad de llegar a la implementación, sin un resultado previo. Para esto se han diseñado diferentes simuladores, entre los que se encuentran principalmente:

- GNS3 (*Graphical Network Simulator*; Simulador de Gráfico de Red)
- eNSP (*Enterprise Network Simulation Platform*; Plataforma de Simulación de Red Empresarial)
- ns-2 (*Network Simulation-2*; Simulador de Red-2)
- ns-3 (*Network Simulation-3*; Simulador de Red-3)
- *Packet Tracer*
- *OPNET Modeler*

Para la realización de este trabajo se realizó un estudio de los simuladores antes mencionados, a pesar de tener características similares, se ha decidido la utilización de los programas GNS3 el cual es un emulador de redes que incluye dispositivos reales del fabricante Cisco y eNSP es un simulador de redes que emplea equipos del fabricante Huawei.

Estos programas presentan varias ventajas respecto a los demás, entre las que se pueden mencionar el alto grado de similitud con los dispositivos de redes reales además de su funcionamiento sobre *software* libre, por lo que se eligieron para el desarrollo del proyecto.

1.2 Arquitectura del emulador GNS3

El emulador GNS3 trabaja en conjunto con otros programas para lograr la emulación de dispositivos de redes reales, creando una plataforma que permite el fácil diseño de topologías de redes complejas. Por lo que es idóneo para el desarrollo de los conocimientos de estudiantes que desean familiarizarse con dispositivos de red.

Este programa es una aplicación elaborada en Python que emplea las librerías de Dynagen para implantar una interfaz gráfica (GUI). Sus primordiales destinos son editar el archivo de texto “.net” y ejecutar las operaciones de la interfaz de líneas de comandos (CLI) hecha por Dynagen y Dynamips. Adicionalmente incorpora la capacidad de simular computadoras. La unificación de la plataforma GNS3 se muestra en la figura 1.1 (Díaz, 2012) y (Hernández R. , 2008).



Figura 1. 1 Plataforma GNS3

Fuente: (Díaz, 2012)

Las principales ventajas que presenta el GNS3 y que han servido como punto de partida para tomar la decisión de estudiar este emulador son las siguientes:

- Es un programa libre de licencia, se puede descargar libremente de Internet, siendo su sitio oficial: <http://www.gns3.net/>.

- Es fácil de instalar, ya que todos los programas que necesita para funcionar se encuentran en un solo paquete de instalación (Monteverde, 2014).
- Emula las plataformas de *hardware* de varios modelos de enrutadores reales de CISCO (1700, 2600, 3600, 3700, 7200) y ejecuta imágenes IOS (*Internetwork Operating System*; Intersistema operativo de red) estándares.
- Permite la conexión del entorno virtual con el mundo real, a través de las interfaces físicas de red con que cuente la computadora (PC) donde se encuentre instalado.
- Es apropiado para simular redes de grandes tamaños, ya que permite que un cliente GNS3 pueda correr en una máquina diferente a la que contiene al emulador Dynamips, repartiendo el procesamiento entre diferentes computadoras (Monteverde, 2014).
- Puede capturar los paquetes que pasan por enlaces virtuales y escribir los resultados de la captura en archivos, que pueden ser interpretados por aplicaciones como *Wireshark* (Monteverde, 2014).
- Posibilita la salva y ejecución posterior, tanto de las configuraciones de los enrutadores como del escenario implementado. Al poder editar los ficheros de configuración con extensión .net, es posible adaptarlos a la configuración del GNS3 que se tenga en otras computadoras, haciéndolos de esta forma interoperables.
- Permite conformar estructuras topológicas complejas, que incluyen enrutadores, conmutadores Ethernet, Frame Relay y ATM, máquinas virtuales, entre otros.
- Las versiones del programa GNS3 están en constante desarrollo y los fórum de discusión en Internet aportan novedosos ejemplos de aplicación y solución a los problemas presentados por los foristas.

Entre las principales desventajas que presenta el GNS3 se encuentran:

- Solo admite imágenes de IOS de enrutadores CISCO.
- Las imágenes no vienen incluidas en el paquete de instalación del programa, son propietarias de CISCO y para su uso se requiere de licencia.
- Demanda una buena cantidad de memoria y CPU en la PC donde se instale, siendo directamente proporcional a la cantidad de equipos que emule. Es por ello que la PC requiere de buenas prestaciones.

1.3 *Dynamips*

Dynamips es el motor de emulación que permite emular diferentes plataformas *hardware*, utiliza imágenes de sistemas operativos de CISCO (IOS) en una misma PC. Entre dichas plataformas se encuentran los enrutadores 1700, 2600, 3600, 3700 y 7200. Además puede emular conmutadores Ethernet, Frame Relay y ATM con funcionalidades básicas. (Boney, 2005)

Dynamips no es capaz de emular conmutadores *Catalyst*, sino que provee una versión limitada de un conmutador virtual, cuyas limitaciones pueden ser resueltas usando métodos alternativos como la emulación de NM-16ESW que el emulador sí soporta.

Inicialmente *Dynamips* consume grandes cantidades de CPU del PC emulador, esto se debe principalmente a que realiza la emulación de los enrutadores instrucción por instrucción ya que no puede saber cuándo un enrutador virtual está inactivo, de modo que ejecuta instrucciones como si la imagen del IOS estuviera realizando algún trabajo útil.

Dynamips también consume memoria RAM del PC emulador, ya que, en teoría cada enrutador virtual debe tener a su disposición, como mínimo, toda la cantidad de memoria RAM que necesita para poder trabajar, por lo tanto, esta cantidad se hace impráctica si se requieren emular redes con varios enrutadores.

Para resolver el problema del excesivo uso de memoria del PC emulador, se usan herramientas que permiten compartir la memoria del mismo entre varios enrutadores emulados con la misma IOS y herramientas que usan el disco en vez de la memoria del emulador. Un programa destinado para este fin es el IDLE-PC (computadora inactiva).

1.3.1 IDLE-PC

IDLE-PC se trata de una herramienta que realiza un análisis en el código de una imagen IOS para determinar los puntos más probables que representen un bucle de inactividad, de modo que, cuando se detecten, haga que los enrutadores virtuales “duerman” durante ese instante, es decir, IDLE-PC ayuda a *Dynamips* a emular el estado inactivo de la CPU virtual de un enrutador. (Boney, 2005)

Algunas características adicionales de este proceso:

- La aplicación de un mal valor de IDLE-PC hace que la PC del emulador trabaje entre 60% y 100% cuando emula un solo enrutador, mientras que un buen valor hace que sólo trabaje entre el 1% y el 10% de la capacidad. Estos valores dependen de la potencia del emulador usado.
- Está ligado a la versión de Dynamips que se use, si se cambia de versión, es muy probable que se necesite cambiar de valor de IDLE-PC.
- De acuerdo a las versiones y plataformas utilizadas los IOS serán diferentes
- No son exclusivos de un PC o sistema operativo, por lo tanto, los archivos “dynagenidldb.ini” pueden ser copiados y compartidos y el valor de IDLE-PC seguirá siendo bueno.

1.3.2 Instrumentales para optimizar uso de memoria.

Dynamips utiliza diferentes herramientas para optimizar el uso de memoria, tanto real como virtual, del PC emulador, los siguientes aspectos son del autor (Hernández R. , 2008) y (Díaz, 2012) :

- *Ghostios*: Se encarga de minimizar la cantidad de memoria real que se necesita del emulador para instituir topologías con enrutadores que corran a la vez, es decir, admite que el emulador participe una parte de su memoria entre todos los enrutadores que usen una misma imagen IOS de modo que cada enrutador emulado no tenga que acumular una copia idéntica de un mismo IOS en su memoria virtual. La consecuencia, en este caso, es un archivo que contiene la región de memoria compartida ubicado en el directorio de trabajo, llamado c2600-i-mz.123-3h.image.ghos.

- *Sparsemem*: Se encarga de reducir la cantidad de memoria virtual que usa un enrutador emulado, es decir, la memoria necesaria para la ejecución de una IOS, ya que sólo asigna la cantidad de memoria que la IOS va a usar en un momento determinado y no toda la memoria RAM configurada, lo que permite crear más enrutadores virtuales por proceso *Dynamips*. Esta herramienta no está habilitada por defecto (Díaz, 2012) y (Hernández R. , 2008).

- *Mmap*: Realiza la correspondencia de archivos temporales del disco con la memoria virtual configurada en los enrutadores emulados, para que cuando se requiera leer estos archivos, el sistema operativo ponga en caché sólo las secciones de los mismos que están siendo utilizados. Estos archivos tienen la extensión “ram”, el tamaño de la memoria RAM configurada y se encuentran en el directorio de trabajo creado por GNS3 en cada simulación.

1.4 Dynagen

Dynagen es una interfaz escrita en *Python* que provee la gestión, mediante línea de comando (CLI), de los escenarios emulados por *Dynamips* creando más fácil su uso. Simplifica la gestión de las redes virtuales ya que implementa comandos para listar, iniciar, parar, reiniciar, suspender, reanudar los diferentes dispositivos emulados, además establece los valores de IDLE-PC y ejecuta capturas de paquetes.

A partir de sus últimas versiones, *Dynagen* es capaz de trabajar con el emulador de *firewalls* PEMU, el cual viene integrado en GNS3 dotando al emulador de capacidad de añadir *firewalls* CISCO en las topologías. Además es capaz de conectar de forma transparente a *Dynamips* los diferentes dispositivos virtuales como conmutadores Ethernet, Frame-Relay y ATM soportados por *Dynamips*.

Dynagen usa un archivo de texto de fácil interpretación llamado “*Network File*”, con extensión “.net”, para conocer todas las características de *hardware* de los dispositivos de red a emular y realizar las interconexiones entre ellos (Díaz, 2012) y (Hernández R. , 2008)

1.4.1 El archivo de la red

Se trata de un archivo, escrito usando sintaxis INI (*INI file syntax*), que almacena la configuración de todos los dispositivos de red de la topología virtual a simular, como son los enrutadores, conmutadores y las interconexiones entre ellos. Este archivo puede especificar valores tan concretos como los descriptores de los adaptadores de red (NIO) que se encargan de la conexión con equipos reales o los puertos en los que trabajan dichos adaptadores de red a red, etc. (Díaz, 2012) y (Hernández R. , 2008)

1.5 Sistema operativo IOS de Cisco

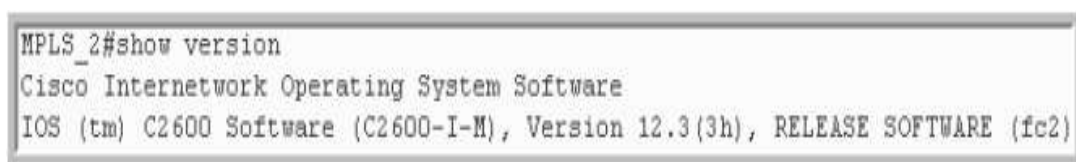
El IOS (*Internetwork Operating System*; Intersistema operativo de red) de CISCO es el sistema operativo usado en los dispositivos de red de CISCO: enrutadores y conmutadores, para la implementación de redes que forman la gran Internet. Una forma de acceso al CISCO IOS es usando CLI (*Command Line Interface*; Interfaz de Línea de Comando), la cual está basada en un conjunto de comandos. Estos están ubicados en los diversos modos de operación definidos en cada equipo de red: usuario, privilegiado y configuración global (Díaz, 2012) y (Hernández R. , 2008).

La selección de una adecuada versión del CISCO IOS está relacionada con las características técnicas que pueden ofrecer los equipos de red, por ejemplo, el

soporte de IPv6, MPLS, DiffServ, seguridad, entre otros. Además, como los enrutadores tienen diversos tipos de interfaces, siendo las más comunes Ethernet y Serie, el IOS debe contener un grupo de *drivers* para soportar esta variedad de interfaces.

Las imágenes de CISCO IOS poseen nombres específicos y estandarizados por CISCO que reflejan sus principales características como la plataforma que soporta, las funciones que realiza y la versión de IOS que utiliza para facilitar su actualización.

Es importante conocer el nombre de la imagen y la versión que está corriendo en un enrutador antes de comenzar a configurarlo. Para ello se usó el comando “*show versión*” que se muestra en la figura 1.2.



```
MPLS_2#show version
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-I-M), Version 12.3(3h), RELEASE SOFTWARE (fc2)
```

Figura 1. 2 Comando que muestra la imagen y versión de un enrutador.

Fuente: (Boney, 2005)

Típicamente el nombre del archivo que contiene a una imagen CISCO IOS consta de una parte que indica las características de la IOS y otra que corresponde a la versión de la misma. En la figura 1.2, el nombre de la imagen IOS es C2600-I-M, la versión que se usa es la 12.3 (3h) y el estado de la versión es fc2. (Merino, 2011)

Las características que posee una imagen IOS se indican en tres grupos de letras estandarizadas por CISCO, el primer grupo hace referencia a la plataforma, el segundo a las funcionalidades de la imagen y el tercero a su formato.

Plataforma: Corresponde al primer grupo de caracteres e identifica la plataforma para la cual la imagen fue creada.

Grupo de características: Este grupo representa las funcionalidades que un enrutador con una IOS puede realizar, estas funcionalidades pueden combinarse, pero siempre dependen de que la plataforma pueda soportarlas.

Formato de imagen: Generalmente en este grupo se encuentra dos caracteres, el primero nos dice desde dónde se ejecuta la imagen, y el segundo cómo se ha realizado la compresión de esta.

1.6 Formas de actuación de terminales de red

Los comandos del CISCO IOS están agrupados en los tres modos de operación: modo usuario, modo privilegiado y modo de configuración.

El modo usuario, es el primer modo al que todo usuario accede. La mejor manera de conocer que se encuentra trabajando en éste modo es observando el siguiente símbolo “>”, este aparece después del nombre del equipo de red. (Scaniello & Sosa, 2005)

El modo privilegiado, permite a los usuarios ver la configuración del equipo de red (enrutador o conmutador), restablecer el equipo de red e ingresar al modo de configuración. Este se puede distinguir observando el símbolo “#” como indicador después del nombre del equipo de red. El administrador del equipo de red puede habilitar la solicitud de una contraseña antes de ingresar al modo usuario y/o privilegiado. (Scaniello & Sosa, 2005)

El modo de configuración, permite modificar la configuración del sistema. El ingreso al mismo se realiza desde el modo privilegiado, ingresando el comando “*configure terminal*”. El modo configuración se puede distinguir observando (*config*) # después del nombre del equipo de red. (Scaniello & Sosa, 2005)

1.7 Arquitectura de los terminales de red

La arquitectura de los dispositivos de red es similar a un computador. Cada enrutador tiene una CPU que varía en rendimiento y capacidad, dependiendo de la plataforma del enrutador, e interfaces para conectar periféricos. Además de la CPU y las

interfaces, los enrutadores necesitan de cuatro tipos de memorias para su normal funcionamiento: ROM, Flash, RAM y NVRAM.

Memoria ROM: es de sólo lectura, por lo que los datos no pueden ser escritos en este tipo de memoria. El programa inicial que corre sobre un enrutador CISCO es llamado programa de secuencia inicial de instrucciones (*bootstrap software*) y es almacenado en la memoria ROM; este programa es invocado cuando se arranca el enrutador. Los enrutadores emulados con GNS3 ignoran la lectura de esta memoria.

Memoria Flash: es usada para almacenar uno o más programas CISCO IOS. En algunos sistemas, la memoria *flash* puede contener el programa de secuencia inicial de instrucciones e incluso archivos de configuración o información del sistema. Los enrutadores emulados con GNS3 carecen de esta memoria.

Memoria RAM o DRAM: es una memoria rápida que pierde la información almacenada cuando la energía del enrutador es desactivada. Es usada para mantener tablas y *buffers* del programa CISCO IOS.

Memoria NVRAM: para almacenar la configuración de arranque, es decir, los archivos que el CISCO IOS lee cuando el enrutador se reinicializa. La memoria NVRAM puede ser obviada cuando se reinicializa el enrutador, modificando el valor del registro interno al valor 0x2142. Esta acción es necesaria cuando se requiere cambiar/eliminar las contraseñas definidas en un enrutador. GNS3 usa el disco en vez de esta memoria.

1.8 Conexión entre en terminales de red

En este epígrafe se describen las recomendaciones de interconexión entre el emulador GNS3 y un dispositivo de red real.

1.8.1 Conexión de dispositivos de redes reales

Para la conexión del emulador GNS3 a dispositivos de red reales, en este caso enrutadores, se necesitan los siguientes elementos:

- Un enrutador CISCO, en cualquiera de sus plataformas.

- Un cable de energía eléctrica.

- Una PC con el emulador GNS3 instalado.

- Un cable UTP cruzado con conectores de RJ45 a RJ45.

Pasos a seguir:

- 1) Conectar el enrutador al PC usando el cable UTP cruzado desde el puerto FastEthernet (RJ45) del enrutador al puerto FastEthernet de la PC (RJ45).

- 2) Cargar el programa de emulación de redes GNS3 en el PC y crear un nuevo enlace a equipos reales. En la configuración del enlace, tener cuidado en seleccionar el nombre del puerto FastEthernet que fue usado para conectar el enrutador.

- 3) Encender el enrutador moviendo el interruptor a la posición de ON. Esperar hasta que se note que el LED SYS PWR se enciende para indicarnos que el enrutador se inició satisfactoriamente.

- 4) En el enrutador, configurar la dirección IP en la interfaz FastEthernet de acuerdo a la topología creada en el emulador.

- 5) En el PC, no asignar ninguna dirección IP a la interfaz FastEthernet, ya que dicha dirección IP será asignada virtualmente tras la configuración del enrutador virtual emulado por GNS3.

1.9 Herramientas y protocolos adicionales usados en este trabajo

En este epígrafe se describen los protocolos y herramientas adicionales usadas en este trabajo.

Protocolo ICM

Es un protocolo usado para informar a la fuente acerca del procesamiento de los datagramas IP que envía, con el fin de saber si se han producido errores en la comunicación. En este sentido, ICMP no se usa estrictamente para dar fiabilidad a IP, ya que esta debe ser implementada por los protocolos de nivel superior que usen IP.

En la práctica, los enrutadores generan mensajes ICMP para reportar errores, mientras que los PCs de destino sólo envían aquellos mensajes que pueden implementar.

Los mensajes ICMP se encapsulan en datagramas IP, por lo tanto, ICMP es parte de IP y debe ser implementado por él, es decir, la cabecera IP siempre contendrá un número de protocolo de 1 (ICMP) y los datos IP serán los auténticos mensajes ICMP. Dichos mensajes ICMP poseen una estructura específica donde el tipo de mensaje que se transporta se representa por números 0, 8, 9, 10 y del 13 al 18.

Unos tipos de mensajes ICMP muy conocidos son los llamados “*echo*” usados principalmente para detectar si otra PC de la red está activa aunque también puede medir la latencia de un paquete IP o auto-comprobar que la interfaz de red de nuestro PC está activa.

El comando que los implementa se llama *ping*, donde la fuente envía un mensaje “*echo request*” al PC de destino y el receptor cambia el tipo del mensaje a “*echo reply*” para devolver el datagrama a la PC fuente.

Protocolo VRRP (*Virtual Router Redundancy Protocol*)

Es un protocolo de redundancia no propietario diseñado para aumentar la disponibilidad de la puerta de enlace por defecto dando servicio a máquinas en la misma subred. El aumento de fiabilidad se consigue mediante el anuncio de un enrutador virtual como una puerta de enlace por defecto en lugar de un enrutador físico.

Dos o más enrutadores físicos se configuran representando al enrutador virtual, con sólo uno de ellos realizando realmente el enrutamiento. Si el enrutador físico actual que está realizando el enrutamiento falla, el otro negocia para sustituirlo.

Protocolo Spanning Tree (STP)

Es un protocolo de red de nivel 2 del modelo OSI. Su función es la de gestionar la presencia de bucles en topologías de red debido a la existencia de enlaces redundantes (necesarios en muchos casos para garantizar la disponibilidad de las conexiones). El protocolo permite a los dispositivos de interconexión activar o desactivar automáticamente los enlaces de conexión, de forma que se garantice la eliminación de bucles.

Hay dos versiones del STP, la original (DEC STP) y la estandarizada por el IEEE (IEEE 802.1D), que no son compatibles entre sí. En la actualidad, se recomienda utilizar la versión estandarizada por el IEEE.

Arquitectura MPLS

Según el trabajo de (García R. , 2014) señala que la tecnología MPLS (*Multiprotocol Label Switching*) es un componente de transporte de datos estándar establecido por la IETF (*Internet Engineering Task Force*) y determinado en el RFC 3031. Maneja su operación entre la capa de enlace de datos y la capa de red del modelo OSI (*Open System Interconnection*).

Fue delineado para agrupar el servicio de transporte de datos para las redes de circuitos y de paquetes. Puede ser manejado para transportar desiguales tipos de tráfico, incluyendo intercambio de voz y de paquetes IP.

Wireshark

Este programa permite la captura de paquetes y el análisis de protocolos, cuenta con una amplia gama de filtros que facilitan la definición de criterios de búsqueda para los más de 1100 protocolos por él soportados.

Dispone de una interfaz sencilla e intuitiva que permite desglosar por capas cada uno de los paquetes capturados. Además es posible visualizar los campos de cada una de las cabeceras y capas que componen los paquetes monitorizados. (Merino, 2011)

Máquinas virtuales

Tal como indica (García R. , 2014) esta aplicación se distingue por implantar una capa de abstracción, pero elaborando instrucciones en una máquina del mismo tipo, acarrea como efecto lograr un computador dentro de otro. Ejemplos de éstas y con las cuales se trabajó en el presente trabajo son:

- Virtual Box.

- Virtual PC Simulator.

Simulador de Multiservidores PEASSLER

La aplicación del simulador de servidores múltiples será capaz de simular redes de servidores de gran tamaño, virtuales del tipo HTTP, FTP, SMTP o DNS, y también conmutadores basados en redes SNMP.

El número de los servidores y también de los conmutadores virtuales que pueden ser creados se encuentra solamente limitado por las direcciones IP que se encuentren disponibles y también por los puertos TCP y además por el desempeño de la red y del sistema.

Con la herramienta simulador de servidores múltiples, la configuración de una red grande de 100 servidores y 20 conmutadores tomaría sólo unos cuantos minutos y le resultará mucho más rápido que tener que instalar y además configurar programas para servidores en una computadora personal.

1.10 Requerimientos del sistema para trabajar con el GNS3

Para poder instalar el emulador GNS3 en una PC, estas deben cumplir algunos requisitos en su sistema. A continuación se muestran las características básicas necesarias para la instalación de esta herramienta.

1.10.1 Requerimientos del sistema en Windows

A continuación se detallan los requerimientos del sistema para trabajar con el GNS3:

Memoria RAM: *Dynamips* asigna por defecto 16MB de memoria RAM al compilador JIT para que realice la compilación del código del simulador en sistema Windows. Además, cada imagen IOS de un enrutador real requiere una cantidad determinada de memoria RAM para funcionar, inicialmente, la suma de los valores anteriores sería la cantidad de memoria RAM real necesaria para la simulación de un enrutador. En la práctica este valor es mucho menor debido a que *Dynamips* implementa herramientas que permiten una optimización del uso de la memoria del simulador.

CPU: En un principio, *Dynamips* usará mucha cantidad de CPU porque no sabe cuándo el CPU virtual del enrutador está inactivo, por lo tanto, ejecuta todas las instrucciones de las rutinas de inactividad de la IOS como si fueran instrucciones que realizan un trabajo real. El cálculo del valor de IDLE-PC hará que el consumo de CPU del emulador baje drásticamente. Si se elige un buen valor, la utilización de CPU por cada enrutador será baja con lo cual el funcionamiento del emulador será óptimo.

Disco: Se necesita 39,65 MB de espacio de disco para almacenar a la aplicación GNS3 y a sus dependencias y emuladores asociados. Además se necesita 0,1 MB para almacenar WinPCAP, lo que hace un aproximado de 40 MB de disco necesario. Este parámetro no es determinante a la hora de escoger un buen PC donde montar nuestra red virtual debido a que para la mayoría de los PCs estos valores son fácilmente alcanzables.

1.10.2 Requerimientos del sistema en Linux

Ahora corresponde establecer cuáles son los requerimientos del sistema cuando se utiliza Linux:

Memoria RAM: en Linux, la memoria RAM requerida teórica para la emulación de un enrutador sería la que Dynamips asigna por defecto al compilador JIT (64MB) y la cantidad de RAM que cada imagen IOS requiere para funcionar en un equipo real, aunque, como ya se ha explicado, en la práctica se necesitan valores inferiores.

CPU: en Linux también se toma en cuenta el valor de IDLE-PC para estimar los requerimientos de CPU del emulador.

Disco: El espacio total necesario en disco para la instalación de GNS3 en Linux es de aproximadamente 117,2MB, valor que es mayor al requerido en Windows debido a la necesidad de instalación adicional de dependencias. Este parámetro no es determinante a la hora de escoger un buen PC de trabajo.

1.11 Empleo del GNS3

A continuación se explicará cómo hacer uso del GNS3, detallando los pasos que hay que seguir para la emulación de las diferentes plataformas CISCO soportadas, así como también se describirán los diferentes métodos existentes para la simulación de PCs, y se explicará cómo usar los conmutadores Ethernet.

1.11.1 Emulación de enrutadores CISCO

Como ya se ha indicado anteriormente, para emular enrutadores CISCO reales se necesita una imagen CISCO IOS perteneciente al enrutador que contiene las características que se quiera clonar. En este sentido, el emulador habilitará un número de ranuras o “*slots*” dependiendo del tipo de plataforma que se emula y en cada una de esas ranuras se podrán colocar solo ciertos tipos de adaptadores de interfaces.

Por lo tanto, si se quiere añadir capacidades de *hardware* en nuestro enrutador virtual, se debe seleccionar el tipo de adaptador de red que este pueda soportar en la configuración del enrutador virtual.

Pasos a seguir para la emulación y configuración de un enrutador en GNS3:

- 1) Arrastrar hasta el área de construcción de topología al enrutador que se quiera emular, como se muestra en la figura 1.3.

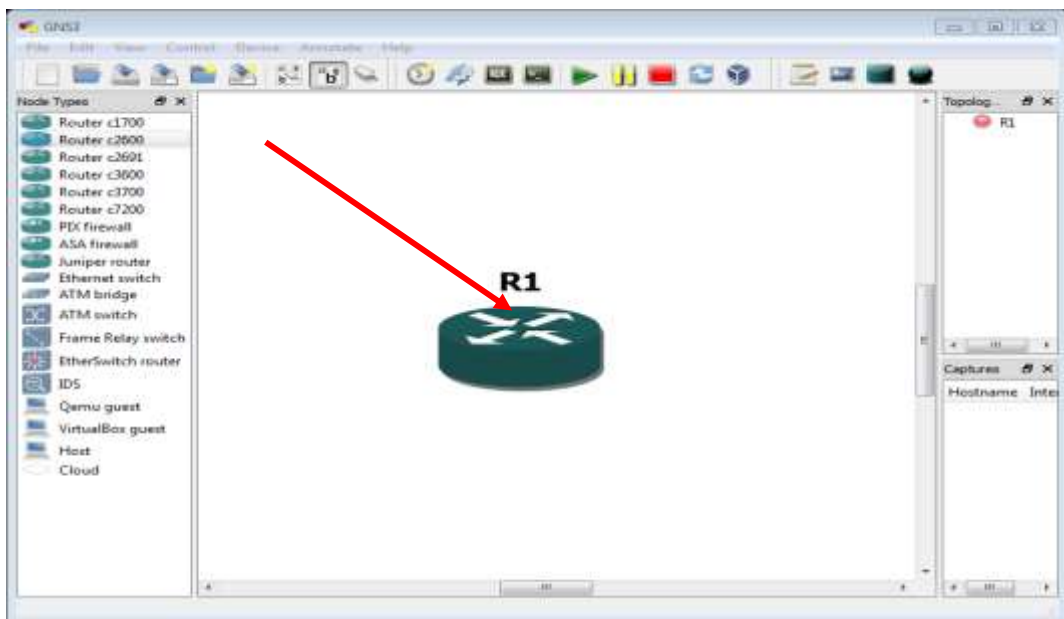


Figura 1. 3 Ventana principal de GNS3.

Ventana capturada por: Autor

- 2) Realizar clic derecho sobre el enrutador y elegir “*configure*” por lo que aparece la ventana mostrada en la figura 1.4.



Figura 1.4 Menú de opciones de un enrutador.

Ventana capturada por: Autor

3) Seleccionar el nombre del enrutador, después la pestaña “slots” y finalmente elegir las interfaces que se desean. Guardar los cambios seleccionando “OK”, de esta manera se obtiene la respuesta que se puede observar en la figura 1.5.

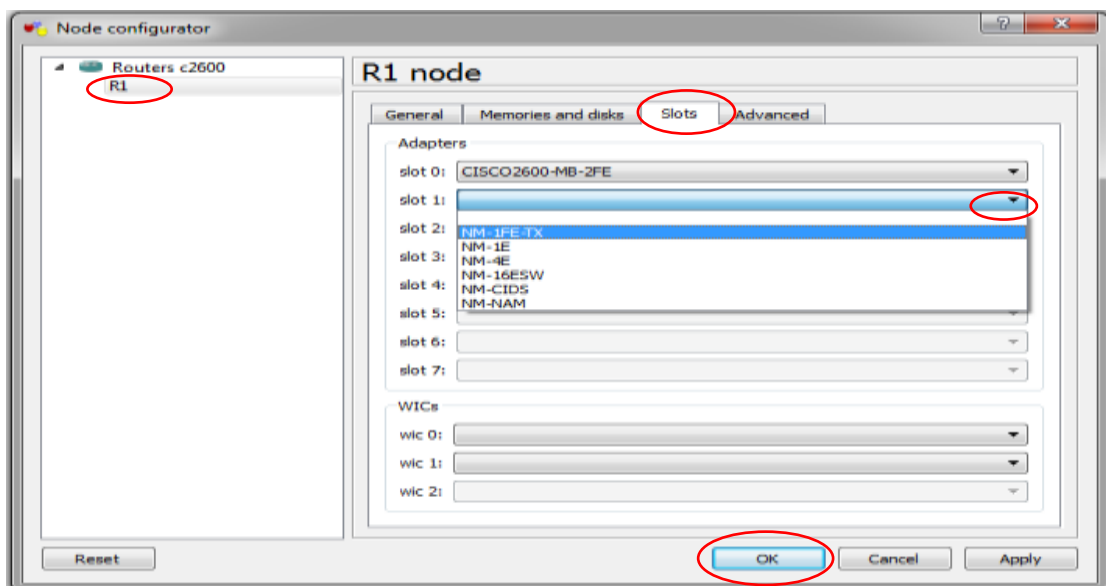


Figura 1.5 Ventana de configuración del nodo.

Ventana capturada por: Autor

4) Encender el enrutador eligiendo “start”.

5) Calcular el valor de IDLE PC para la imagen de IOS utilizada al realizar clic derecho en el enrutador y eligiendo “Idle PC” del menú desplegado.

6) Escoger un valor de los posibles de IDLE PC calculados, se recomiendan escoger los que tienen un * a la izquierda, elegir uno de ellos y seleccionar “*apply*”. Obteniéndose de esta manera la respuesta que se presenta en la figura 1.6 a continuación:

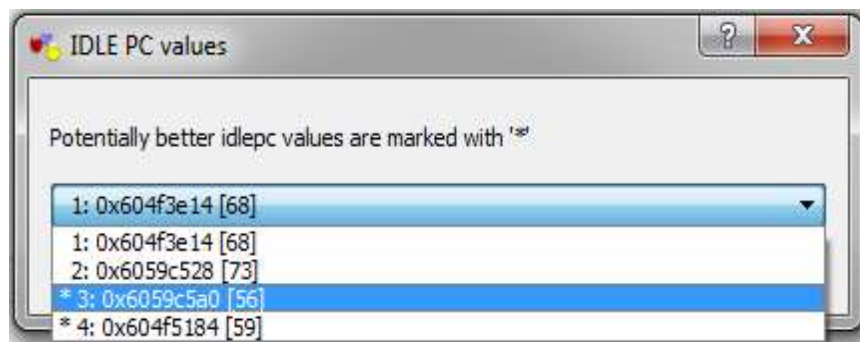


Figura 1. 6 Ventana de IDLE PC
Ventana capturada por: Autor

7) Finalmente elegir “*console*” para obtener una ventana de *Telnet* con la consola del enrutador que se ha emulado.

1.11.2 Emulación de conmutadores Ethernet

GNS3 posee integrada la capacidad de emulación de conmutadores Ethernet con funcionalidades básicas como la creación de VLANs o el funcionamiento del IEEE 802.1q.

Por defecto, un conmutador emulado con GNS3 tiene 8 puertos de acceso configurados en la VLAN1, pero se puede añadir hasta 10.000 puertos, pudiendo ser cada uno de ellos, un puerto de acceso o uno troncal.

En este sentido, si se desea trabajar con conmutadores que poseen más funcionalidades, GNS3 puede emular una tarjeta “EtherSwitch” que puede ser soportada solo por determinadas plataformas CISCO.

La tarjeta “EtherSwitch” que emula *Dynamips* es “NM-16ESW” y, puede ser incluida en casi todas las plataformas disponibles en GNS3.

La emulación y configuración de un conmutador Ethernet usando GNS3 se hace de la siguiente manera:

- 1) Arrastrar hasta el área de construcción de topologías el conmutador Ethernet situado en la parte izquierda de la ventana principal.
- 2) Realizar clic derecho sobre el conmutador y elegir “configure”.
- 3) Seleccionar el nombre del dispositivo y borrar la configuración inicial del conmutador Ethernet, para ello seleccionar cada puerto y luego “Delete”. Esta acción genera la ventana que se muestra en la figura 1.7.

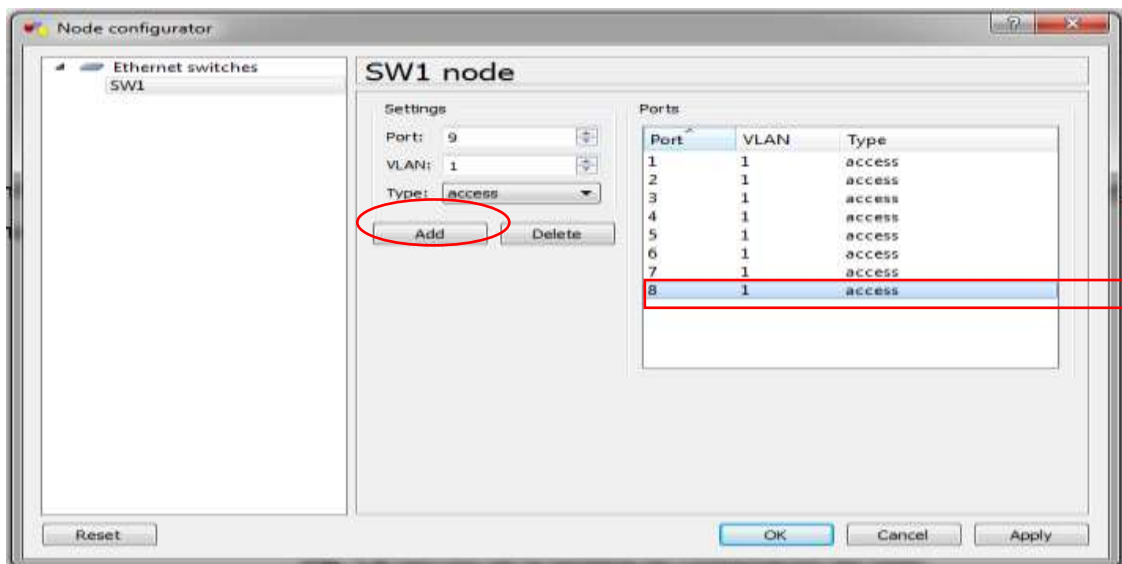


Figura 1. 7 Ventana de configuración del nodo.

Ventana capturada por: Autor

4) Configurar los nuevos puertos ingresando los parámetros de la sección “*Settings*” y adicionarlos. Finalmente guardar los cambios en “OK”.

5) Crear enlaces entre los diferentes dispositivos disponibles y cualquiera de las interfaces del conmutador que se añadió.

1.11.3 Simulación de PCs

GNS3 permite, además de los equipos de red, la incorporación de PCs en las topologías creadas, lo que facilita la comprobación y el estudio de las redes simuladas. Existen varias formas de simulación de PCs en GNS3, una de ellas es usando el programa *Virtual PC Simulator* (VPC) que usa puertos UDP para la comunicación entre el simulador y cada uno de los PCs simulados.

VPC es un programa que corre tanto en Windows como en Linux y que se puede descargar desde Internet de forma gratuita. Las ventajas de usar VPC es que su uso es simple y que no usa grandes cantidades de memoria ni ciclos de CPU para su funcionamiento; por otro lado, tiene la desventaja de que tiene funcionalidad limitada, ya que solo permite el uso de comandos como “*ping*” y “*traceroute*”, además soporta un máximo de nueve PCs simulados simultáneamente.

Pasos a seguir para la simulación de PCs utilizando este método:

1) Descargar e instalar el programa “*Virtual Pc Simulator*” disponibles en:

<http://wiki.freecode.com.cn/doku.php?id=wiki:vpcs>.

2) Escribir el carácter “?” para observar los comandos disponibles, y “*show*” para ver la configuración de red actual de los PCs simulados. Para cambiar de PC basta con escribir un número del 1 al 9 asignado a cada una de ellas.

3) Configurar los datos de cada PC ingresando el comando “*ip*” seguido de la dirección IP (192.168.0.2), la puerta de enlace por defecto del PC (192.168.0.1) y la máscara de subred (24). Escribir un número y luego seleccionar “*enter*” para cambiar de PC.

4) Arrastrar tantas nubes como PCs se quiera integrar al simulador, realizar clic derecho en cada uno de ellos y elegir “*configure*”. Seleccionar C0 debajo de “*clouds*” y elegir la pestaña “NIO UDP”.

5) Configurar los parámetros “*Local port*”, “*Remote host*” y “*Remote port*” debajo de “*Settings*” que corresponden a los puertos asignados por VPC para un PC virtual. Seleccionar “Add” y finalmente “OK”.

6) Repetir los pasos 4 y 5 en cada una de las otras nubes añadidas pero asignar valores correlativos a los mostrados anteriormente, tanto para el puerto local como para el remoto.

7) Añadir enlaces entre PCs y otros dispositivos con normalidad.

1.11.4 Elaboración de enlaces a módulos reales

Uno de los principales logros de Dynamips está relacionado con la comunicación de redes virtuales con el mundo real, dicha proeza se logra por medio de enlaces que tienen la capacidad de comunicar interfases de los enrutadores virtuales con las interfases de red reales del simulador, de modo que, se puede agregar a la topología, tantos equipos reales como adaptadores de red tenga el simulador.

Por lo tanto, los paquetes que salen de un enrutador virtual son colocados en red real a través del correspondiente adaptador de red del emulador asignado y los paquetes que entran son enviados de vuelta al enrutador virtual por el mismo adaptador.

En GNS3 los enlaces descritos anteriormente son simulados siguiendo los siguientes pasos:

1) Arrastrar la nube hasta el área de construcción de topologías.

2) Realizar clic derecho sobre la nube y elegir “*configure*” para abrir una ventana de configuración. Así se obtiene la ventana que se muestra a continuación en la figura 1.8:

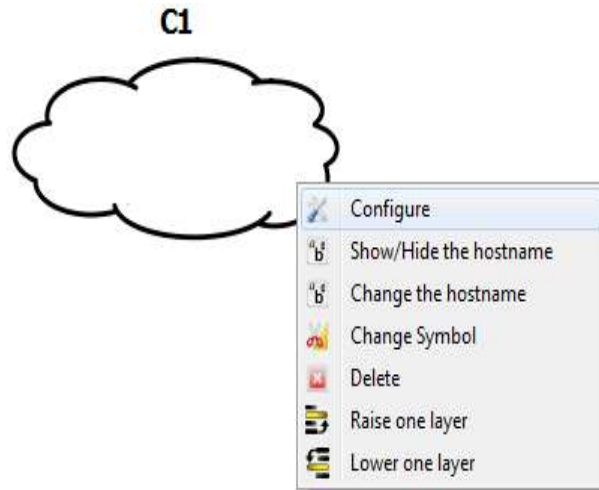


Figura 1. 8 Menú de opciones de una nube IP.

Ventana capturada por: Autor

3) Seleccionar el nombre de la nube IP, elegir la pestaña “NIO Ethernet” y seleccionar la caja que está debajo de “*Generic Ethernet NIO*” para observar todos los adaptadores de red reconocidos por el simulador. Seleccionar el que se desea usar para conectar el dispositivo externo y añadirlo a la topología, finalmente aceptar los cambios en “OK” y en seguida se podrá observar la ventana que aparece en la figura 1.9:

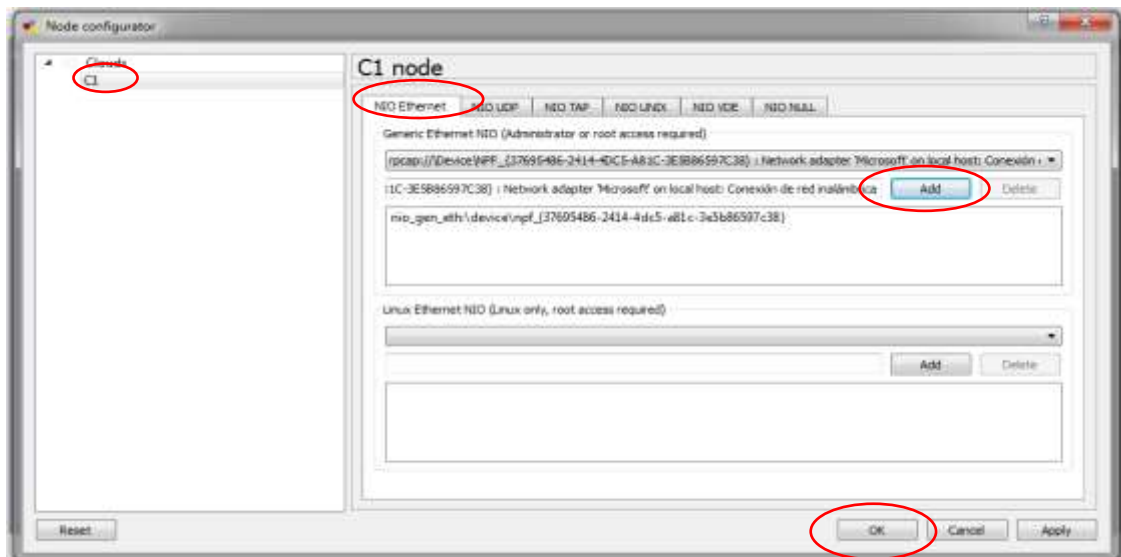


Figura 1. 9 Ventana de configuración de nodo.

Ventana capturada por: Autor

4) En el otro extremo del enlace, configurar la dirección IP necesaria para la conexión. Si el equipo externo es un PC, configurar una IP en su adaptador de red que se va a usar, además, si se trata de un enrutador, configurar la dirección IP desde la consola del equipo en la interfaz que se conectará al simulador. No olvidar elegir direcciones IP que mantengan concordancia con la topología a simular.

5) Realizar una conexión física entre el adaptador de red que ha sido elegido anteriormente en el simulador y el correspondiente al equipo externo.

6) Crear un enlace virtual entre un enrutador virtual y la nube creada.

1.11.5 Alojamiento de referencia de red

GNS3 accede el alojamiento de la referencia de red o en palabras técnicas similares a su topología y configuración efectuada en los enrutadores emulados en archivos diferentes; la topología se guarda, en forma de texto, en un archivo “.net” que es interpretado por *Dynagen* y mostrado gráficamente por GNS3; mientras que la configuración de los enrutadores emulados se guardan en archivos “.cfg”, los cuales abiertos con un editor de texto, muestran los comandos realizados en el enrutador de la misma forma que se pueden encontrar en un enrutador real.

El almacenamiento se realiza de la siguiente forma:

1) Al iniciar el programa aparecerá una ventana, en ella, habilitar el almacenamiento de las NVRAMs y archivos adicionales (*logfiles* y *bootfiles*) y de los archivos de configuración de todos los enrutadores, además asignar un nombre al proyecto. Elegir “OK”. Esta acción se muestra en la figura 1.10.

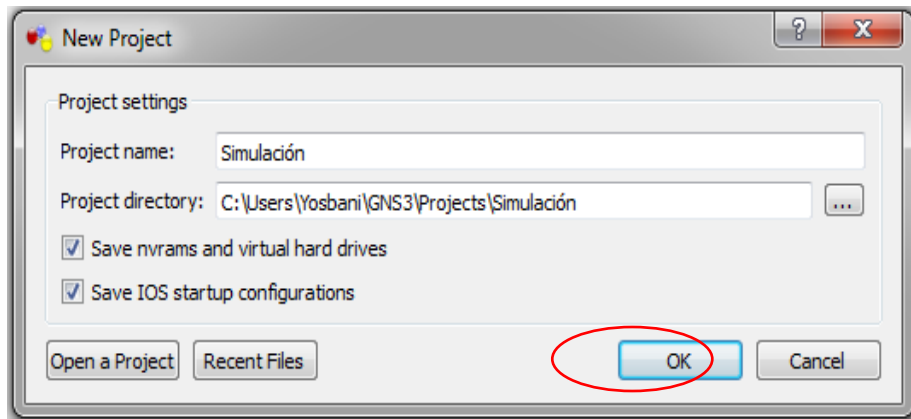


Figura 1. 10 Ventana del nuevo proyecto.

Ventana capturada por: Autor

2) Arrastrar un enrutador al área de creación de topologías, realizar clic derecho sobre él y elegir “*Change the hostname*”, a continuación escribir el nuevo nombre y seleccionar “OK”.

3) Guardar los cambios de configuración realizados en el enrutador usando el comando “wr” para almacenar los datos que se encuentran en la RAM a la NVRAM.

4) Ir a la barra principal de GNS3 y aparecerá una ventana, elegir “*Extracting to a directory*” para guardar los archivos de configuración de todos los enrutadores existentes en la topología. Elegir “Ok”, buscar la ubicación de la carpeta *simulación_config* y aceptar el destino. De esta manera se obtiene la ventana mostrada en la figura 1.11.



Figura 1. 11 Ventana de almacenamiento de configuraciones.

Ventana capturada por: Autor

5) Seguidamente se puede comprobar que en la carpeta *simulación_config* se ha creado un archivo “.cfg” que contiene la configuración del enrutador. No olvidar escribir el comando “wr” antes de volver a extraer la configuración.

6) Para guardar la topología dibujada, conexiones y el escenario elegir “File” y después “Save”.

1.11.6 Capacidad de captar datos

GNS3 integra la capacidad de capturar los paquetes que pasan por interfaces Ethernet o Serie y almacenarlos en archivos con formato *libpcap* para que puedan ser interpretados por aplicaciones como *Wireshark*, *tcpdump*, etc.

Pasos a seguir para realizar la captura de datos:

1) Descargar e instalar el programa *Wireshark* disponibles en: <http://www.wireshark.org>. En esta página se puede encontrar la versión tanto para Windows como para Linux.

2) En el menú principal elegir “Edit” y luego “Preferences”. Este procedimiento se muestra en la figura 1.12.

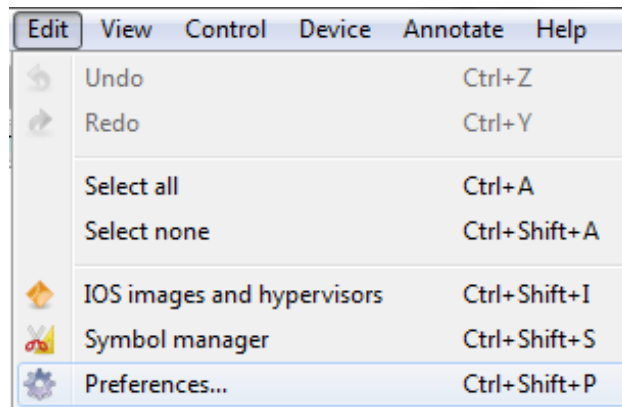


Figura 1. 12 Menú de opciones.

Ventana capturada por: Autor

3) Elegir “*Capture*” para obtener la pestaña relacionada con la configuración del proceso de captura del simulador. Comprobar que el camino indicado en *settings* es el correcto para el programa de captura que se va a usar, en este caso se trata de *Wireshark*. Para terminar seleccionar “OK”. Las acciones indicadas en este numeral se pueden apreciar en la figura 1.13

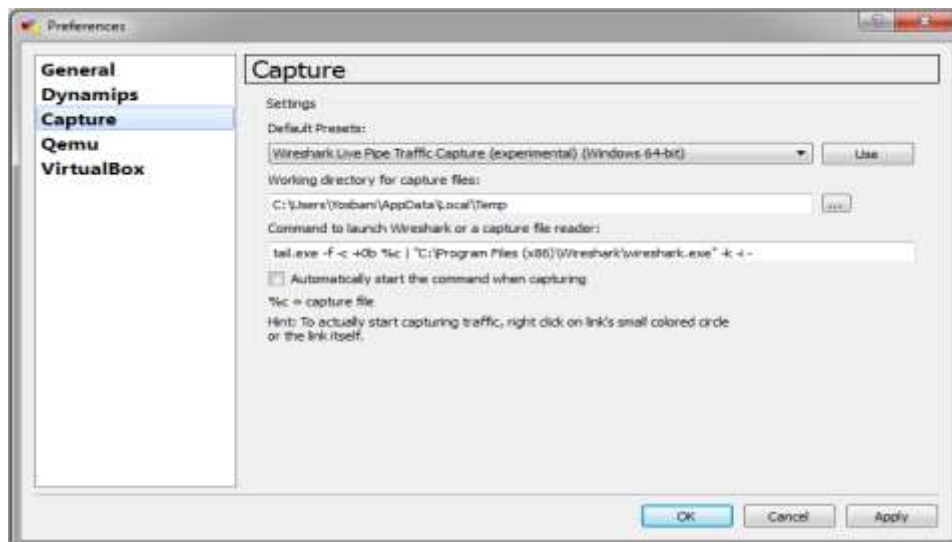


Figura 1. 13 Ventana de almacenamiento de configuraciones.

Ventana capturada por: Autor

4) Realizar clic derecho sobre cualquier parte del enlace del que se desea obtener una captura y seleccionar “*Start capturing*”. Los enlaces pueden ser Ethernet o serie y

pueden unir dos equipos Ethernet o uno Ethernet y otro Frame Relay, como puede observarse en la figura 1.14.

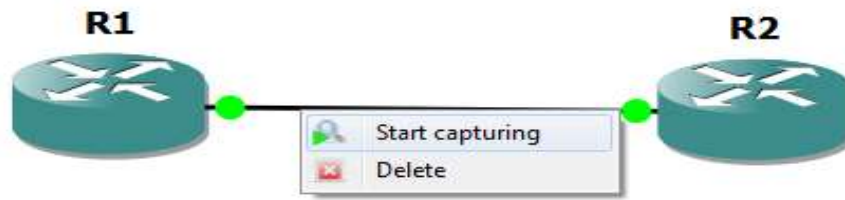


Figura 1. 14 Enlace Ethernet entre enrutadores
Ventana capturada por: Autor

5) A continuación aparecerá una ventana de captura. Desplegar el menú y elegir el enrutador que actuará como fuente de los paquetes enviados y la encapsulación que realiza dicho enrutador, obteniéndose la ventana que se presenta a continuación en la figura 1.15.

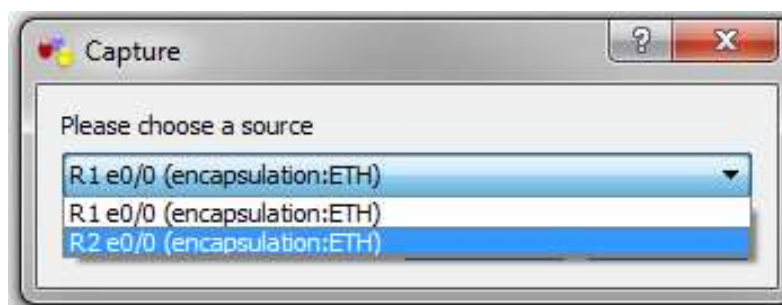


Figura 1. 15 Ventana de captura.
Ventana capturada por: Autor

6) Al cabo de unos segundos se tiene una ventana de *Wireshark* con los datos de la captura realizada.

1.11.7 Trabajo en forma *hypervisors*

GNS3 posee la capacidad de permitir que un PC cliente con *Dynagen* se comunique con otro PC externo que contiene al emulador *Dynamips* por medio de conexiones TCP/IP, es decir, el PC con *Dynagen* escuchará al emulador por un puerto TCP determinado.

A este modo de trabajo del simulador se le conoce como *hypervisors* y tiene como principal ventaja que permite la repartición del uso de recursos de procesamiento y memoria en más de un PC de la red, con lo cual, es posible simular topologías de gran tamaño.

Los equipos externos que contendrán al emulador *Dynamips* pueden trabajar tanto en Windows como en Linux, lo único que hay que tener presente es que se use correctamente la nomenclatura para la definición de los caminos hacia los directorios de trabajo, es decir, los directorios donde se guardarán los archivos generados.

Pasos a seguir para trabajar en modo *hypervisor*:

1) En la ventana principal del emulador, elegir “*Edit*” y después seleccionar “*IOS images and hypervisor*”.

2) Elegir la pestaña “*External hypervisors*” y añadir la dirección IP, el puerto TCP, los puertos base UDP y el de consola para la comunicación con el PC remoto, crear la carpeta de trabajo en el emulador *Dynamips* y colocar dicho camino en la configuración. Finalizar eligiendo “*Save*” y después “*Close*.” Esto se muestra en la figura 1.16.

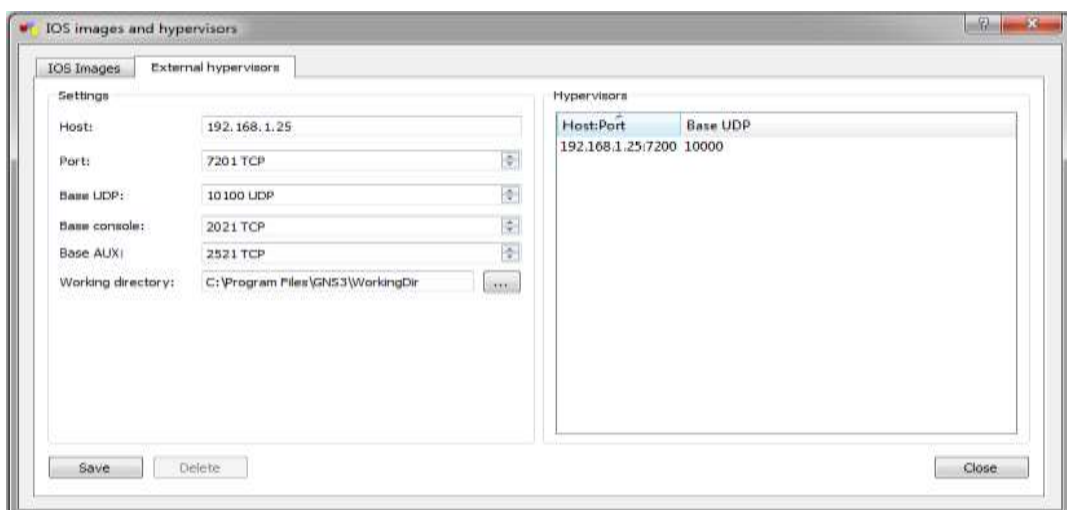


Figura 1. 16 Ventana de configuración de hypervisors.

Ventana capturada por: Autor

3) Seleccionar la pestaña “*IOS Images*” para unir los PCs remotos con una imagen de CISCO IOS correspondiente.

4) Copiar la imagen a simular en el PC remoto.

5) Escribir la ubicación de la IOS en la sección “*Image file*” junto con sus demás parámetros, después deshabilitar la opción “*Use the hypervisors manager*” y elegir un PC de la lista. Finalizar deshabilitando “*Default image for this plataforma*” eligiendo “*Save*” y “*Close*”. Esto se observa en la figura 1.17.

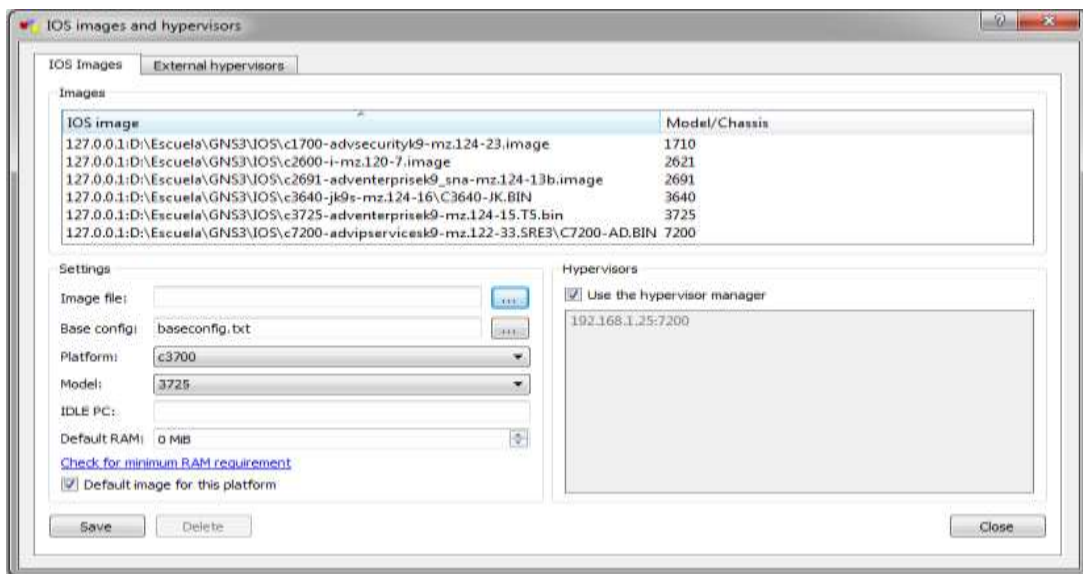


Figura 1. 17 Creación de un *hypervisors*.

Ventana capturada por: Autor

Lo que se ha logrado hasta ahora es que un enrutador con un determinado IOS se emule en un PC externo. También es posible conseguir que el consumo de memoria que produzca un enrutador se reparta entre varios PC externos simultáneamente dependiendo del nivel de dicho consumo en cada *hypervisors* externo, provocando un balanceo de carga.

6) Activar *Dynamips* en el PC remoto desde GNS3 o desde la ventana de comandos. Como se dijo, por defecto el PC escuchará el puerto 7200, si se usan varios *hypervisores*, es necesario cambiar este puerto para cada uno de ellos.

7) Cuando se arrastra un enrutador, el emulador nos pedirá elegir qué imagen nos gustaría usar.

1.12 Arquitectura del simulador eNSP

La plataforma de simulación de red eNSP (*Enterprise Network Simulation Platform*; Plataforma de Simulación de Red Empresarial), provisto por Huawei, es una plataforma gratis y extensible de simulación de red con gráficas e interfaces de usuario (GUIs). Simula computadoras (PCs), enrutadores y conmutadores, soporta simulación de red de gran escala y deja implementar pruebas experimentales y aprender tecnologías de la red sin usar dispositivos de red reales.

Además de la implementación del nodo, eNSP también soporta implementación distribuida. En este modo, el servidor del eNSP se separa en servidores múltiples, formando una red complicada. Los recursos en los servidores se ubican automáticamente. La consola se separa del ambiente de simulación donde corre el paquete de simulación del producto.

Los dispositivos virtuales pueden conectarse a los dispositivos reales conectando una interfaz virtual a un adaptador real de la red. Una red simulada se diferencia muy poco de una red física real, por lo que sirve como referencia y mejor comprensión de la misma.

Principales ventajas que presenta el eNSP y que han servido como punto de partida para tomar la decisión de estudiar este simulador:

- Es un programa libre de licencia, se puede descargar libremente de Internet.
- Con el eNSP se pueden simular redes que empleen dispositivos del fabricante Huawei.
- Permite conformar estructuras topológicas complejas, que incluyen enrutadores, conmutadores Ethernet, Frame Relay y ATM, máquinas virtuales, entre otros.

- Puede capturar los paquetes que pasan por enlaces virtuales y escribir los resultados de la captura en archivos, que pueden ser interpretados por aplicaciones como Wireshark.
- Posibilita la salva y ejecución posterior, tanto de las configuraciones de los enrutadores como del escenario implementado.

Principales desventajas que presenta el eNSP:

- Solo permite simular dispositivos del fabricante Huawei.
- Demanda una buena cantidad de memoria y CPU en la PC donde se instale, siendo directamente proporcional a la cantidad de equipos que emule. Es por ello que la PC requiere de buenas prestaciones.

1.13 Requisitos de configuración del sistema

Para poder instalar el simulador eNSP en una PC, estas deben cumplir algunos requisitos en su sistema. En la tabla 1.1 se muestra las características básicas necesarias para la instalación de esta herramienta.

Artículo	La Configuración Mínima	La Configuración Recomendable	La Configuración Expandida
CPU	Dual-core 2.0 GHz	Dual-core 2.0 GHz	Dual-core 2.0 GHz
Memoria (GB)	2	4	4 + n (n > 0)
Espacio libre del disco(GB)	1	2	2
Sistema operativo	Windows XP Windows Server 2003 Windows 7	Windows XP Windows Server 2003 Windows 7	Windows XP Windows Server 2003 Windows 7

El máximo número de enlazar dispositivos en red	4	8	$8 + 4 * n$
---	---	---	-------------

Tabla 1 Requisitos de configuración del sistema

Elaborada por: Autor

1.14 Uso del eNSP

A continuación se explicará claramente cómo hacer uso del eNSP, detallando los pasos a seguir para la simulación de los enrutadores Huawei, para la creación de una topología y realizar pruebas en ella, así como la captura de paquetes y la configuración de los dispositivos.

1.14.1 Configuración de enrutadores Huawei

Pasos a seguir para la configuración de un enrutador en el eNSP:

1) Arrastrar hasta el área de construcción de topología al enrutador, como se muestra en la figura 1.18

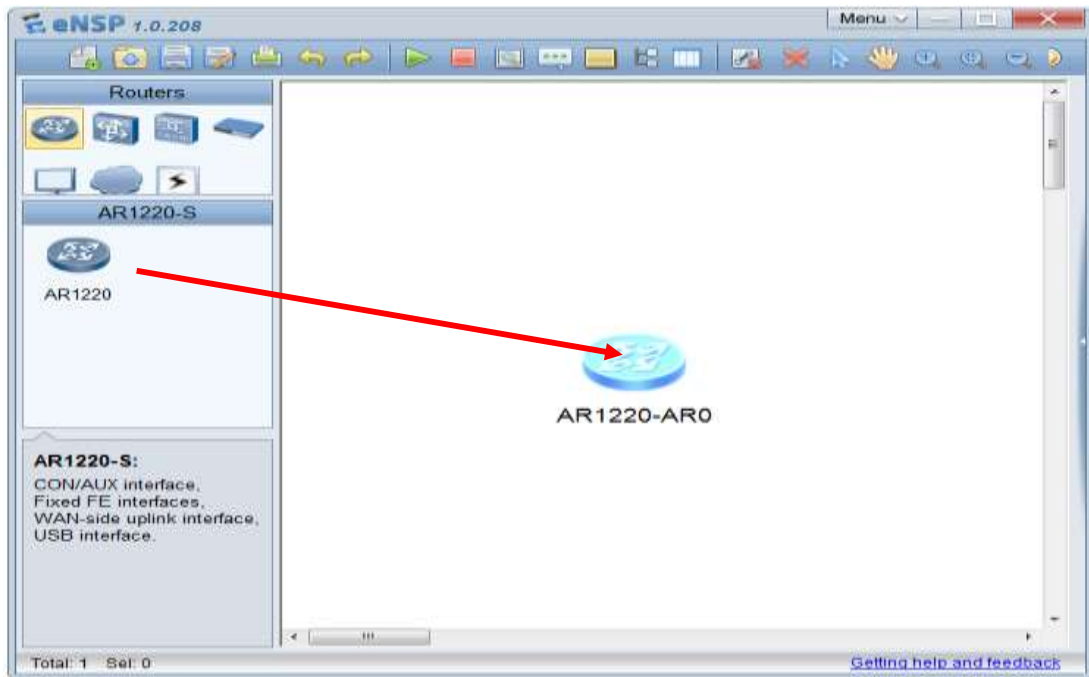


Figura 1. 18 Ventana principal del eNSP.

Ventana capturada por: Autor

2) Realizar clic derecho sobre el enrutador y elegir “Settings”, tal como se demuestra en la figura 1.19



Figura 1. 19 Menú de opciones de un enrutador.

Ventana capturada por: Autor

3) Escoger el número de puertos que desea tener en cada interfaz y seleccionar “Apply”, obteniéndose la ventana mostrada en la figura 1.20.

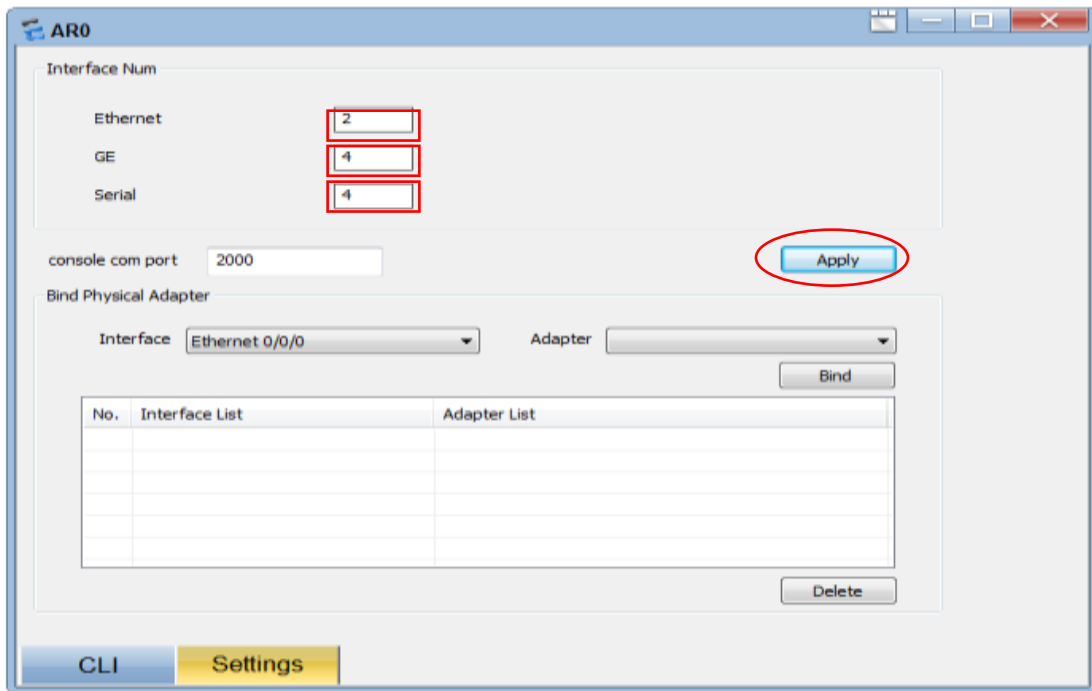


Figura 1. 20 Ventana de configuración del nodo.

Ventana capturada por: Autor

4) Encender el enrutadoreligiendo “Start”.

5) Realizar clic derecho sobre el enrutador y elegir “CLI” para entrar a la ventana de Telnet.

1.14.2 Crear una topología en el eNSP

Un experimento de la red se realiza basado en una topología. Esta sección describe cómo crear una topología simple en el eNSP con un conmutador de tramas Ethernet y dos PCs.

Pasos para crear una topología:

1) Iniciar el cliente del eNSP.

2) Añadir un conmutador y dos PCs al área de trabajo.

➤ Seleccionar el conmutador en el área de tipo de dispositivo.

- Establecer el dispositivo modelo para S5700.
- Arrastrar el dispositivo para el área de trabajo.
- Repetir los pasos anteriores para añadirle dos PCs al área de trabajo.

3) Conectar las dos PCs al conmutador usando los cables de red.

- Seleccionar el tipo de conexión en el área de tipo de dispositivo.
- Establecer la conexión “Auto”.
- Enlazar el conmutador y una PC en el área de trabajo.
- Enlazar el conmutador y la otra PC en el área de trabajo.

El resultado de estas acciones se muestra a continuación en la figura 1.21

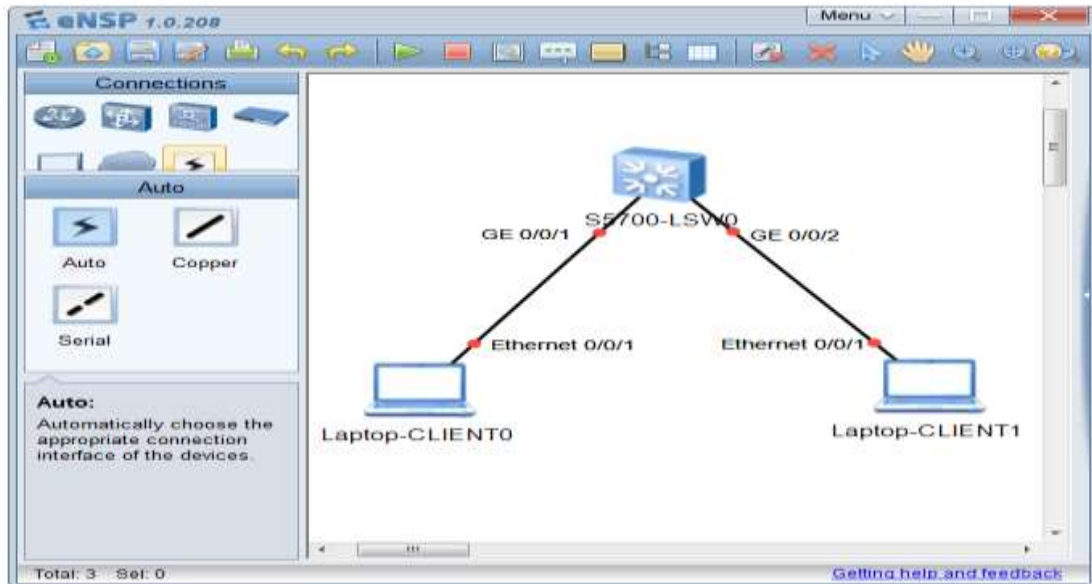


Figura 1. 21 Escenario del eNSP con un conmutador Ethernet y 2 PC.

Ventana capturada por: Autor

4) Encender los dispositivos seleccionando “*Start Device*” en el área de trabajo.

5) Realizar doble clic sobre el conmutador en el área de trabajo para abrir el CLI. Las configuraciones principales son realizadas ejecutando comandos en el CLI.

6) Seleccionar “*Save*” en la barra de herramientas para salvar la topología.

1.14.3 Captura de datos

El eNSP integra la capacidad de capturar los paquetes que pasan por interfaces Ethernet o Serie y almacenarlos en archivos, para que puedan ser interpretados por aplicaciones como *Wireshark*.

Pasos a seguir para realizar la captura de datos:

1) Iniciar el cliente del eNSP.

2) Seleccionar “*Open*” en la barra de herramientas para abrir la topología creada.

3) Seleccionar “*Start Device*” en la barra de herramientas para encender todos los dispositivos.

4) Realizar clic derecho sobre el conmutador, escoger “*Capture Data*” entre el menú desplegado y seleccionar una interfaz que el *Wireshark* comience a capturar paquetes, como se observa en la figura 1.22

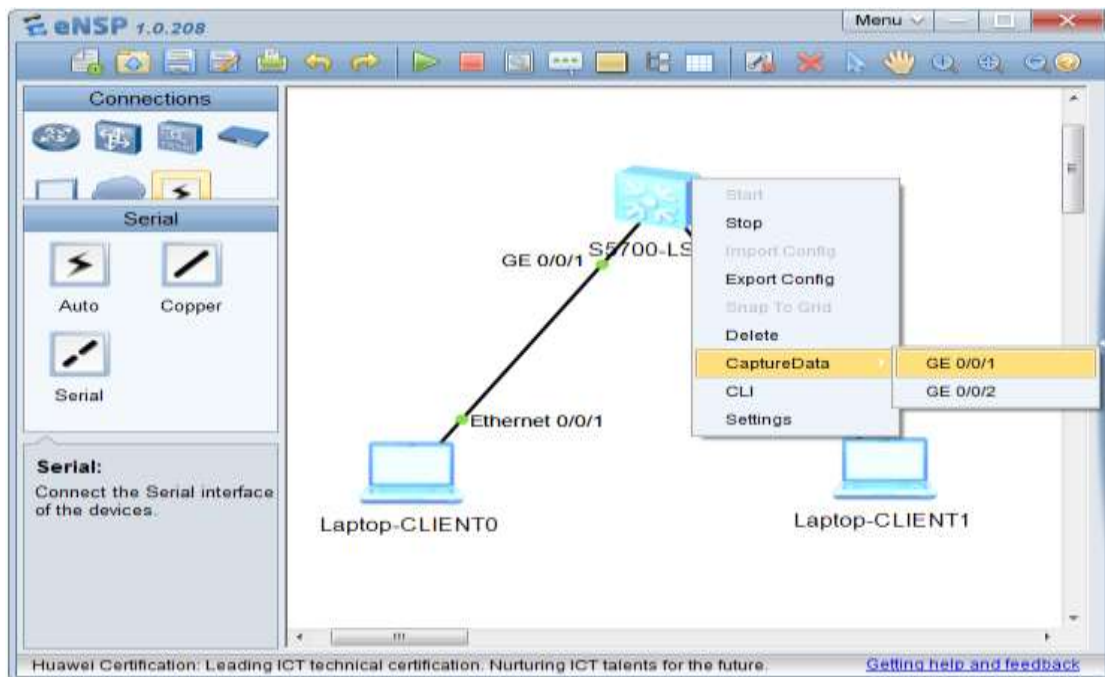


Figura 1. 22 Escenario del eNSP con un conmutador Ethernet y 2 PC.

Ventana capturada por: Autor

5) Realizar un *ping* de la PC que capta los paquetes para la otra PC. Observela captura obtenida con la ayuda del programa *Wireshark*. También se puede enviar un paquete de Protocolo de Resolución de Direcciones (ARP) y observar el paquete capturado.

6) Para dejar de capturar paquetes, realizar clic derecho en el conmutador, escoger “*Capture Data*” en el menú desplegado, y reelegir la misma interfaz antes seleccionada.

6) Salvar la topología y abandonar el programa.

1.14.4 Configuración del servidor y el cliente

A continuación se describe cómo configurar el servidor y el cliente en el modo de implementación distribuida.

Configuración del servidor

El servidor provee el servicio del eNSP basado en la dirección IP y número de puerto. Descripción de la configuración del número de puerto:

- 1) Abrir el archivo “*config sim.ini*” en el “*SimServer*” del directorio de instalación del programa.
- 2) Entrar en el número de puerto junto a “PORT =” y salvar el archivo.
- 3) Volver a arrancar el programa del servidor para hacer el ajuste.

Configuración del cliente

- 1) Escoger “*Menu*”, “*Tools*” y “*Options*” en la interfaz principal. Seleccionar la etiqueta “*Server Config*” en la ventana que es exhibida.
- 2) Un solo sistema del nodo provee la dirección IP predeterminada y el número de puerto que se miró en el área de configuración del servidor local. Colocar el número de puerto en el archivo “*config sim.ini*” según lo solicitado y vuelva a arrancar el programa del servidor para hacer el ajuste.

Lo detallado permite generar la ventana que se muestra a continuación en la figura 1.23:

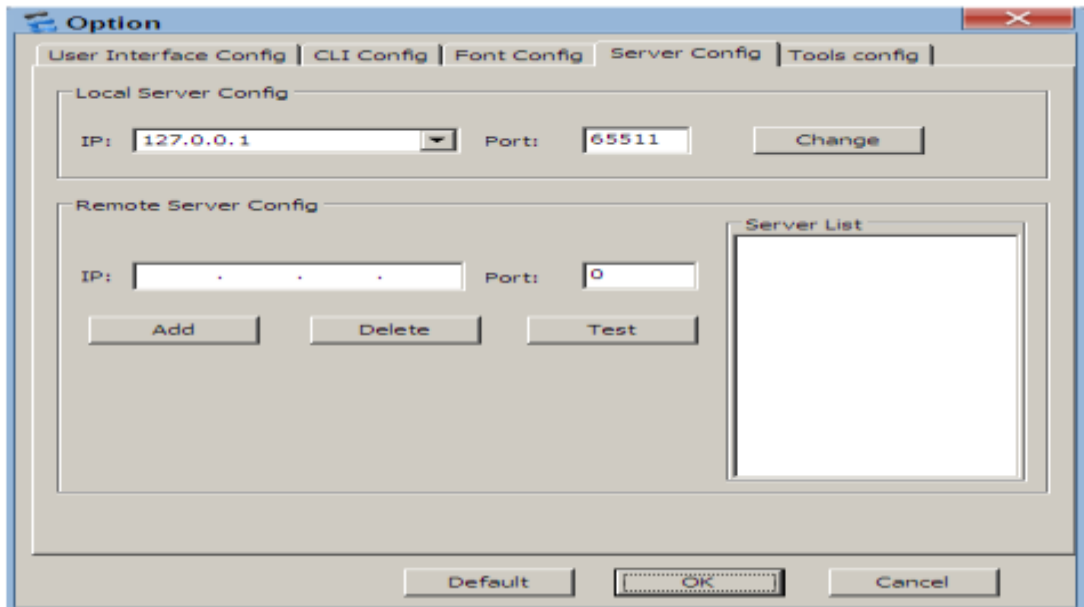


Figura 1. 23 Ventana de configuración del nodo.

Ventana capturada por: Autor

3) Configurar el servidor remoto cuando el modo distribuido de implementación es usado. Seleccione “*Test*” después de poner la dirección IP y el número de puerto en el área remota “*Server Config*”. Cuando la prueba es completa, elegir *Add* para añadirle el servidor a la lista del servidor.

1.14.5 Importación o exportación de un archivo de configuración

En este epígrafe se describirán los pasos a seguir para lograr la importación o exportación de un archivo de configuración.

Pasos a seguir para la importación de un archivo de configuración:

1) Antes de encender un dispositivo (conmutador o enrutador), realizar clic derecho en el dispositivo y seleccionar “*Import Config*”, como se presenta continuación en la figura 1.24



Figura 1. 24 Menú de opciones del enrutador.

Ventana capturada por: Autor

2) Seleccione el archivo de configuración (.cfg o .zip) del dispositivo e importe el archivo para el dispositivo.

3) El archivo de configuración importado estará cargado la próxima vez que el dispositivo arranque.

Pasos a seguir para la exportación de un archivo de configuración:

1) Encender el dispositivo y realizar doble clic sobre el dispositivo para abrir la página CLI.

2) En la página CLI, corra el comando “save” para salvar la configuración.

3) Realizar clic derecho sobre el dispositivo y seleccione “Export Config”, tal como puede apreciarse en la figura 1.25

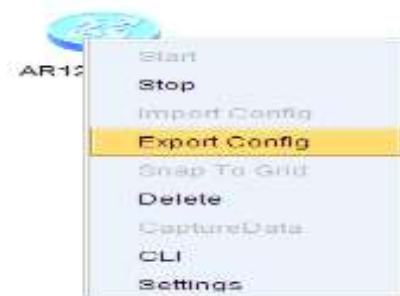


Figura 1. 25 Menú de opciones del enrutador.

Ventana capturada por: Autor

4) Introduzca el nombre del archivo de configuración y exporte la configuración del dispositivo en un archivo .cfg.

CAPÍTULO 2. LABORATORIOS VIRTUALES USANDO EL GNS3 Y EL ENSP

Dadas las potencialidades de los programas GNS3 y eNSP antes mencionadas, se considera estos programas apropiados para la creación y empleo de escenarios virtuales con fines docentes, como recursos didácticos de apoyo a la carencia de maquetas elaboradas con equipamiento real, en los cursos de transmisión de datos.

El uso del GNS3 y el eNSP resultan de gran utilidad, específicamente en temas donde se tratan los elementos básicos de transmisión de datos, en contenidos relacionados con el funcionamiento de los enrutadores, el estudio de varios protocolos y su operación por capas en el modelo OSI (*Open System Interconexión*) de la ISO (*International Standardization Organization*), como son: ARP, Frame-Relay, ATM, Ethernet, SMTP, HTTP, FTP, SNMP, ICMP y UDP.

2.1 Aplicación del GNS3

Este epígrafe aborda sobre la creación y empleo de escenarios virtuales con fines docentes, utilizando el emulador de redes GNS3. Se realizó un estudio de varios protocolos empleados en estos escenarios con la ayuda del *Wireshark*.

2.1.1 Escenario del GNS3 con 2 enrutadores y 2 PCs

En la figura 2.1 se muestra, a manera de ejemplo, un escenario de prueba en el GNS3. Este está constituido por 2 enrutadores (R1 y R2) interconectados entre sí por sus interfaces Ethernet 0/1.

El enrutador R1 se conecta a su vez por la Ethernet 0/0 a una interfaz *Loopback* configurada convenientemente en la propia PC donde corre el GNS3 (a la izquierda en la figura 2.1).

Mientras que el enrutador R2 se conecta por la Ethernet 0/0 a la interfaz

asociada a una máquina virtual de Windows, emulada en la PC con ayuda del programa VirtualBox de Sun (a la derecha en la figura 2.1).

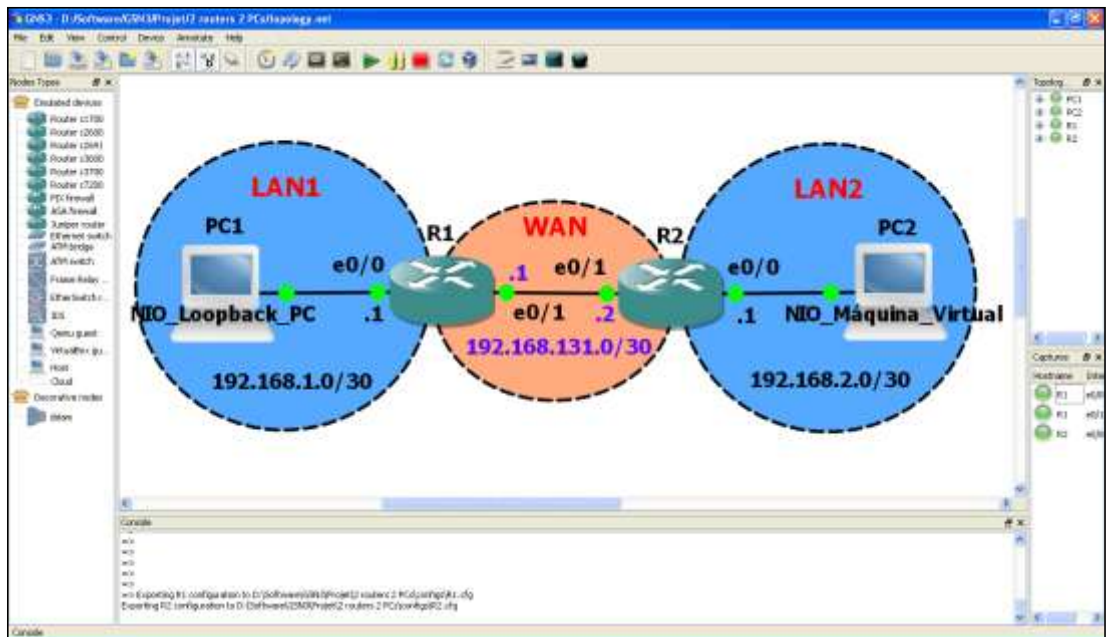
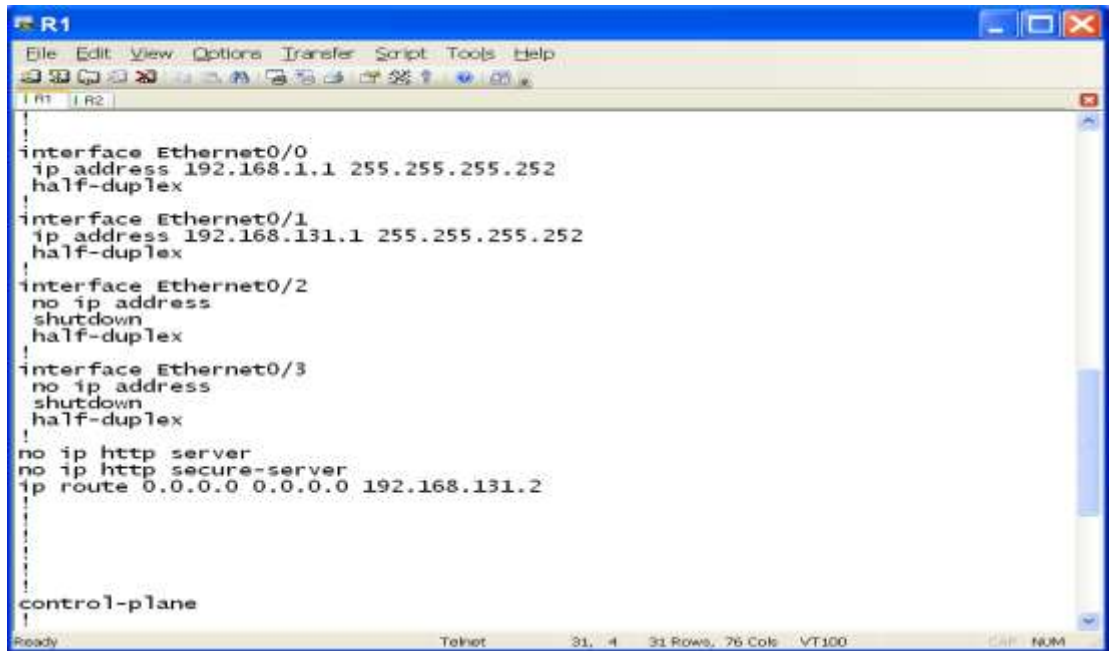


Figura 2. 1 Escenario del GNS3 con 2 enrutadores y 2 PCs.

Ventana capturada por: Autor

En la figura 2.2 se muestra la configuración del enrutador R1 del escenario mostrado en la figura 2.1, utilizando para ello el cliente Telnet SecureCRT. Este último no es un programa libre, pero si se dispone de él, es posible utilizarlo con GNS3, declarándolo adecuadamente en su configuración.



```
interface Ethernet0/0
ip address 192.168.1.1 255.255.255.252
half-duplex
interface Ethernet0/1
ip address 192.168.131.1 255.255.255.252
half-duplex
interface Ethernet0/2
no ip address
shutdown
half-duplex
interface Ethernet0/3
no ip address
shutdown
half-duplex
!
no ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 192.168.131.2

control-plane
```

Figura 2. 2 Configuración del enrutador R1 visto con el cliente Telnet de SecureCRT

Ventana capturada por: Autor

Para ver la configuración completa de los dispositivos empleados en este escenario consultar el anexo I.

A la izquierda de la figura 2.3 se muestra el empleo del uso del programa Simulador de Multiservidores de Paessler, corriendo sobre la PC1 (es la misma PC donde se emulan el GNS3 y la máquina virtual PC2).

Este está disponible gratis en Internet y resulta útil para demostrar los contenidos relacionados con los protocolos HTTP, SMTP, FTP y SNMP. A la derecha de la figura 2.3 se muestra cómo se accede desde la máquina virtual (PC2) a la página WEB configurada en la PC1, con ayuda del Servidor HTTP incorporado en el Simulador de Multiservidores Paessler.

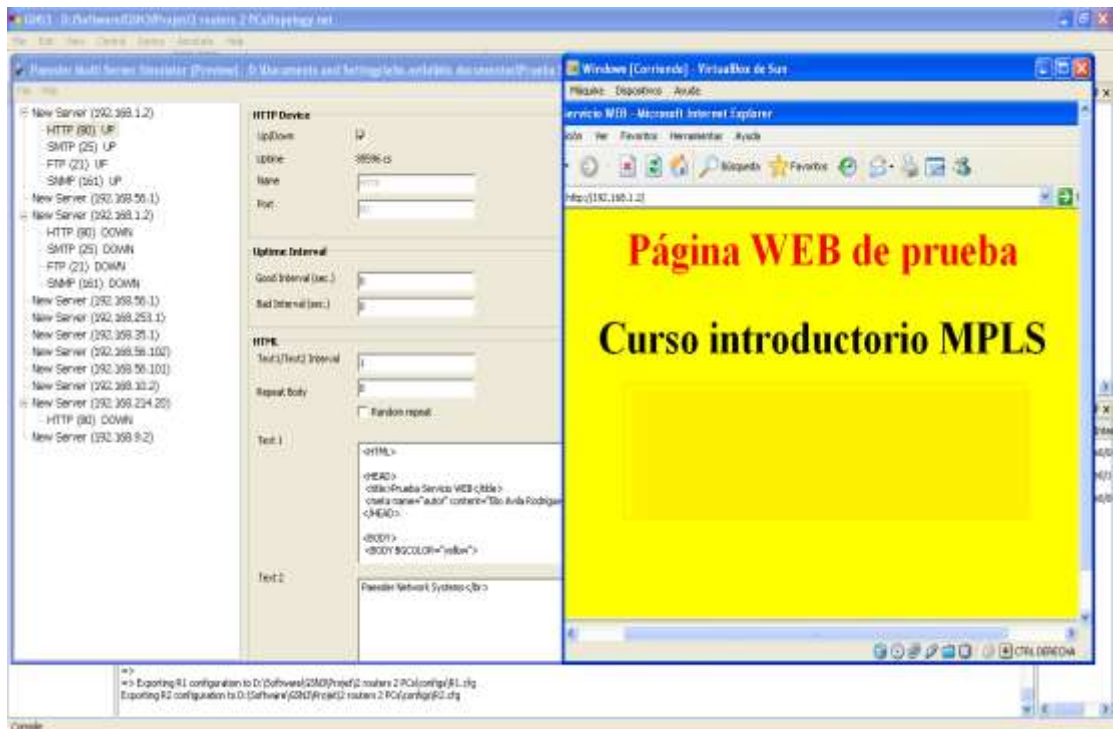


Figura 2. 3 Uso del Multiservidor Paessler (izquierda) y una máquina virtual VirtualBox de Sun.

Ventana capturada por: Autor

En la figura 2.4 se muestra una ventana de captura de tráfico en el escenario antes mencionado, posible de obtener con la ayuda del *Wireshark*.

En la parte superior de la figura 2.4 (Zona 1) es posible definir varios filtros de búsqueda, para visualizar aquellos paquetes o protocolos que resulten de interés. Consecutivo a esta en la (Zona 2) aparece la lista de visualización de todos los paquetes que se estén capturando en tiempo real por la interfaz seleccionada (tipo de protocolo, números de secuencia, banderas, marcas de tiempo, puertos, etc.).

Posteriormente se encuentra una franja (Zona 3), que permite desglosar por capas cada una de las cabeceras de los paquetes seleccionados en la zona 2 y facilita la revisión de cada uno de sus campos. Por último la (Zona 4) donde se muestra el paquete en bruto (en formato hexadecimal); es decir, tal y como fue capturado en la interfaz en cuestión (Merino, 2011)

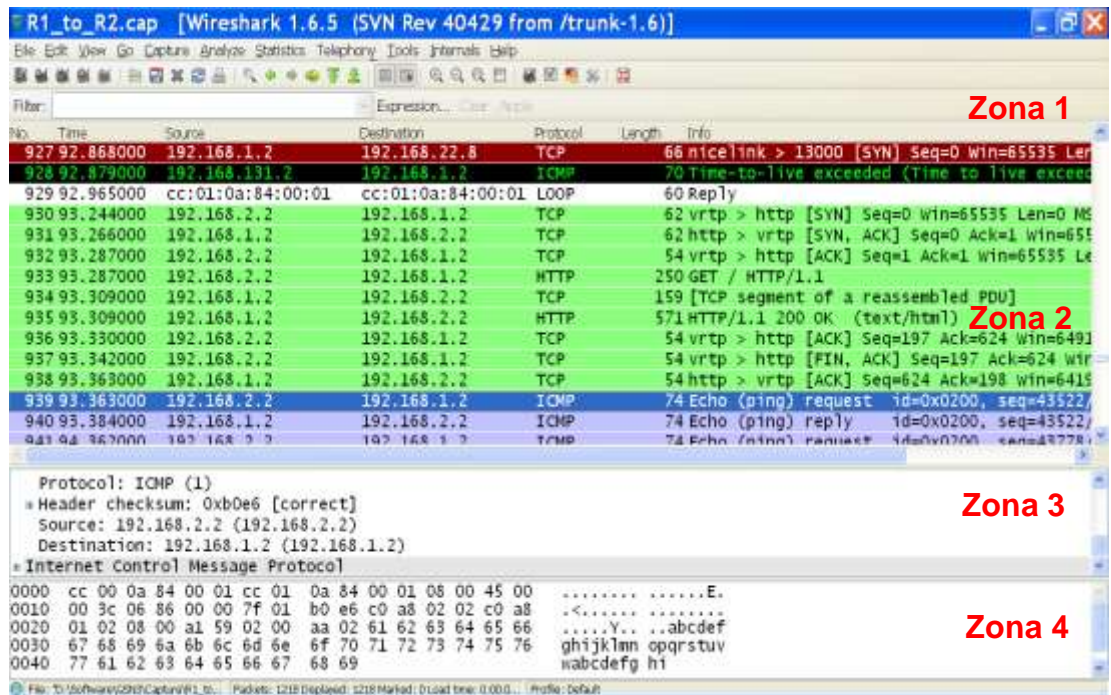


Figura 2. 4 Muestra de captura con el *Wireshark* al realizar un ping a 192.168.2.2.

Ventana capturada por: Autor

2.1.2 Escenario del GNS3 usando el protocolo VRRP

En la figura 2.5 se muestra, un escenario de prueba del GNS3 usando el protocolo VRRP (*Virtual Router Redundancy Protocol*; Protocolo Virtual de Redundancia de Enrutamiento). Este está constituido por 3 enrutadores (R1, R2 y R3) Los enrutadores R1 y R2 se conectan por la interfaz Ethernet 0/0 a un conmutador y este a una PC(C4) virtual. Mientras que el enrutador R3 se conecta por la interfaz Ethernet 0/0 al enrutador R1 y por la interfaz Ethernet 0/1 al enrutador R2. R1 se conecta por la interfaz Ethernet 0/2 a una PC (C3) virtual para cerrar la red.

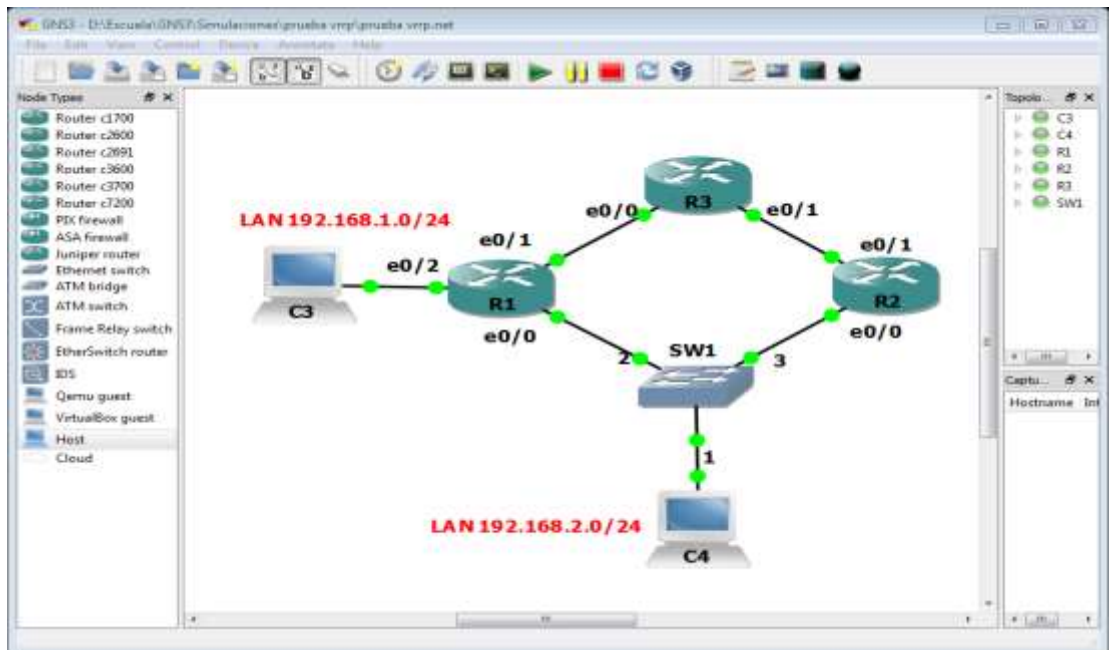


Figura 2. 5 Escenario del GNS3 usando el protocolo VRRP.

Ventana capturada por: Autor

En esta prueba se puede analizar el comportamiento de estos dispositivos de red, realizar capturas en cualquiera de sus enlaces y observar los protocolos que estén corriendo.

Las figuras 2.6 y 2.7 muestran una ventana de captura de tráfico del *Wireshark* al realizar un ping a la dirección 192.168.2.2 y un tracer a la dirección 192.168.1.1 respectivamente.

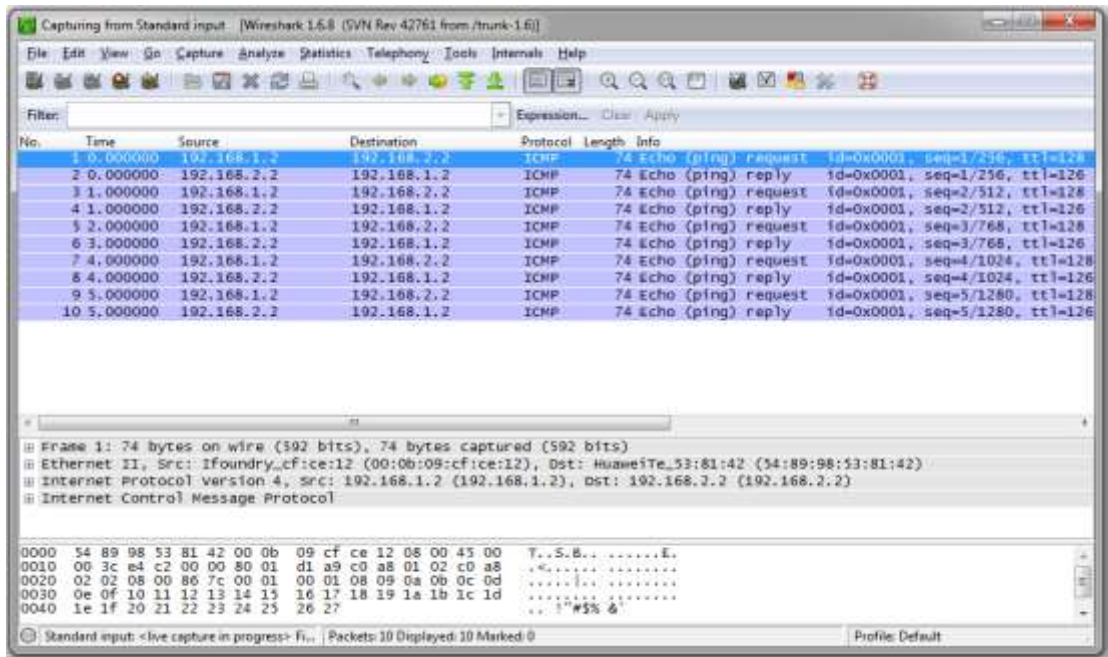


Figura 2. 6 Muestra de captura con el *Wireshark* al realizar un ping a 192.168.2.2.

Ventana capturada por: Autor

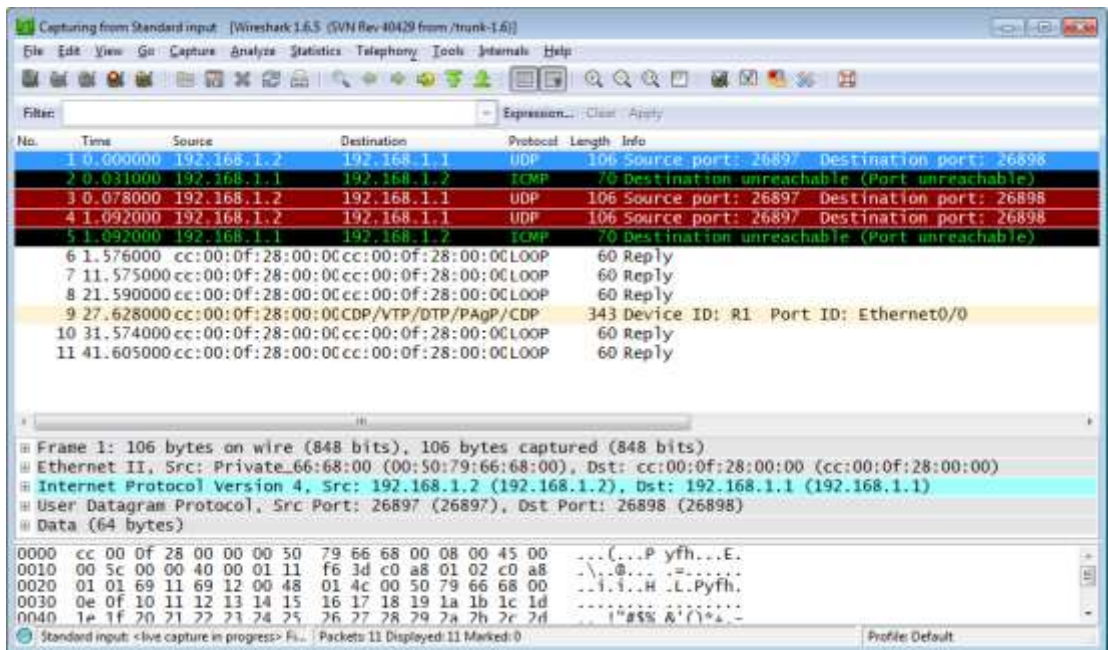


Figura 2. 7 Muestra de captura con el *Wireshark* al realizar un tracer a 192.168.1.1.

Ventana capturada por: Autor

Para ver la configuración completa de los dispositivos empleados en este escenario consultar el anexo II.

2.1.3 Escenario del GNS3 con una red IP/MPLS

En la figura 2.8 se muestra un escenario de prueba del GNS3 con una red IP/MPLS. El mismo está constituido por siete enrutadores (R1, R2, R3, R4, R5, R6 y R7) y cuatro PC (PC1, PC2, PC3 y PC4). En este escenario se muestran dos VPNs (*Virtual Private Network*, Red Privada Virtual), en las cuales con la ayuda del programa *Wireshark* nos permite realizar un estudio más amplio de la operación de la arquitectura IP/MPLS, al detallar el comportamiento de cada paquete en la red y el funcionamiento de los protocolos que estén corriendo.

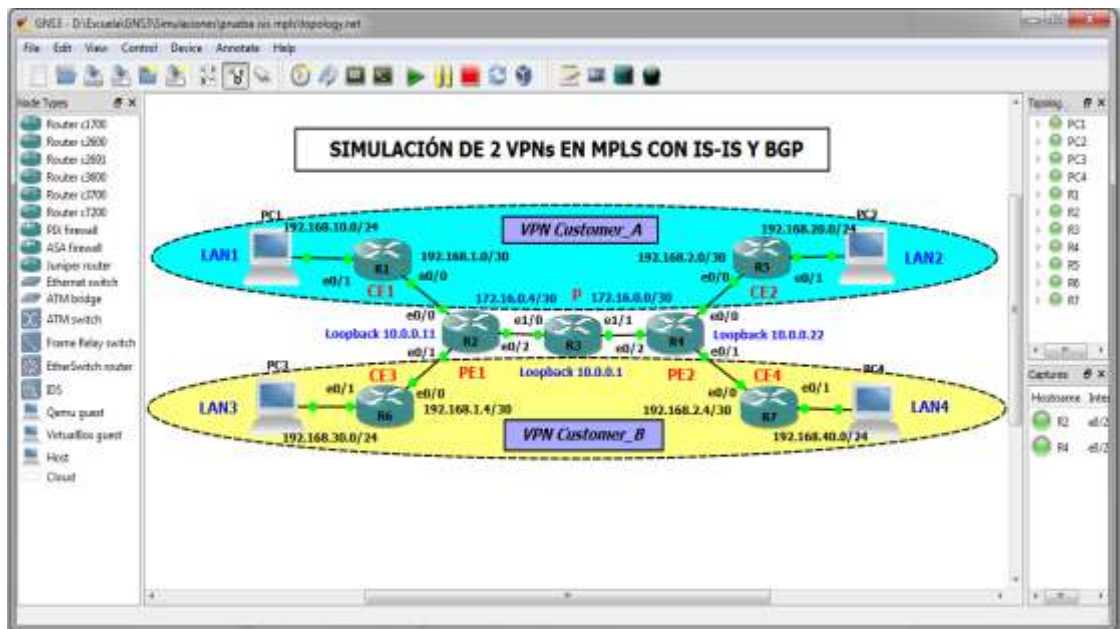


Figura 2. 8 Escenario del GNS3 con una red IP/MPLS.

Ventana capturada por: Autor

En la figura 2.9 se muestra una ventana de captura de tráfico posible de obtener con la ayuda del *Wireshark* en una de las interfaces del enrutador P o LSR (*Label Switching Router*) que constituye el núcleo de la red MPLS, a través de esta captura es posible demostrar con mayor claridad cómo trabajan los protocolos IS-IS (*Intermedia System - Intermedia System*) y BGP (*Border Gateway Protocol*), entre otros aspectos.

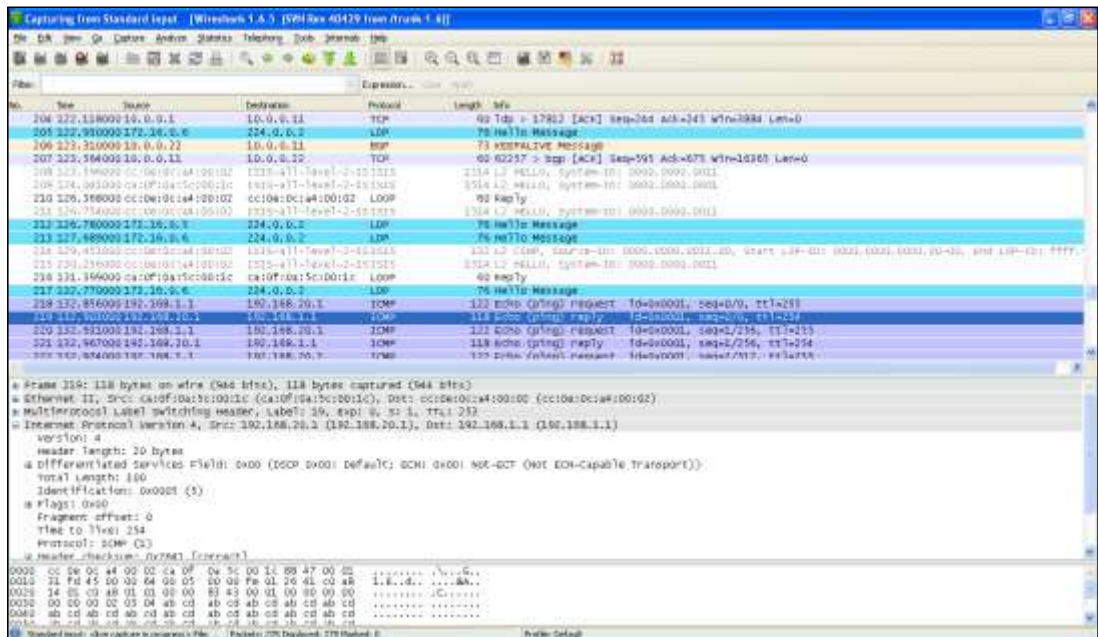


Figura 2. 9 Captura de tráfico con el *Wireshark* en el escenario IP/MPLS.

Ventana capturada por: Autor

Para ver la configuración completa de los dispositivos empleados en este escenario consultar el anexo III.

2.2 Aplicación del eNSP

Este epígrafe aborda sobre la creación y empleo de escenarios virtuales con fines docentes, utilizando el simulador de redes eNSP. Se realizó un estudio de varios protocolos empleados en estos escenarios con la ayuda del *Wireshark*.

2.2.1 Escenario del eNSP con 2 enrutadores y 2 PCs

En la figura 2.10 se muestra un escenario de prueba del eNSP utilizado. Este está constituido por 2 enrutadores interconectados entre sí por sus interfases Ethernet 0/0/1. Los 2 enrutadores se conectan por la interfaz Ethernet 0/0/0 a las PCs respectivamente, máquinas virtuales de Windows.

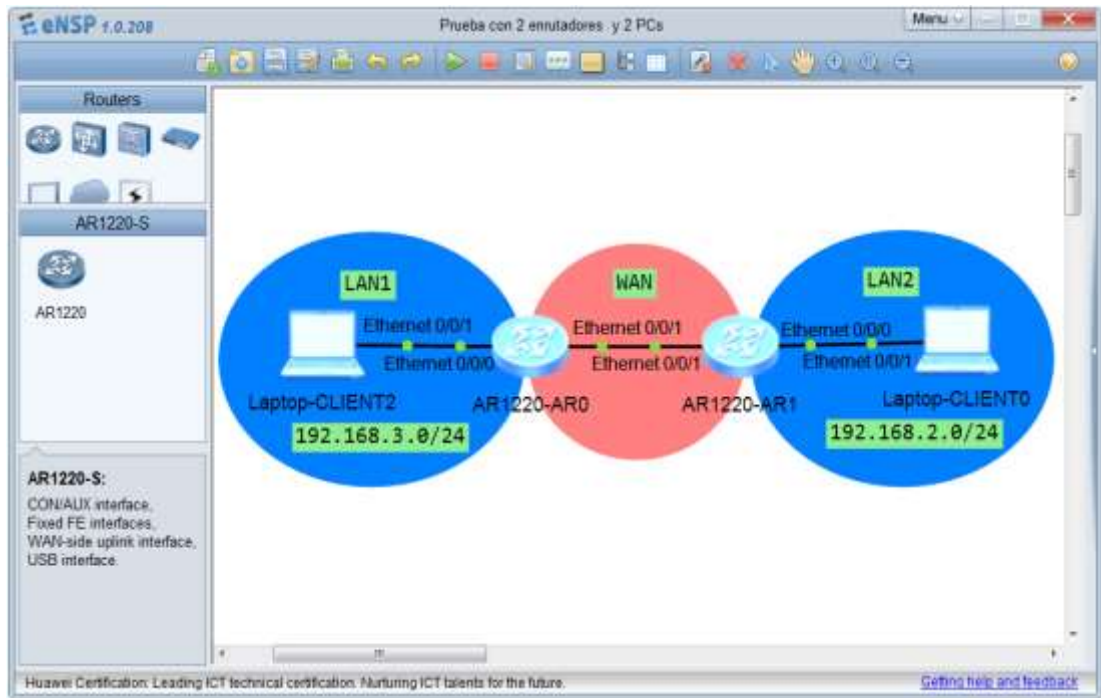


Figura 2. 10 Escenario del eNSP con 2 enrutadores y 2 PCs.

Ventana capturada por: Autor

En esta prueba se puede analizar el comportamiento de estos dos enrutadores, realizar capturas de tráfico en sus enlaces y observar los protocolos que estén corriendo.

En las figuras 2.11 y 2.12 se muestra una ventana de captura de tráfico en el escenario antes mencionado, posible de obtener con el *Wireshark* al realizar un ping a la dirección 192.168.2.2 y un tracert a la dirección 192.168.3.2 respectivamente.

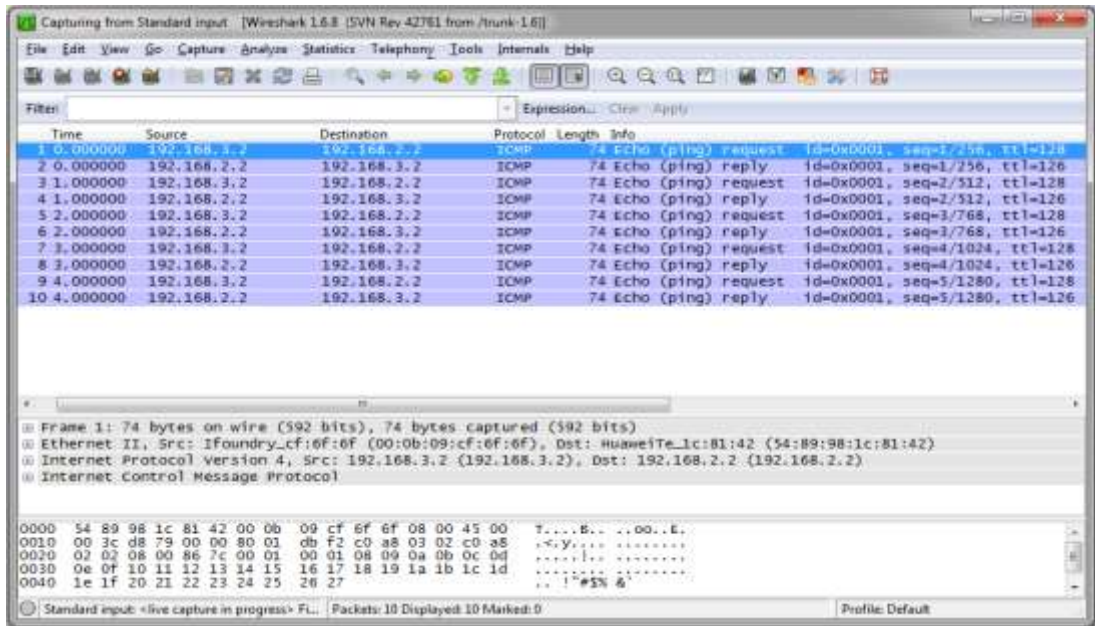


Figura 2. 11 Muestra de captura con el *Wireshark* al realizar un ping a 192.168.2.2.

Ventana capturada por: Autor

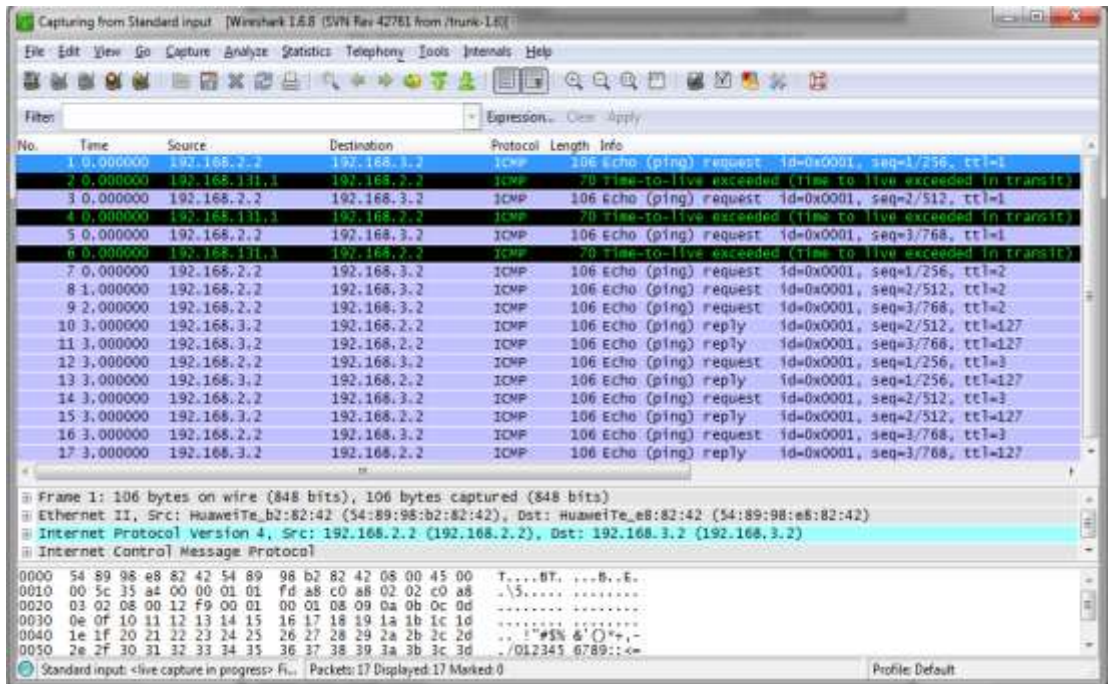


Figura 2. 12 Muestra de captura con el *Wireshark* al realizar un tracert a 192.168.3.2.

Ventana capturada por: Autor

Para ver la configuración completa de los dispositivos empleados en este escenario consultar el anexo IV.

2.2.2 Escenario del eNSP con una red Frame Relay

En la figura 2.13 se muestra un escenario de prueba del eNSP utilizado, en el cual se presenta una red frame relay constituida por tres enrutadores (AR0, AR1 y AR2) conectados cada uno por la interfaz Ethernet 0/0/0 a una máquina virtual (CLIENT0, CLIENT1 y CLIENT2) respectivamente.

Los enrutadores AR0 y AR1 se conectan por la interfaz Serial 0/0/0 al conmutador frame relay FRSW0 y el enrutador AR2 se conecta por la interfaz Serial 0/0/0 al conmutador frame relay FRSW1. FRSW0 y FRSW1 a su vez se conectan entre sí para cerrar la red.

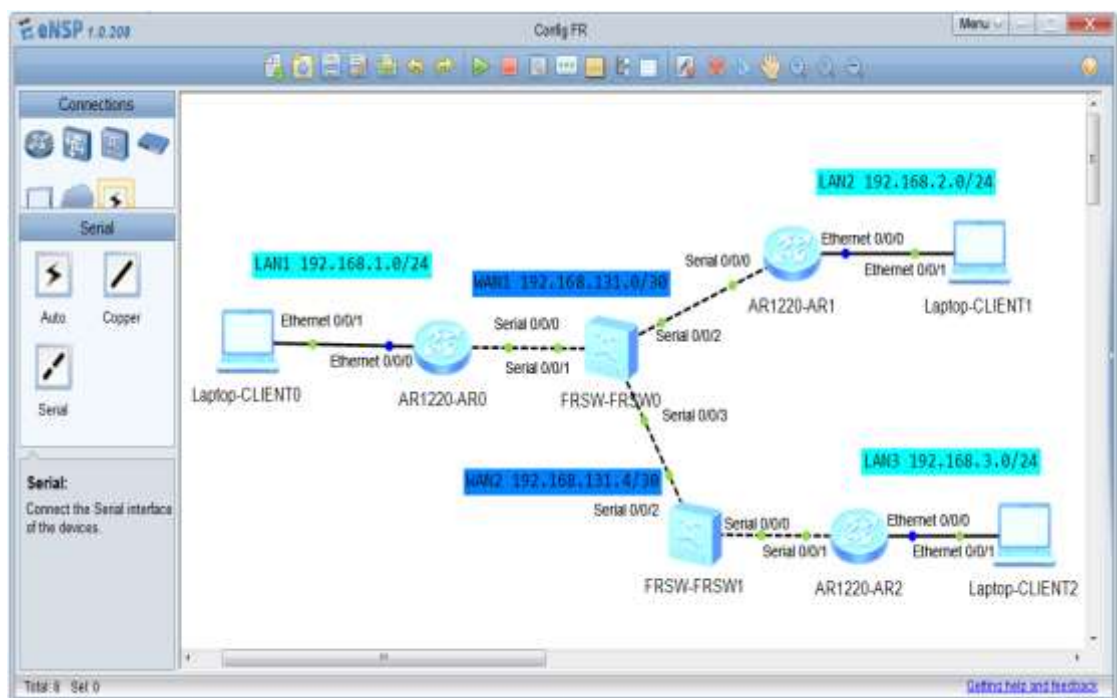


Figura 2. 13 Escenario del eNSP con una red Frame Relay.

Ventana capturada por: Autor

En este escenario del eNSP con una red frame relay se pueden realizar pruebas de conectividad realizando *ping* y *tracert* de una PC a otra PC y realizar

capturas de tráfico en sus enlaces. Así como poder analizar los protocolos que estén corriendo. En las figuras 2.14 y 2.15 se muestra una ventana de captura al realizar un *tracert* a la dirección 192.168.2.2 y un *ping* a la dirección 192.168.1.2 respectivamente, posible obtener con la ayuda del *Wireshark*.

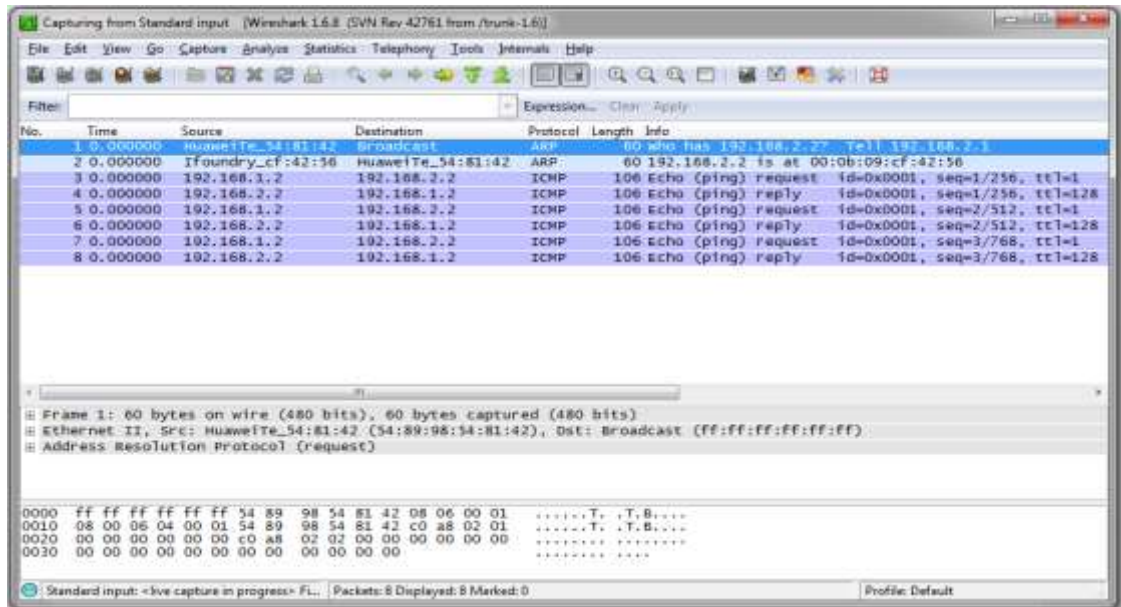


Figura 2. 14 Muestra de captura con el *Wireshark* al realizar un *tracert* a 192.168.2.2.

Ventana capturada por: Autor

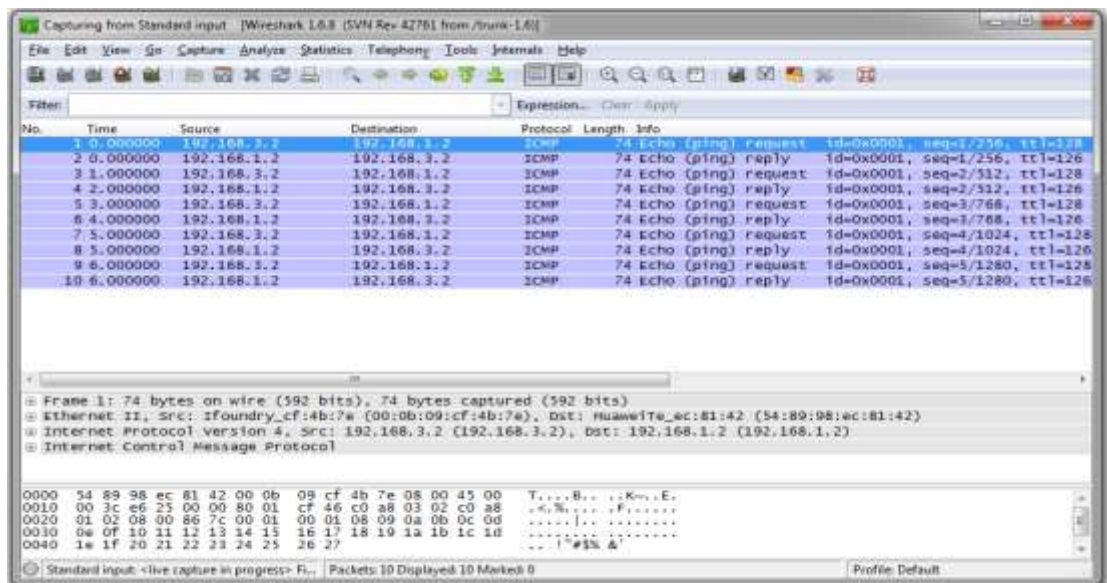


Figura 2. 15 Muestra de captura con el *Wireshark* al realizar un *ping* a 192.168.1.2.

Ventana capturada por: Autor

Para ver la configuración completa de los dispositivos empleados en este escenario consultar el anexo V.

2.2.3 Escenario del eNSP usando el protocolo *Spanning Tree*

En la figura 2.16 se muestra un escenario de prueba del eNSP utilizado, en el cual se presenta una red usando el protocolo *Spanning Tree* constituida por tres conmutadores Ethernet (LSW0, LSW1 y LSW2) interconectados entre sí. LSW0 además está conectado a una nube IP a través de la interfaz Ethernet 0/0/1.

Mientras que LSW1 y LSW2 se conectan por la interfaz Ethernet 0/0/3 a una máquina virtual (CLIENT0 y CLIENT1) respectivamente.

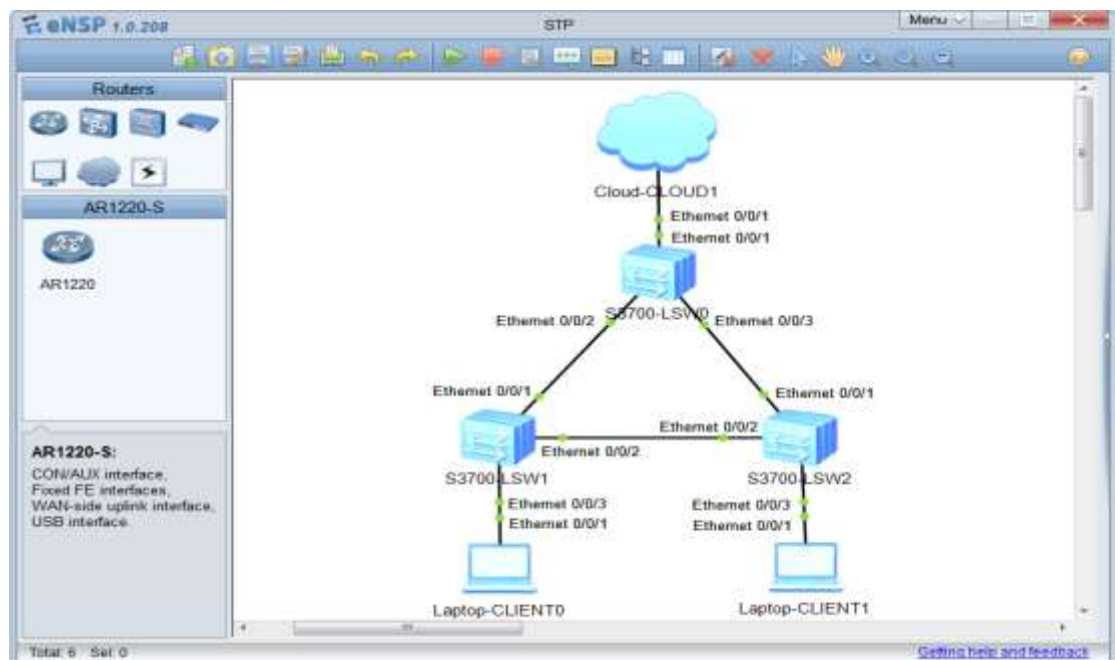


Figura 2. 16 Escenario del eNSP con una red *Spanning Tree*.

Ventana capturada por: Autor

En este escenario del eNSP con una red *Spanning Tree* se pueden realizar pruebas de conectividad realizando *ping* y *tracert* de una PC a otra PC, ver el

comportamiento de estos conmutadores y realizar capturas de tráfico en sus enlaces.

En las figuras 2.17 y 2.18 se muestran una ventana de captura de tráfico en el escenario antes mencionado, posible de obtener con el *Wireshark* al realizar un *ping* a la dirección 192.168.1.30 y un *tracert* a la dirección 192.168.1.20 respectivamente.

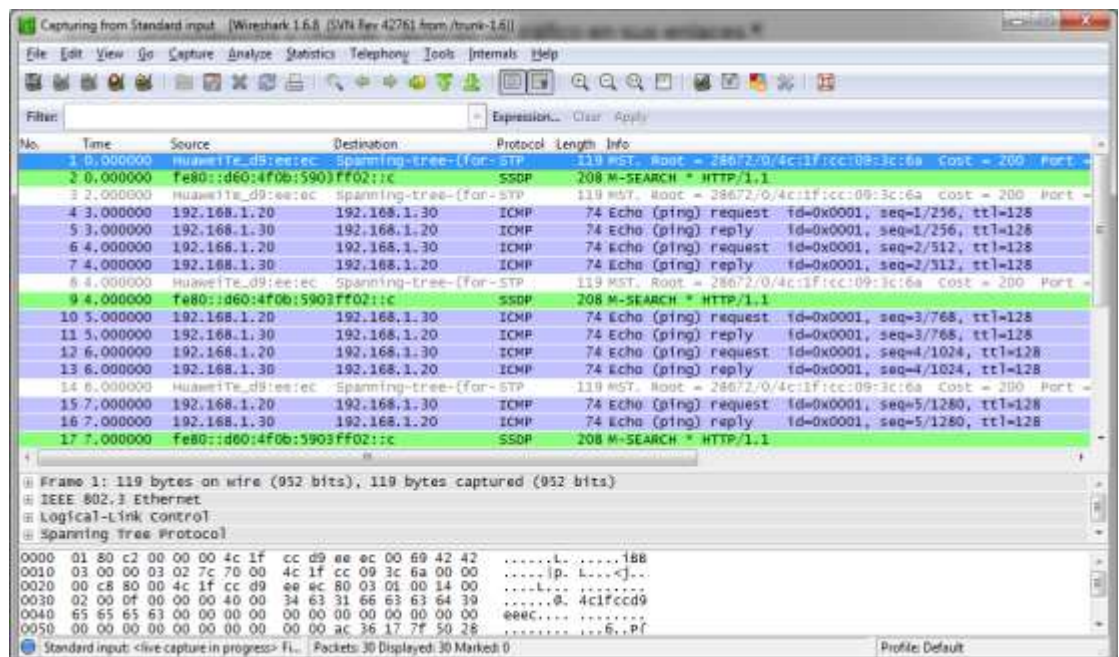


Figura 2. 17 Muestra de captura con el *Wireshark* al realizar un *ping* a 192.168.1.30.

Ventana capturada por: Autor

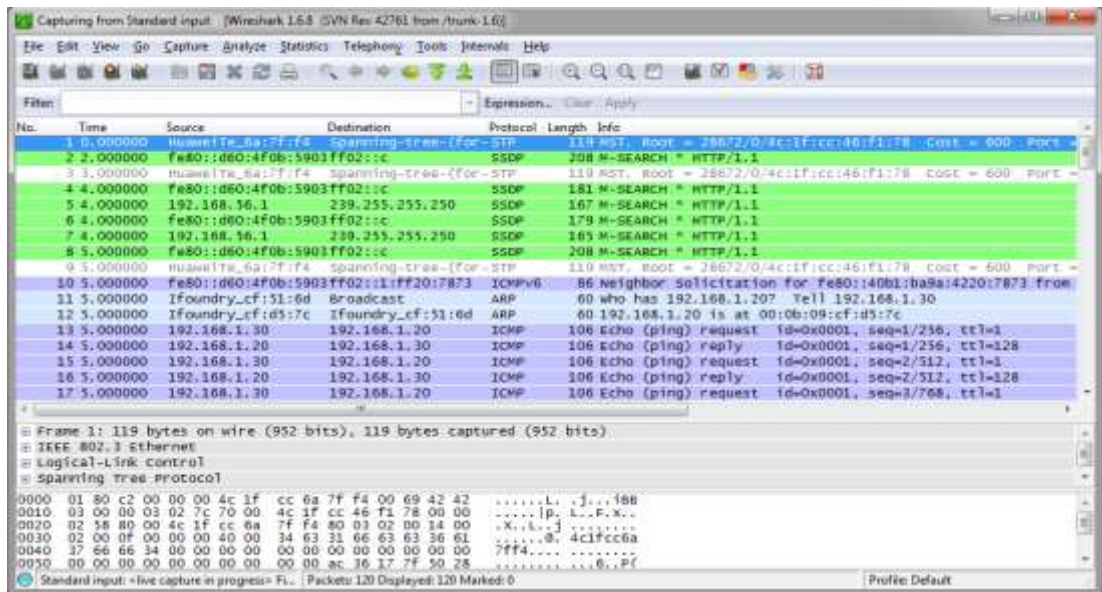


Figura 2. 18 Muestra de captura con el *Wireshark* al realizar un *tracert* a 192.168.1.20.

Ventana capturada por: Autor

Para ver la configuración completa de los dispositivos empleados en este escenario consultar el anexo VI.

2.2.4 Escenario del eNSP con una red IP/MPLS

En la figura 2.19 se muestra un escenario de prueba del eNSP con una red IP/MPLS. El mismo está constituido por seis enrutadores (Router_1, Router_2, Router_3, Router_4, Router_5 y Router_6), dos PC (CLIEN1 y CLIEN2) y dos conmutadores Ethernet (LSW1 y LSW2).

En este escenario se puede realizar un estudio más amplio de la operación de la arquitectura IP/MPLS con la ayuda del programa *Wireshark*, al detallar el comportamiento de cada paquete en la red y el funcionamiento de los protocolos que estén corriendo.

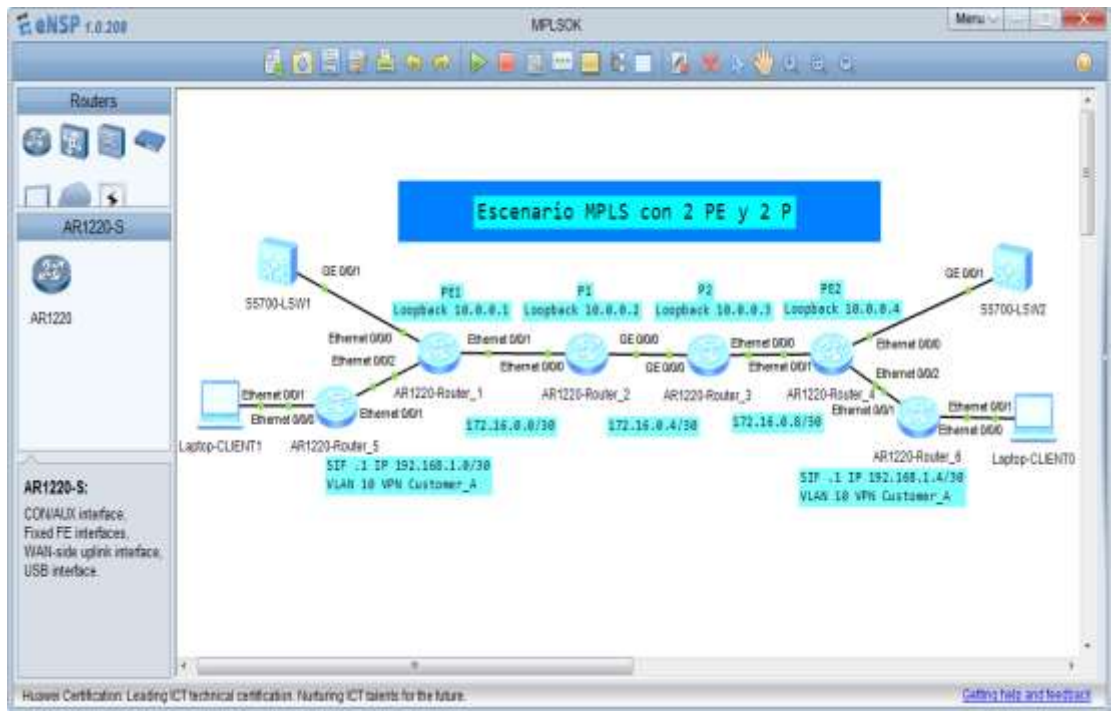


Figura 2. 19 Escenario del eNSP con una red IP/MPLS.

Ventana capturada por: Autor

En la figura 2.20 se muestra una ventana de captura de tráfico posible de obtener con la ayuda del *Wireshark* en una de las interfaces del enrutador P1 (Router_2) que pertenece al núcleo de la red IP/MPLS, a través de esta captura es posible se demuestra con mayor claridad cómo trabajan el protocolos IS-IS (*Intermedia System - Intermedia System*), entre otros aspectos.

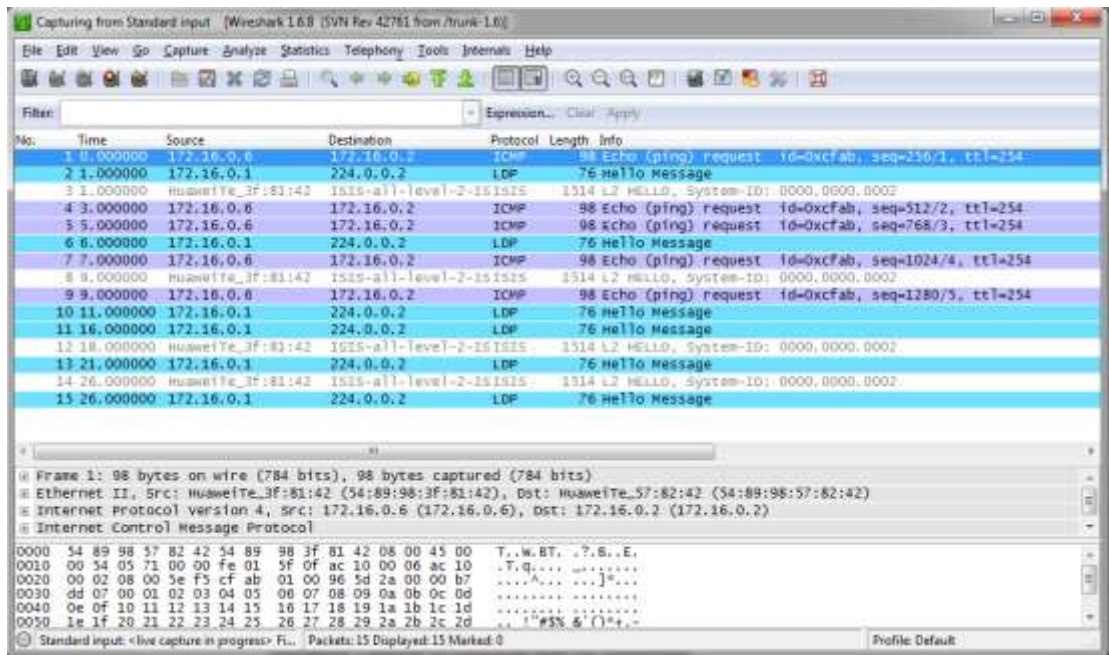


Figura 2. 20 Captura de tráfico con el *Wireshark* en el escenario IP/MPLS.

Ventana capturada por: Autor

Para ver la configuración completa de los dispositivos empleados en este escenario consultar el anexo VII.

VALORACIÓN ECONÓMICA

Para el desarrollo de este trabajo no fue necesaria la adquisición de nuevos dispositivos de redes que conlleven a nuevas inversiones, dado que los objetivos del proyecto tienen aspectos metodológicos y educativos los cuales resultaran en una mayor eficiencia y desempeño del personal.

La implementación de los laboratorios virtuales de redes que se presentan en este trabajo, permite el ahorro de recursos financieros, pues la implementación de maquetas con equipos de redes reales no es viable económicamente.

La tabla 1 muestra los precios aproximados de los dispositivos empleados.

Cantidades	Artículo	Precio Unitario (USD)	Costo total (USD)	Rubros	Según Apéndice 3
5	Router c3600	328,32	1641,6	Equipamiento	equipo/computación
4	Router c1700	93,86	375,44	Equipamiento	equipo/computación
3	Router c7200	547,24	1641,7	Equipamiento	equipo/computación
2	Router AT-AR024	84,45	168,9	Equipamiento	equipo/computación
5	Router AT-AR410-10	404	2020	Equipamiento	equipo/computación
4	Router AT-AR1220-S	607,25	2429	Equipamiento	equipo/computación
5	Conmutador AT-FS713FC/SC-10	201,16	1008	Equipamiento	equipo/computación
2	Conmutador AT-8024-10	356,14	712,28	Equipamiento	equipo/computación

Precio total (USD)
9996,94

Tabla 2 Valoración económica de los precios aproximados de los dispositivos empleados.

CONCLUSIONES

Después de haber realizado este proyecto y teniendo en cuenta los objetivos planteados, se arriba a las siguientes conclusiones:

- Debido a la alta similitud que presentan las redes simuladas en el GNS3 y en el eNSP, se minimiza la necesidad de emplear maquetas con dispositivos de redes reales en los cursos de transmisión de datos impartidos en las empresas de telecomunicaciones para una mejor comprensión del funcionamiento los dispositivos.
- Con la implementación de los laboratorios virtuales de redes que se presentan en este trabajo, se podría ahorrar una suma de \$ 9996,94 pues la implementación de maquetas con equipos de redes reales se elevaría a este precio.
- Con el programa GNS3 se pueden emular plataformas de *hardware* de varios modelos de enrutadores reales de CISCO, como las plataformas 1700, 2600, 2691, 3600, 3700 y 7200.
- Con el programa eNSP se pueden simular redes que empleen equipos del fabricante Huawei, permitiendo a los estudiantes una mejor comprensión de su funcionamiento.
- El GNS3 y el eNSP permitensu interacción con otros programas como el *Wireshark*, permitiendola captura de paquetes y el análisis de protocolos.
- De los dos simuladores empleados para la confección del presente trabajo el GNS3 es más complejo de utilizar, sin embargo, es más completo para su empleo en la rama de la enseñanza pues trabaja sobre imágenes CISCO reales y es mayor su similitud con dispositivos de redes reales.
- La integración en el GNS3 de varios programas de emulación, permitió virtualizar en una única PC múltiples escenarios de prueba, que unido a las

facilidades de inicio, detención, reinicio, carga de configuraciones, entre otras acciones sobre los dispositivos emulados, redundaron en un mejor aprovechamiento del tiempo en el desarrollo de las actividades planificadas.

RECOMENDACIONES

En este trabajo de investigación se ha observado que los avances en materia de las Tecnologías de la Información y las Comunicaciones son importantes para todos los sectores de la comunidad debido especialmente a las ventajas que éstas introducen en el proceso enseñanza-aprendizaje y que por lo tanto se considera pertinente su aplicación.

Programas como GNS3 y eNSP, permiten simular dispositivos de redes con un alto grado de similitud con los reales, siendo ambos programas libres, con entornos amistosos, flexibles e intuitivos y resultan fáciles de instalar y utilizar, por lo que se recomienda su utilización.

En el presente trabajo se diseñaron prácticas de laboratorio utilizando este simulador de redes de datos eNSP y el GNS3 como emulador de dispositivos, estas se recomiendan que sean utilizadas en los cursos de transmisión de datos.

Se recomienda utilizar los principales conceptos y características de los programas utilizados en este trabajo así como las simulaciones efectuadas en varios escenarios con diferentes tecnologías de redes, los cuales pueden ser empleadas en las prácticas de laboratorio.

BIBLIOGRAFÍA

- Álvarez, J., García, V., González, D., González, G., Rodríguez, D., Rubio, M., y otros. (2011). *Transmisión de datos por la red eléctrica*. Recuperado el 15 de Enero de 2014, de <http://www.victorgarcia.org/>:
<http://www.victorgarcia.org/files/PLC-v2.0RC.pdf>
- Barragan, A. (7 de mayo de 2012). *Topologías de red*. Recuperado el 30 de Julio de 2013, de [uhu.es](http://www.uhu.es):
<http://www.uhu.es/antonio.barragan/content/5topologias>
- Belmonte, P. (2008). *LA TECNOLOGÍA WIMAX*. Recuperado el Abril de 2011, de <http://www.pedrocores.com/wimax.pdf>
- Boney, J. (2005). *Cisco IOS in a Nutshell, 2nd Edition*. O'Reilly Media.
- bsi. (s.f.). *Seguridad de la información ISO/IEC 27001*. Recuperado el 15 de Mayo de 2013, de <http://www.bsigroup.es/certificacion-y-auditoria/Sistemas-de-gestion/estandares-esquemas/Seguridad-de-la-Informacion-ISOIEC27001>
- Bucker, C. (28 de Septiembre de 2012). *Seguridad Informática*. Recuperado el 4 de Enero de 2014, de calebbucker.blogspot.com:
<http://calebbucker.blogspot.com/2012/09/information-gathering-mediante-el-uso.html>
- Bustamante, R. (s.f.). *Seguridad en redes*. Universidad Autónoma del Estado de Hidalgo.
- Calero, N. (2009). *Análisis comparado de las diferentes alternativas tecnológicas de conectividad*. Recuperado el Mayo de 2013, de <http://www.itu.int/osg/spu/spunews/tecnologia/comparativa.html>
- Castaño, M. (2012). *Planificación y Administración de Redes*. Recuperado el 20 de Octubre de 2013, de Suarez de Figueroa A.S.I.R.: <http://www.suarezdefigueroa.es/manuel/PAR/index.php>
- Castro, L. (8 de Octubre de 2012). *Topologías de las redes*. Recuperado el 30 de Julio de 2013, de Prezi:
<http://prezi.com/ppgfyt22yelv/copy-of-topologias-de-las-redes/>
- Cisco. (s.f.). *Cisco 1700 Series Modular Access Routers*. Obtenido de http://www.cisco.com/en/US/docs/routers/access/1700/1711/hardware/quick/guide/171Xq_sp.html
- CiscoSystems. (s.f.). *Portable Product Sheets-Routing Performance*. Obtenido de http://www.optimumdata.com/shop/files/cisco/3600/3600_Response_Time_Report_Enhance.pdf

- CiscoSystemsInc. (s.f.). *Cisco 2600 Series Router Architecture, Cisco 2600 Series Multiservice Platforms*. Obtenido de http://www9.cisco.com/application/pdf/paws/23852/2600_architecture_23852.pdf
- Comer, D., & Stevens, D. (2000). *INTERCONECTIVIDAD DE REDES CON TCP/IP: DISEÑO E IMPLEMENTACION (TOMO 2)*. Mexico: PRENTICE HALL .
- Cristian, F. (2005). *Linux máxima seguridad*.
- Cristian, F. (2008). *Seguridad Informática, sus implicaciones e implementación*.
- Cuevas, R. (s.f.). *CONSULTORÍA INTEGRAL DE TIC'S*. Recuperado el 1 de Diciembre de 2013, de itescam.edu.mx: www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r85351.PDF
- Díaz, L. (2012). *Evaluación de la herramienta GNS3 con conectividad a enrutadores reales*. Lima: Pontificia Universidad Católica del Perú.
- Forouzan, B. (2007). *TRANSMISION DE DATOS Y REDES DE COMUNICACIONES (4ª ED.)*. MCGRAW-HILL.
- García, A. (2009). *Contribución al Desarrollo de Herramientas Estratégicas para el Diseño, Dimensionado y Evaluación de Redes de Telecomunicación de Banda Ancha*. Santander: Universidad de Cantabria.
- García, P., Díaz, J., & López, J. (2003). *TRANSMISION DE DATOS Y REDES DE COMPUTADORES*. PEARSON EDUCACION.
- Garzón, J. (s.f.). *Monitorización gráfica del tráfico de red y otros parámetros del sistema*. Obtenido de http://beta.redeslinux.com/manuales/Monitorizacion_redes/mrtg.pdf Enero 2013
- Hernández, R. (2008). *Virtualización de equipos de comunicaciones para la creación de un laboratorio en la modalidad de teleenseñanza*. Madrid: Universidad politécnica de Madrid.
- Hernández, R., Fernández, c., & Baptista, P. (2003). *Metodología de la Investigación (Tercera ed.)*. México D.F.: McGraw Hill.
- Herrera, E. (2003). *Tecnologías y redes de transmisión de datos*. Editorial Limusa.
- Huidrobo, J. (2010). *TELECOMUNICACIONES: TECNOLOGIAS, REDES Y SERVICIOS* . RA-MA.

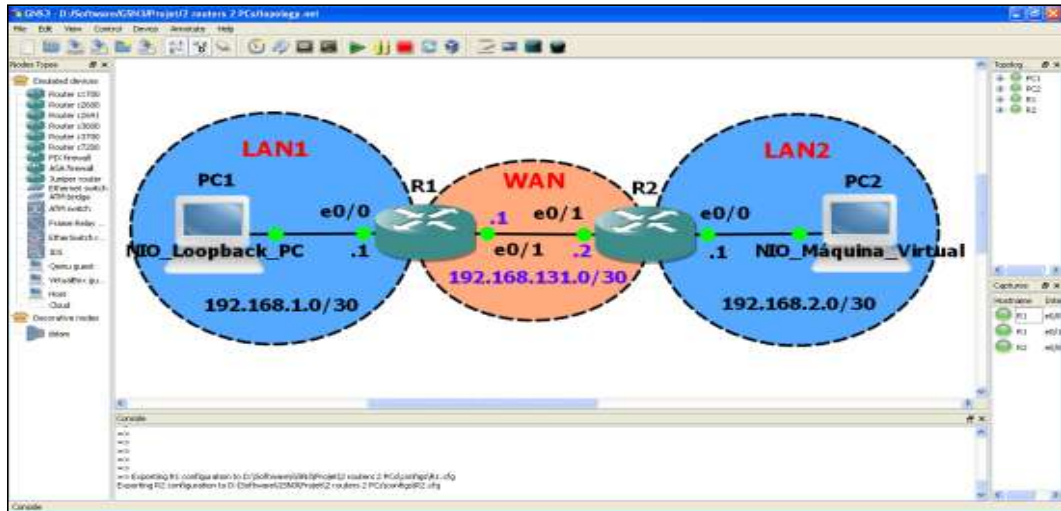
- Huidrobo, J., & Lozano, P. (2011). *NORMATIVA DE LAS INFRAESTRUCTURAS COMUNES DE TELECOMUNICACIONES: INFRAESTRUCTURAS DE ACCESO ULTRARRAPIDAS Y HOGAR DIGITAL 8REAL DECRETO 346/2011. NUEVO REGLAMENTO DE ICT* . CREACIONES COPYRIGHT.
- Lorandi, A., Hermida, G., Hernández, J., & Guevara, E. L. (2011). Los Laboratorios Virtuales y Laboratorios Remotos en la Enseñanza de la Ingeniería . *Revista Internacional de Educación en Ingeniería*, 24-30.
- Martín, E., & Angulo, I. (2003). *Microcontroladores PIC*. Copibook.
- Merino, B. (2011). *Análisis de tráfico con Wireshark*. España: Instituto Nacional de tecnologías de la Comunicación.
- Meyers, M. (2010). *REDES: ADMINISTRACION Y MANTENIMIENTO*. ANAYA MULTIMEDIA.
- Millan, R. (2007). *REDES DE DATOS Y CONVERGENCIA IP*. CREACIONES COPYRIGHT.
- Quisbert, M., & Calle, F. (2010). *Laboratorio Virtual para el Aprendizaje de Electrónica*. LA PAZ- BOLIVIA: UNIVERSIDAD MAYOR DE SAN ANDRES.
- Sánchez, A., & Hinojosa, G. (2009). *Análisis, diseño e implementación de una red LAN por medios guiados y no guiados en el Colegio Técnico Semi-presencial Intercultural Bilingüe Rumiloma*. Guaranda-Ecuador: Universidad Estatal de Bolívar.
- Scaniello, P., & Sosa, G. (2005). *Medidas a realizar en dispositivos de redes*. Recuperado el 15 de Mayo de 2013, de Curso de Evaluación de performance de Redes:
http://iie.fing.edu.uy/ense/assign/perfredes/trabajos/trabajos_2005/dispositivos/dispRedes.pdf
- Tomasi, W. (2003). *Sistemas de Comunicaciones Electrónicas*. Mexico: Prentice Hall.
- UIT. (s.f.). *Tendencia y evolución en el entorno de las telecomunicaciones*. Obtenido de <http://www.uit.int/tendencia> y evolución del entorno de las telecomunicaciones/
- UNAM. (s.f.). *Seguridad Informática*. Recuperado el 4 de Enero de 2014, de Universidad Nacional Autónoma de México:
<http://redyseguridad.fi-p.unam.mx/proyectos/seguridad/index.php>
- wndw. (2007). *Redes Inalámbricas en los Países en Desarrollo*. Recuperado el Mayo de 2013, de <http://wndw.net/>:
wndw.net/pdf/wndw2-es/wndw2-es-ebook.pdf

Zhenzhen, G., & Zhonglin, L. (2008). Summa Telecom: a preemptive strike. *Huawei Technologies Co. Ltd.*, p.13-14.

ANEXOS

Anexo I

Configuración de los enrutadores del escenario con dos enrutadores y dos PCs en el GNS3.



Enrutador R1:

```
!  
version 12.4  
service timestamps debug datetime msec  
service timestamps log datetime msec  
no service password-encryption  
!  
hostname R1  
!  
boot-start-marker  
boot-end-marker  
!  
no aaa new-model  
memory-size iomem 5  
!  
ip cef  
no ip domain lookup  
!  
interface Ethernet0/0
```

```
ip address 192.168.1.1 255.255.255.252
half-duplex
!
interface Ethernet0/1
ip address 192.168.131.1 255.255.255.252
half-duplex
!
interface Ethernet0/2
no ip address
shutdown
half-duplex
!
interface Ethernet0/3
no ip address
shutdown
half-duplex
!
no ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 192.168.131.2
!
control-plane
!
line con 0
exec-timeout 0 0
logging synchronous
line aux 0
line vty 0 4
!
end
```

Enrutador R2:

```
!
version 12.4
```

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R2
!
boot-start-marker
boot-end-marker
!
no aaa new-model
memory-size iomem 5
!
ip cef
no ip domain lookup
!
interface Ethernet0/0
 ip address 192.168.2.1 255.255.255.252
 half-duplex
!
interface Ethernet0/1
 ip address 192.168.131.2 255.255.255.252
 half-duplex
!
interface Ethernet0/2
 no ip address
 shutdown
 half-duplex
!
interface Ethernet0/3
 no ip address
 shutdown
 half-duplex
!
no ip http server
```



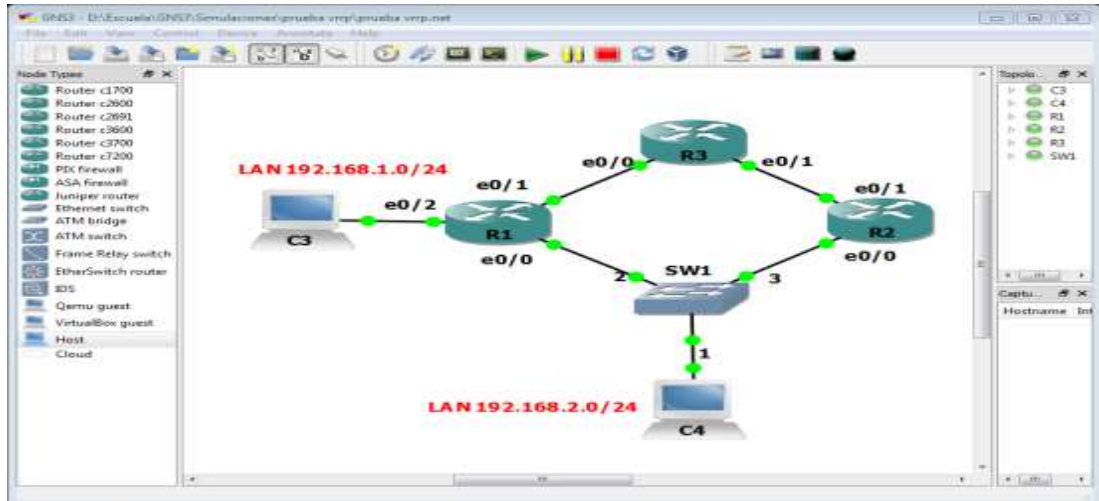
```

no ip http secure-server
ip route 0.0.0.0 0.0.0.0 192.168.131.1
!
control-plane
!
line con 0
exec-timeout 0 0
logging synchronous
line aux 0
line vty 0 4
login
!
end

```

Anexo II

Configuración de los enrutadores del escenario con una red VRRP en el GNS3.



Enrutador R1:

```

!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption

```

```
!  
hostname R1  
!  
boot-start-marker  
boot-end-marker  
!  
no aaa new-model  
memory-size iomem 5  
!  
ip cef  
!  
interface Ethernet0/0  
ip address 191.168.1.1 255.255.255.0  
half-duplex  
vrrp 1 description Grupo1  
vrrp 1 ip 192.168.1.10  
vrrp 1 timers learn  
vrrp 1 priority 150  
!  
interface Ethernet0/1  
ip address 191.168.3.1 255.255.255.0  
half-duplex  
!  
interface Ethernet0/2  
ip address 192.168.1.1 255.255.255.0  
half-duplex  
!  
interface Ethernet0/3  
no ip address  
shutdown  
half-duplex  
!  
router ospf 1  
log-adjacency-changes
```

```
network 192.168.1.0 0.0.0.255 area 0
network 192.168.2.0 0.0.0.3 area 0
network 192.168.3.0 0.0.0.3 area 0
network 192.168.4.0 0.0.0.0 area 0
!
ip http server
no ip http secure-server

!
access-list 99 permit 192.168.1.3
access-list 99 deny any log
snmp-server community public RO 99
!
control-plane
!
line con 0
line aux 0
line vty 0 4
login
!
end
```

Enrutador R2:

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router2
!
boot-start-marker
boot-end-marker
!
```

```
no aaa new-model
memory-size iomem 5
!
ip cef
!
interface Ethernet0/0
ip address 192.168.1.2 255.255.255.0
half-duplex
vrrp 1 description Grupo1
vrrp 1 ip 192.168.1.10
vrrp 1 timers learn
!
interface Ethernet0/1
ip address 192.168.3.1 255.255.255.252
half-duplex
!
interface Ethernet0/2
no ip address
shutdown
half-duplex
!
interface Ethernet0/3
no ip address
shutdown
half-duplex
!
router ospf 1
router-id 192.168.3.1
log-adjacency-changes
network 192.168.1.0 0.0.0.255 area 0
network 192.168.2.0 0.0.0.3 area 0
network 192.168.3.0 0.0.0.3 area 0
network 192.168.4.0 0.0.0.0 area 0
!
```

```
ip http server
no ip http secure-server
!
control-plane
!
line con 0
line aux 0
line vty 0 4
login
!
end
```

Enrutador R3:

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R3
!
boot-start-marker
boot-end-marker
!
no aaa new-model
memory-size iomem 5
!
ip cef
!
interface Loopback0
 ip address 192.168.4.1 255.255.255.255
 ip ospf 1 area 0
!
interface Ethernet0/0
```

```
ip address 192.168.2.2 255.255.255.252
half-duplex
!
interface Ethernet0/1
ip address 192.168.3.2 255.255.255.252
half-duplex
!
interface Ethernet0/2
ip address 192.168.10.1 255.255.255.0
half-duplex
!
interface Ethernet0/3
ip address 192.168.20.1 255.255.255.0
half-duplex
!
router ospf 1
router-id 192.168.2.2
log-adjacency-changes
network 192.168.1.0 0.0.0.255 area 0
network 192.168.2.0 0.0.0.3 area 0
network 192.168.3.0 0.0.0.3 area 0
network 192.168.4.0 0.0.0.0 area 0
!
ip http server
no ip http secure-server
!
control-plane
!
line con 0
line aux 0
line vty 0 4
login
!
end
```



```
!  
interface Loopback11  
  ip address 192.168.11.11 255.255.255.0  
!  
interface Loopback101  
  ip address 192.168.101.101 255.255.255.0  
!  
interface Ethernet0/0  
  description Conexion_PE1  
  ip address 192.168.1.2 255.255.255.252  
  half-duplex  
!  
interface Ethernet0/1  
  description Conexion_LAN1  
  ip address 192.168.10.1 255.255.255.0  
  half-duplex  
!  
interface Ethernet0/2  
  no ip address  
  shutdown  
  half-duplex  
!  
interface Ethernet0/3  
  no ip address  
  shutdown  
  half-duplex  
!  
no ip http server  
no ip http secure-server  
ip route 0.0.0.0 0.0.0.0 192.168.1.1  
!  
control-plane  
!  
line con 0
```



```
exec-timeout 0 0
logging synchronous
line aux 0
line vty 0 4
login
!
end
```

Enrutador R2:

```
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname PE1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
memory-size iomem 5
!
ip cef
no ip domain lookup
!
ip vrf Customer_A
rd 11:111
route-target export 11:111
route-target import 11:111
!
ip vrf Customer_B
rd 22:222
route-target export 22:222
```

```
route-target import 22:222
!
mpls label protocol ldp
!
interface Loopback0
description is-is/ bgp/ mpls ldp router-id
ip address 10.0.0.11 255.255.255.255
ip router isis
!
interface Ethernet0/0
description Conexion_CE1
ip vrf forwarding Customer_A
ip address 192.168.1.1 255.255.255.252
half-duplex
!
interface Ethernet0/1
description Conexion_CE3
ip vrf forwarding Customer_B
ip address 192.168.1.5 255.255.255.252
half-duplex
!
interface Ethernet0/2
description Conexion_P
ip address 172.16.0.6 255.255.255.252
ip router isis
half-duplex
mpls ip
!
interface Ethernet0/3
no ip address
shutdown
half-duplex
!
router isis
```

```
net 01.0000.0000.0000.0000.0010.0000.0000.0011.00
is-type level-2-only
!
router bgp 65001
  bgp router-id 10.0.0.11
  no bgp default ipv4-unicast
  bgp log-neighbor-changes
  neighbor 10.0.0.22 remote-as 65001
  neighbor 10.0.0.22 update-source Loopback0
!
  address-family ipv4
    neighbor 10.0.0.22 activate
    no auto-summary
    no synchronization
  exit-address-family
!
  address-family vpnv4
    neighbor 10.0.0.22 activate
    neighbor 10.0.0.22 send-community extended
  exit-address-family
!
  address-family ipv4 vrf Customer_B
    redistribute connected
    redistribute static
    no synchronization
  exit-address-family
!
  address-family ipv4 vrf Customer_A
    redistribute connected
    redistribute static
    no synchronization
  exit-address-family
!
no ip http server
```

```
no ip http secure-server
ip route vrf Customer_A 192.168.10.0 255.255.255.0 192.168.1.2
ip route vrf Customer_A 192.168.11.0 255.255.255.0 192.168.1.2
ip route vrf Customer_A 192.168.101.0 255.255.255.0 192.168.1.2
ip route vrf Customer_B 192.168.30.0 255.255.255.0 192.168.1.6
!
control-plane
!
line con 0
  exec-timeout 0 0
  logging synchronous
line aux 0
line vty 0 4
  login
!
end
```

Enrutador R3:

```
!
! Last configuration change at 20:43:12 UTC Mon Mar 19 2012
!
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
!
hostname P
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip source-route
!
no ip domain lookup
```

```
ip cef
no ipv6 cef
!
mpls label protocol ldp
multilink bundle-name authenticated
!
interface Loopback0
description is-is/ bgp/ mpls ldp router-id
ip address 10.0.0.1 255.255.255.255
ip router isis
!
interface FastEthernet0/0
no ip address
shutdown
duplex half
!
interface Ethernet1/0
description Conexion_PE1
ip address 172.16.0.5 255.255.255.252
ip router isis
duplex half
mpls ip
!
interface Ethernet1/1
description Conexion_PE2
ip address 172.16.0.1 255.255.255.252
ip router isis
duplex half
mpls ip
!
interface Ethernet1/2
no ip address
shutdown
duplex half
```

```
!  
interface Ethernet1/3  
  no ip address  
  shutdown  
  duplex half  
!  
interface Ethernet1/4  
  no ip address  
  shutdown  
  duplex half  
!  
interface Ethernet1/5  
  no ip address  
  shutdown  
  duplex half  
!  
interface Ethernet1/6  
  no ip address  
  shutdown  
  duplex half  
!  
interface Ethernet1/7  
  no ip address  
  shutdown  
  duplex half  
!  
router isis  
  net 01.0000.0000.0000.0000.0010.0000.0000.0001.00  
  is-type level-2-only  
!  
no ip http server  
no ip http secure-server  
!  
mpls ldp router-id Loopback0
```

```
!  
control-plane  
!  
line con 0  
  exec-timeout 0 0  
  logging synchronous  
  stopbits 1  
line aux 0  
  stopbits 1  
line vty 0 4  
  login  
!  
end
```

Enrutador R4:

```
!  
version 12.4  
service timestamps debug datetime msec  
service timestamps log datetime msec  
no service password-encryption  
!  
hostname PE2  
!  
boot-start-marker  
boot-end-marker  
!  
no aaa new-model  
memory-size iomem 5  
!  
ip cef  
no ip domain lookup  
!  
ip vrf Customer_A  
  rd 11:111
```

```
route-target export 11:111
route-target import 11:111
!
ip vrf Customer_B
rd 22:222
route-target export 22:222
route-target import 22:222
!
mpls label protocol ldp
!
interface Loopback0
description is-is/ bgp/ mpls ldp router-id
ip address 10.0.0.22 255.255.255.255
ip router isis
!
interface Ethernet0/0
description Conexion_CE2
ip vrf forwarding Customer_A
ip address 192.168.2.1 255.255.255.252
half-duplex
!
interface Ethernet0/1
description Conexion_CE4
ip vrf forwarding Customer_B
ip address 192.168.2.5 255.255.255.252
half-duplex
!
interface Ethernet0/2
description Conexion_P
ip address 172.16.0.2 255.255.255.252
ip router isis
half-duplex
mpls ip
!
```



```

interface Ethernet0/3
  no ip address
  shutdown
  half-duplex
  !
router isis
  net 01.0000.0000.0000.0000.0010.0000.0000.0022.00
  is-type level-2-only
  !
router bgp 65001
  bgp router-id 10.0.0.22
  no bgp default ipv4-unicast
  bgp log-neighbor-changes
  neighbor 10.0.0.11 remote-as 65001
  neighbor 10.0.0.11 update-source Loopback0
  !
  address-family ipv4
    redistribute connected
    redistribute static
    neighbor 10.0.0.11 activate
    no auto-summary
    no synchronization
  exit-address-family
  !
  address-family vpnv4
    neighbor 10.0.0.11 activate
    neighbor 10.0.0.11 send-community extended
  exit-address-family
  !
  address-family ipv4 vrf Customer_B
    redistribute connected
    redistribute static
    no synchronization
  exit-address-family

```

```
!  
address-family ipv4 vrf Customer_A  
  redistribute connected  
  redistribute static  
  no synchronization  
exit-address-family  
!  
no ip http server  
no ip http secure-server  
ip route vrf Customer_A 192.168.20.0 255.255.255.0 192.168.2.2  
ip route vrf Customer_A 192.168.22.0 255.255.255.0 192.168.2.2  
ip route vrf Customer_A 192.168.102.0 255.255.255.0 192.168.2.2  
ip route vrf Customer_B 192.168.40.0 255.255.255.0 192.168.2.6  
!  
control-plane  
!  
line con 0  
  exec-timeout 0 0  
  logging synchronous  
line aux 0  
line vty 0 4  
  login  
!  
end
```

Enrutador R5:

```
!  
version 12.4  
service timestamps debug datetime msec  
service timestamps log datetime msec  
no service password-encryption  
!  
hostname CE2  
!
```

```
boot-start-marker
boot-end-marker
!
no aaa new-model
memory-size iomem 5
!
ip cef
no ip domain lookup
!
interface Loopback22
 ip address 192.168.22.22 255.255.255.0
!
interface Loopback102
 ip address 192.168.102.102 255.255.255.0
!
interface Ethernet0/0
 description Conexion_PE2
 ip address 192.168.2.2 255.255.255.252
 half-duplex
!
interface Ethernet0/1
 description Conexion_LAN2
 ip address 192.168.20.1 255.255.255.0
 half-duplex
!
interface Ethernet0/2
 no ip address
 shutdown
 half-duplex
!
interface Ethernet0/3
 no ip address
 shutdown
 half-duplex
```

```
!  
no ip http server  
no ip http secure-server  
ip route 0.0.0.0 0.0.0.0 192.168.2.1  
!  
control-plane  
!  
line con 0  
  exec-timeout 0 0  
  logging synchronous  
line aux 0  
line vty 0 4  
  login  
!  
end
```

Enrutador R6:

```
!  
version 12.4  
service timestamps debug datetime msec  
service timestamps log datetime msec  
no service password-encryption  
!  
hostname CE3  
!  
boot-start-marker  
boot-end-marker  
!  
no aaa new-model  
memory-size iomem 5  
!  
ip cef  
no ip domain lookup  
!
```

```
interface Ethernet0/0
  description Conexion_PE1
  ip address 192.168.1.6 255.255.255.252
  half-duplex
  !
interface Ethernet0/1
  description Conexion_LAN1
  ip address 192.168.30.1 255.255.255.0
  half-duplex
  !
interface Ethernet0/2
  no ip address
  shutdown
  half-duplex
  !
interface Ethernet0/3
  no ip address
  shutdown
  half-duplex
  !
no ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 192.168.1.5
  !
control-plane
  !
line con 0
  exec-timeout 0 0
  logging synchronous
line aux 0
line vty 0 4
  login
  !
end
```

Enrutador R7:

```
!  
version 12.4  
service timestamps debug datetime msec  
service timestamps log datetime msec  
no service password-encryption  
!  
hostname CE4  
!  
boot-start-marker  
boot-end-marker  
!  
no aaa new-model  
memory-size iomem 5  
!  
ip cef  
no ip domain lookup  
!  
interface Ethernet0/0  
description Conexion_PE2  
ip address 192.168.2.6 255.255.255.252  
half-duplex  
!  
interface Ethernet0/1  
description Conexion_LAN2  
ip address 192.168.40.1 255.255.255.0  
half-duplex  
!  
interface Ethernet0/2  
no ip address  
shutdown  
half-duplex  
!
```

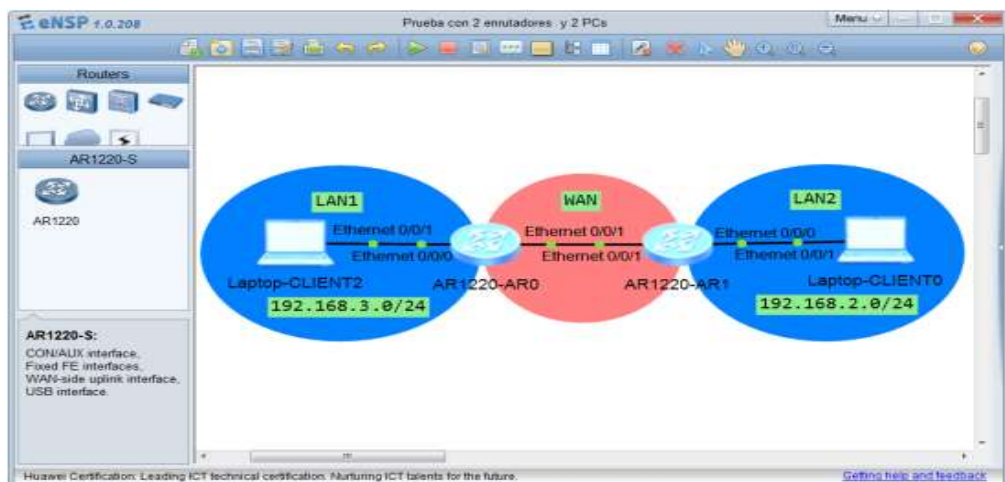
```

interface Ethernet0/3
no ip address
shutdown
half-duplex
!
no ip http server
no ip http secure-server
ip route 0.0.0.0 0.0.0.0 192.168.2.5
!
control-plane
!
line con 0
exec-timeout 0 0
logging synchronous
line aux 0
line vty 0 4
login
!
end

```

Anexo IV

Configuración de los enrutadores del escenario con dos enrutadores y dos PCs en el eNSP



Enrutador R1:

```
#
sysname R1
#
undo http server enable
#
nd user-bind detect retransmit 0 interval 0
#
aaa
authentication-scheme default
#
authorization-scheme default
#
accounting-scheme default
#
domain default
#
interface Ethernet0/0/0
ip address 192.168.1.1 255.255.255.0
#
interface Ethernet0/0/1
ip address 192.168.131.1 255.255.255.252
#
interface NULL0
#
ip route-static 0.0.0.0 0.0.0.0 192.168.131.2
#
snmp-agent
snmp-agent local-engineid 800007DB0354899831631C
snmp-agent community read public
snmp-agent sys-info version all
snmp-agent trap enable
#
user-interface con 0
user-interface vty 0 4
```



```
user-interface vty 16 20
```

```
#
```

```
return
```

Enrutador R2:

```
#
```

```
sysname R2
```

```
#
```

```
undo http server enable
```

```
#
```

```
nd user-bind detect retransmit 0 interval 0
```

```
#
```

```
aaa
```

```
authentication-scheme default
```

```
#
```

```
authorization-scheme default
```

```
#
```

```
accounting-scheme default
```

```
#
```

```
domain default
```

```
#
```

```
interface Ethernet0/0/0
```

```
ip address 192.168.2.1 255.255.255.0
```

```
#
```

```
interface Ethernet0/0/1
```

```
ip address 192.168.131.2 255.255.255.252
```

```
#
```

```
interface NULL0
```

```
#
```

```
ip route-static 0.0.0.0 0.0.0.0 192.168.131.1
```

```
#
```

```
user-interface con 0
```

```
user-interface vty 0 4
```

```
user-interface vty 16 20
```



```
domain default
#

interface Ethernet0/0/0
ip address 192.168.1.1 255.255.255.0
#
interface Ethernet0/0/1
#
interface Serial0/0/0
aaa
authentication-scheme default
#
authorization-scheme default
#
accounting-scheme default
#
domain default
#
interface Ethernet0/0/0
ip address 192.168.2.1 255.255.255.0
#
interface Ethernet0/0/1
link-protocol fr
#
interface Serial0/0/0.1
fr dlc1 16
ip address 192.168.131.1 255.255.255.252
#
interface Serial0/0/0.2
fr dlc1 17
ip address 192.168.131.5 255.255.255.252
#
interface Serial0/0/1
```

```
link-protocol ppp
#
interface NULL0
#
ip route-static 192.168.2.0 255.255.255.0 192.168.131.2
ip route-static 192.168.3.0 255.255.255.0 192.168.131.6
#
user-interface con 0
user-interface vty 0 4
user-interface vty 16 20
#
return
```

Enrutador R2:

```
#
sysname Router2
#
undo http server enable
#
nd user-bind detect retransmit 0 interval 0
#
#
interface Serial0/0/0
link-protocol fr
#
interface Serial0/0/0.1
fr dlc1 16
ip address 192.168.131.2 255.255.255.252
#
interface Serial0/0/1
link-protocol ppp
#
interface NULL0
#
```

```
ip route-static 0.0.0.0 0.0.0.0 192.168.131.1
#
user-interface con 0
user-interface vty 0 4
user-interface vty 16 20
#
return
```

Enrutador R3:

```
#
sysname Router3
#
undo http server enable
#
nd user-bind detect retransmit 0 interval 0
#
aaa
authentication-scheme default
#
authorization-scheme default
#
accounting-scheme default
#
domain default
#
interface Ethernet0/0/0
ip address 192.168.3.1 255.255.255.0
#
interface Ethernet0/0/1
#
interface Serial0/0/0
link-protocol fr
#
interface Serial0/0/0.1
```

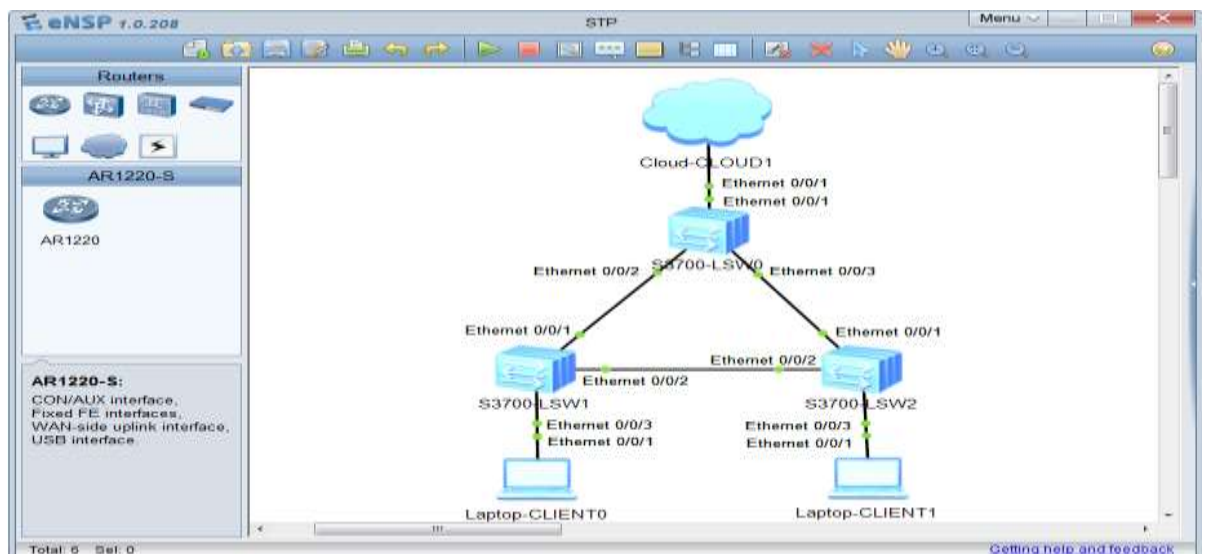
```

fr dlc1 16
ip address 192.168.131.5 255.255.255.252
#
interface Serial0/0/1
link-protocol ppp
#
interface NULL0
#
ip route-static 0.0.0.0 0.0.0.0 192.168.131.5
#
user-interface con 0
user-interface vty 0 4
user-interface vty 16 20
#
return

```

Anexo VI

Configuración de los conmutadores del escenario con una red Spanning Tree en el eNSP



Conmutador LSW0

```
#
sysname LSW0
#
stp instance 0 priority 28672
#
cluster enable
ntdp enable
ndp enable
#
drop illegal-mac alarm
#
diffserv domain default
#
drop-profile default
#
aaa
authentication-scheme default
authorization-scheme default
accounting-scheme default
domain default
domain default_admin
local-user admin password simple admin
local-user admin service-type http
#
interface Vlanif1
#
interface MEth0/0/1
#
interface Ethernet0/0/1
#
interface Ethernet0/0/2
#
interface Ethernet0/0/3
#
```

```
interface Ethernet0/0/4
#
interface Ethernet0/0/5
#
interface Ethernet0/0/6
#
interface Ethernet0/0/7
#
interface Ethernet0/0/8
#
interface Ethernet0/0/9
#
interface Ethernet0/0/10
#
interface Ethernet0/0/11
#
interface Ethernet0/0/12
#
interface Ethernet0/0/13
#
interface Ethernet0/0/14
#
interface Ethernet0/0/15
#
interface Ethernet0/0/16
#
interface Ethernet0/0/17
#
interface Ethernet0/0/18
#
interface Ethernet0/0/19
#
interface Ethernet0/0/20
#
```



```
interface Ethernet0/0/21
#
interface Ethernet0/0/22
#
interface GigabitEthernet0/0/1
#
interface GigabitEthernet0/0/2
#
interface NULL0
#
user-interface con 0
user-interface vty 0 4
#
return
```

Conmutador LSW1

```
#
sysname LSW1
#
cluster enable
ntdp enable
ndp enable
#
drop illegal-mac alarm
#
diffserv domain default
#
drop-profile default
#
aaa
authentication-scheme default
authorization-scheme default
accounting-scheme default
domain default
```

```
domain default_admin
local-user admin password simple admin
local-user admin service-type http
#
interface Vlanif1
#
interface MEth0/0/1
#
interface Ethernet0/0/1
  stp instance 0 cost 200
#
interface Ethernet0/0/2
  stp instance 0 cost 400
#
interface Ethernet0/0/3
#
interface Ethernet0/0/4
#
interface Ethernet0/0/5
#
interface Ethernet0/0/6
#
interface Ethernet0/0/7
#
interface Ethernet0/0/8
#
interface Ethernet0/0/9
#
interface Ethernet0/0/10
#
interface Ethernet0/0/11
#
interface Ethernet0/0/12
#
```

```
interface Ethernet0/0/13
#
interface Ethernet0/0/14
#
interface Ethernet0/0/15
#
interface Ethernet0/0/16
#
interface Ethernet0/0/17
#
interface Ethernet0/0/18
#
interface Ethernet0/0/19
#
interface Ethernet0/0/20
#
interface Ethernet0/0/21
#
interface Ethernet0/0/22
#
interface GigabitEthernet0/0/1
#
interface GigabitEthernet0/0/2
#
interface NULL0
#
user-interface con 0
user-interface vty 0 4
#
return
```

Conmutador LSW2

```
#
sysname LSW2
```

```
#
cluster enable
ntdp enable
ndp enable
#
drop illegal-mac alarm
#
diffserv domain default
#
drop-profile default
#
aaa
authentication-scheme default
authorization-scheme default
accounting-scheme default
domain default
domain default_admin
local-user admin password simple admin

local-user admin service-type http
#
interface Vlanif1
#
interface MEth0/0/1
#
interface Ethernet0/0/1
stp instance 0 cost 200
#
interface Ethernet0/0/2
stp instance 0 cost 400
#
interface Ethernet0/0/3
#
interface Ethernet0/0/4
```

```
#  
interface Ethernet0/0/5  
#  
interface Ethernet0/0/6  
#  
interface Ethernet0/0/7  
#  
interface Ethernet0/0/8  
#  
interface Ethernet0/0/9  
#  
interface Ethernet0/0/10  
#  
interface Ethernet0/0/11  
#  
interface Ethernet0/0/12  
#  
interface Ethernet0/0/13  
#  
interface Ethernet0/0/14  
#  
interface Ethernet0/0/15  
#  
interface Ethernet0/0/16  
#  
interface Ethernet0/0/17  
#  
interface Ethernet0/0/18  
#  
interface Ethernet0/0/19  
#  
interface Ethernet0/0/20  
#  
interface Ethernet0/0/21
```

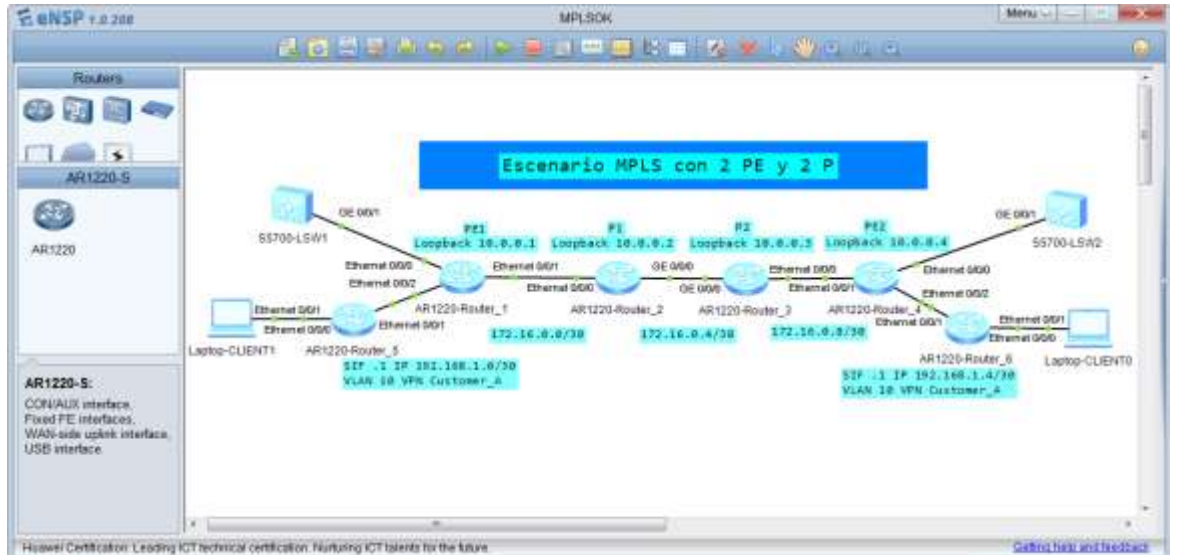
```

#
interface Ethernet0/0/22
#
interface GigabitEthernet0/0/1
#
interface GigabitEthernet0/0/2
#
interface NULL0
#
user-interface con 0
user-interface vty 0 4
#
return

```

Anexo VII

Configuración de los enrutadores en el escenario del eNSP con una red IP/MPLS



Enrutador Router_1

```

#
sysname Router1
#
undo http server enable

```

```
#
nd user-bind detect retransmit 0 interval 0
#
ip vpn-instance Customer_A
  ipv4-family
    route-distinguisher 11:111
    vpn-target 11:111 export-extcommunity
    vpn-target 11:111 import-extcommunity
#
ip vpn-instance Customer_B
  ipv4-family
    route-distinguisher 22:222
    vpn-target 22:222 export-extcommunity
    vpn-target 22:222 import-extcommunity
#
mpls lsr-id 10.0.0.1
mpls
#
mpls ldp
#
aaa
  authentication-scheme default
#
  authorization-scheme default
#
  accounting-scheme default
#
  domain default
#
isis 1
  is-level level-2
  network-entity 01.0000.0000.0000.0000.0010.0000.0000.0001.00
  import-route direct
#
```

```

interface Ethernet0/0/0
  description Conexion_LSW1
#
interface Ethernet0/0/0.1
  vlan-type dot1q 10
  ip binding vpn-instance Customer_A
  ip address 192.168.1.1 255.255.255.252
#
interface Ethernet0/0/0.2
  vlan-type dot1q 11
  ip binding vpn-instance Customer_A
  ip address 192.168.10.1 255.255.255.0
#
interface Ethernet0/0/1
  ip address 172.16.0.2 255.255.255.252
  isis enable 1
  isis circuit-level level-2
  mpls
  mpls ldp
#
interface Ethernet0/0/2
  ip binding vpn-instance Customer_B
  ip address 192.168.131.1 255.255.255.252
#
interface NULL0
#
interface LoopBack0
  description is-is/ bgp/ mpls ldp router-id
  ip address 10.0.0.1 255.255.255.255
  isis enable 1
#
bgp 65001
  group RR internal
  peer RR connect-interface LoopBack0

```



```
peer 10.0.0.4 as-number 65001
peer 10.0.0.4 group RR
#
ipv4-family unicast
undo synchronization
import-route direct
import-route static
peer RR enable
peer 10.0.0.4 enable
peer 10.0.0.4 group RR
#
ipv4-family vpnv4
policy vpn-target
peer RR enable
peer 10.0.0.4 enable
peer 10.0.0.4 group RR
#
ipv4-family vpn-instance Customer_A
import-route direct
import-route static
#
ipv4-family vpn-instance Customer_B
import-route direct
import-route static
#
ip route-static vpn-instance Customer_B 192.168.2.0 255.255.255.0
192.168.131.2
#
user-interface con 0
user-interface vty 0 4
user-interface vty 16 20
#
return
```

Enrutador Router_2

```
#
sysname Router2
#
arp learning strict
#
undo http server enable
#
nd user-bind detect retransmit 0 interval 0
#
mpls lsr-id 10.0.0.2
mpls
#
mpls ldp
#
aaa
 authentication-scheme default
#
 authorization-scheme default
#
 accounting-scheme default
#
 domain default
#
isis 1
 is-level level-2
 network-entity 01.0000.0000.0000.0000.0010.0000.0000.0002.00
import-route direct
#
interface Ethernet0/0/0
 description Conexion_PE1
 ip address 172.16.0.1 255.255.255.252
 isis enable 1
 isis circuit-level level-2
```

```

mpls
mpls ldp
#
interface Ethernet0/0/1
#
interface GigabitEthernet0/0/0
description Conexion_P2
ip address 172.16.0.5 255.255.255.252
isis enable 1
isis circuit-level level-2
mpls
mpls ldp
#
interface GigabitEthernet0/0/1
#
interface NULL0
#
interface LoopBack0
description is-is/ bgp/ mpls ldp router-id
ip address 10.0.0.2 255.255.255.255
isis enable 1
#
user-interface con 0
user-interface vty 0 4
user-interface vty 16 20
#
return

```

Enrutador Router_3

```

#
sysname Router3
#
arp learning strict
#

```

```
undo http server enable
#
nd user-bind detect retransmit 0 interval 0
#
mpls lsr-id 10.0.0.3
mpls
#
mpls ldp
#
#
aaa
authentication-scheme default
#
authorization-scheme default
#
accounting-scheme default
#
domain default
#
isis 1
is-level level-2
network-entity 01.0000.0000.0000.0000.0010.0000.0000.0003.00
import-route direct
#
interface Ethernet0/0/0
description Conexion_PE2
ip address 172.16.0.9 255.255.255.252
isis enable 1
isis circuit-level level-2
mpls
mpls ldp
#
interface Ethernet0/0/1
#
```

```
interface GigabitEthernet0/0/0
description Conexion_P1
ip address 172.16.0.6 255.255.255.252
isis enable 1
isis circuit-level level-2
mpls
mpls ldp
#
interface GigabitEthernet0/0/1
#
interface NULL0
#
interface LoopBack0
description is-is/ bgp/ mpls ldp router-id
ip address 10.0.0.3 255.255.255.255
isis enable 1
#
user-interface con 0
user-interface vty 0 4
user-interface vty 16 20
#
return
```

Enrutador Router_4

```
#
sysname Router4
#
arp learning strict
#
undo http server enable
#
nd user-bind detect retransmit 0 interval 0
#
ip vpn-instance Customer_A
```

```

ipv4-family
  route-distinguisher 11:111
  vpn-target 11:111 export-extcommunity
  vpn-target 11:111 import-extcommunity
#
ip vpn-instance Customer_B
  ipv4-family
    route-distinguisher 22:222
    vpn-target 22:222 export-extcommunity
    vpn-target 22:222 import-extcommunity
#
mpls lsr-id 10.0.0.4
mpls
#
mpls ldp
#
aaa
  authentication-scheme default
#
  authorization-scheme default
#
  accounting-scheme default
#
  domain default
#
isis 1
  is-level level-2
  network-entity 01.0000.0000.0000.0000.0010.0000.0000.0004.00
  import-route direct
#
interface Ethernet0/0/0
  description Conexion_LWS2
#
interface Ethernet0/0/0.1

```

```
vlan-type dot1q 10
ip binding vpn-instance Customer_A
ip address 192.168.1.5 255.255.255.252
#
interface Ethernet0/0/1
description Conexion_P2
ip address 172.16.0.10 255.255.255.252
isis enable 1
isis circuit-level level-2
mpls
mpls ldp
#
interface Ethernet0/0/2
ip binding vpn-instance Customer_B
ip address 192.168.131.5 255.255.255.252
#
interface NULL0
#
interface LoopBack0
description is-is/ bgp/ mpls ldp router-id
ip address 10.0.0.4 255.255.255.255
isis enable 1
#
bgp 65001
group RR internal
peer RR connect-interface LoopBack0
peer 10.0.0.1 as-number 65001
peer 10.0.0.1 group RR
#
ipv4-family unicast
undo synchronization
import-route direct
import-route static
peer RR enable
```

```

peer 10.0.0.1 enable
peer 10.0.0.1 group RR
#
ipv4-family vpnv4
policy vpn-target
peer RR enable
peer 10.0.0.1 enable
peer 10.0.0.1 group RR
#
ipv4-family vpn-instance Customer_A
import-route direct
import-route static
#
ipv4-family vpn-instance Customer_B
import-route direct
import-route static
#
ip route-static vpn-instance Customer_B 192.168.3.0 255.255.255.0
192.168.131.6
#
user-interface con 0
user-interface vty 0 4
user-interface vty 16 20
#
return

```

Enrutador Router_5

```

#
sysname Router5
#
undo http server enable
#
nd user-bind detect retransmit 0 interval 0
#

```



```
aaa
authentication-scheme default
#
authorization-scheme default
#
accounting-scheme default
#
domain default
#
interface Ethernet0/0/0
ip address 192.168.2.1 255.255.255.0
#
interface Ethernet0/0/1
ip address 192.168.131.2 255.255.255.252
#
interface NULL0
#
ip route-static 0.0.0.0 0.0.0.0 192.168.131.1
#
user-interface con 0
user-interface vty 0 4
user-interface vty 16 20
#
return
```

Enrutador Router_6

```
#
sysname Router6
#
undo http server enable
#
nd user-bind detect retransmit 0 interval 0
#
aaa
```

```
authentication-scheme default
#
authorization-scheme default
#
accounting-scheme default
#
domain default
#
interface Ethernet0/0/0
ip address 192.168.3.1 255.255.255.0
#
interface Ethernet0/0/1
ip address 192.168.131.6 255.255.255.252
#
interface NULL0
#
ip route-static 0.0.0.0 0.0.0.0 192.168.131.5
#
user-interface con 0
user-interface vty 0 4
user-interface vty 16 20
#
return
```