UNIVERSIDAD DE VALLADOLID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

EVALUACIÓN DE HERRAMIENTAS DE SIMULACIÓN PARA UNA ASIGNATURA DE DISEÑO Y CONFIGURACIÓN DE REDES

GRADO EN INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS DE TELECOMUNICACIÓN - MENCIÓN EN TELEMÁTICA

21 DE ABRIL DE 2020

AUTOR GEMMA MARÍA GARCÍA MÍNGUEZ
TUTORES MARÍA JESÚS VERDÚ PÉREZ
LUISA MARÍA REGUERAS SANTOS

TÍTULO: **EVALUACIÓN DE HERRAMIENTAS DE SIMULACIÓN PARA UNA ASIGNATURA DE DISEÑO Y CONFIGURACIÓN DE REDES**

AUTOR: **GEMMA MARÍA GARCÍA MÍNGUEZ**

TUTORES: **DÑA. MARÍA JESÚS VERDÚ PÉREZ**

DÑA. LUISA MARÍA REGUERAS SANTOS

DEPARTAMENTO: TEORÍA DE SEÑAL Y COMUNICACIONES E INGENIERÍA TELEMÁTICA

TRIBUNAL:

PRESIDENTE:	**D. JUAN PABLO DE CASTRO FERNÁNDEZ**
VOCAL:	**DÑA. MARÍA JESÚS VERDÚ PÉREZ**
SECRETARIO:	**DÑA. LUISA MARÍA REGUERAS SANTOS**
SUPLENTE:	**D. FRANCISCO MERINO CAMINERO**
SUPLENTE:	**D. JAIME GÓMEZ GIL**

RESUMEN

La simulación de red es una herramienta muy útil para reducir el tiempo de diseño y para la comprobación de configuraciones. En la actualidad, existen numerosos programas de simulación de diseño y configuración de red que se encuentran en el mercado, a disposición de todos los usuarios.

El objetivo de este trabajo fin de grado ha sido encontrar un simulador que permita desarrollar todos los objetivos y competencias de la asignatura "Laboratorio de Diseño y Configuración de Redes" que se imparte en el tercer curso del Grado en Ingeniería de Tecnologías Específicas de Telecomunicación - Mención en Telemática, ofrecido por la E.T.S. de Ingenieros de Telecomunicación de la Universidad de Valladolid.

Después del estudio de las características, ventajas e inconvenientes de un importante número de herramientas de simulación que están disponibles en el mercado, decidimos que el simulador que cumple la mayoría de los requisitos que se pedían es GNS3.

GNS3 es un simulador de software libre, que nos ha permitido la reproducción de las prácticas de la asignatura. Dicho simulador cuenta con muchas ventajas ya que permite la configuración y simulación de una amplia variedad de escenarios de red, pero también presenta algunos puntos débiles, especialmente en lo referente a la representación y análisis gráfico, por lo que es conveniente complementar esta herramienta con otro tipo de software como los analizadores de protocolos.

PALABRAS CLAVE

Simuladores de redes; software libre; direccionamiento; protocolos de encaminamiento; aprendizaje a través de prácticas.

ABSTRACT

Network simulation is a useful tool to reduce design time and for configuration testing. Nowadays, there are many simulation programs on the market for network design and configuration.

The objective of this final project has been to find a simulator that allows us to develop the objectives and competences of the course “Laboratory of Network Design and Configuration”. This course is taught in the third year of the Degree in Telecommunication Technologies Engineering– Speciality in Telematics by the University of Valladolid.

Once we completed the study of the characteristics, advantages and disadvantages of a big number of simulation tools that are available on the market, we decided that the best simulator was GNS3. GNS3 meets most of our requirements.

GNS3 is a free software simulator. The program has allowed us to reproduce the practices of the course. GNS3 has many advantages, since it allows us to configure and simulate many network scenarios. However, it has some weak points too, especially in representation and graphic analysis. Therefore, we would recommend using this tool with other type of software such as protocol analyzers.

KEYWORDS

Network simulation; free software; addressing; routing protocols; learning through practice.

Contenido

Capítulo 1 – Introducción

1.1. INTRODUCCIÓN

El propósito de este trabajo consiste en el estudio de una serie de simuladores de red con el objetivo de encontrar la forma de sustituir el programa que se utiliza en la actualidad en la asignatura "Laboratorio de Diseño y Configuración de Redes" que se imparte en el tercer curso del Grado en Ingeniería de Tecnologías Específicas de Telecomunicación - Mención en Telemática, ofrecido por la E.T.S. de Ingenieros de Telecomunicación de la Universidad de Valladolid.

A continuación, en la tabla 1, se muestran algunos de los datos básicos de la asignatura.

Asignatura	LABORATORIO DE DISEÑO Y CONFIGURACIÓN DE REDES		
Materia	PLANIFICACIÓN Y GESTIÓN DE REDES Y SERVICIOS TELEMÁTICOS		
Módulo	MATERIAS ESPECÍFICAS DE LA MENCIÓN EN TELEMÁTICA		
Titulación	GRADO EN INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS DE TELECOMUNICACIÓN – MENCIÓN EN TELEMÁTICA		
Periodo de impartición	2º CUATRIMESTRE	**Tipo/Carácter**	OPTATIVA (OBLIGATORIA DE LA MENCIÓN)
Nivel/Ciclo	GRADO	**Curso**	3º
Créditos ECTS	6 ECTS		

Tabla 1. Descripción de la asignatura.

La asignatura “Laboratorio de Diseño y Configuración de Redes” es una asignatura de orientación eminentemente práctica dentro del bloque de materias específicas de telemática, donde se aplican conceptos de redes de datos que han sido vistos anteriormente en otras asignaturas del plan de estudios, como “Arquitectura de Redes, Sistemas y Servicios”, “Redes y Servicios Telemáticos” y “Conmutación y Encaminamiento”. Las 60 horas de actividades presenciales de la asignatura se imparten durante 15 semanas con 4 horas semanales de clase distribuidas de la siguiente manera: 3 horas de laboratorio y 1 hora de seminario.

En el seminario se realizan técnicas de trabajo colaborativo (como el *jigsaw*) y se analizan diferentes casos de estudio desde un enfoque práctico; y en el laboratorio se realizan prácticas de simulación donde se aplican los conceptos vistos en el seminario, atendiendo a diferentes aspectos del diseño y configuración de redes. Para ello, actualmente, en el laboratorio, se utiliza una herramienta de simulación de redes, Riverbed Modeler, en su versión académica (Riverbed Modeler Academic Edition).

Riverbed Modeler Academic Edition es la versión académica y gratuita de Riverbed Modeler, que si bien es cierto que reúne bastantes características, no dispone de todos los servicios que tiene la versión completa, la cual conlleva un pago de licencia que supondría un amplio gasto para la universidad. Asimismo, la versión académica tiene previsto dejar de dar servicio en 2021. En este sentido, se pretende eliminar la dependencia que se tiene con el fabricante, así como evitar el pago de unas licencias de uso. La solución que se busca es un simulador de red preferiblemente de software libre con el fin de eliminar el coste de compra de producto o de licencias y la dependencia con los fabricantes.

Mediante el uso del simulador se pretende que los alumnos aprendan de forma práctica los conceptos abordados durante la asignatura. De forma más concreta, con la ayuda del simulador se busca que el alumnado, una vez finalice la asignatura, adquiera las siguientes competencias y conocimientos (Regueras, 2019):

- Aplicar los conceptos adquiridos sobre protocolos, redes y servicios telemáticos en la configuración y puesta en marcha de un sistema telemático real.
- Conocer los principios de diseño de una red de comunicaciones.
- Analizar cómo un cambio de diseño en la red puede afectar a su comportamiento.
- Simular y comparar el funcionamiento de diferentes diseños y configuraciones de red.
- Utilizar una herramienta de simulación para analizar el funcionamiento de una red.

- Diseñar una arquitectura de red, un plan de direccionamiento IP y un esquema de interconexión para una red sencilla.
- Elegir el protocolo de encaminamiento y sus parámetros de configuración en función de los requerimientos de la red.
- Configurar equipos de interconexión reales.
- Consultar y utilizar la documentación técnica proporcionada por los fabricantes de los dispositivos de interconexión.
- Conocer los elementos de los que consta un Sistema de Cableado Estructurado (SCE).
- Diseñar un Sistema de Cableado Estructurado aplicando la normativa existente.
- Diseñar una red de comunicaciones sobre el SCE previamente diseñado, y configurar, validar y depurar su funcionamiento en una herramienta de simulación.

Este trabajo se divide en dos partes. La primera consiste en la investigación y análisis de distintos simuladores de red, estudiando sus características, comprobando sus ventajas e inconvenientes, y su utilidad hacia los objetivos de la asignatura con el fin de seleccionar el simulador que cumple de forma más satisfactoria todos los requisitos establecidos. En la segunda parte, después del estudio inicial y una vez elegido el simulador, vamos a comprobar de manera práctica si son realizables las prácticas de la asignatura.

La metodología que se ha seguido durante la realización del trabajo es la siguiente:

- Análisis del problema.
- Estudio de Riverbed Modeler, e investigación de los diversos programas software que se podrían usar como alternativa.
- Comparativa de las alternativas: características, ventajas e inconvenientes.
- Elección de un software y estudio exhaustivo del mismo.
- Realización de las prácticas.
- Conclusiones.

La investigación ha estado basada en la búsqueda de información en libros, manuales e Internet. También se han realizado consultas a los fabricantes y lectura de guías. La parte práctica ha consistido en la instalación de los simuladores e implementación de las prácticas de la asignatura.

1.2. PLANTEAMIENTO DEL PROBLEMA

El objetivo de este trabajo es encontrar un simulador de redes que permita desarrollar todas o una gran mayoría de las competencias que se necesitan cubrir en la asignatura de Laboratorio y Diseño de Configuración de Redes.

La materia desarrolla una serie de competencias generales y específicas, las cuales se recogen a continuación.

Competencias generales (Regueras, 2018):

- Conocimiento, comprensión y capacidad para aplicar la legislación necesaria durante el desarrollo de la profesión de Ingeniero Técnico de Telecomunicación y facilidad para el manejo de especificaciones, reglamentos y normas de obligado cumplimiento.
- Capacidad para resolver problemas con iniciativa, creatividad y razonamiento crítico.
- Capacidad para diseñar y llevar a cabo experimentos, así como analizar e interpretar datos.
- Capacidad para elaborar informes basados en el análisis crítico de la bibliografía técnica y de la realidad en el campo de su especialidad.
- Capacidad para trabajar en un grupo multidisciplinar y multilingüe, responsabilizándose de la dirección de actividades objeto de los proyectos del ámbito de su especialidad y consiguiendo resultados eficaces.
- Capacidad, y compromiso ético en la elaboración de soluciones de ingeniería y en las diversas situaciones de gestión de recursos humanos y de gestión económica, así como capacidad para comprender el impacto de las soluciones de Ingeniería en un contexto social global.
- Capacidad de organización, planificación y gestión del tiempo.
- Capacidad para comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas relacionadas con las telecomunicaciones y la electrónica.
- Capacidad para trabajar en cualquier contexto, individual o en grupo, de aprendizaje o profesional, local o internacional, desde el respeto a los derechos fundamentales, de igualdad de sexo, raza o religión y los principios de accesibilidad universal, así como la cultura de paz.

Competencias específicas (Regueras, 2018):

- Capacidad para aplicar las técnicas en que se basan las redes, servicios y aplicaciones telemáticas, tales como sistemas de gestión, señalización y

conmutación, encaminamiento y enrutamiento, seguridad (protocolos criptográficos, tunelado, cortafuegos, mecanismos de cobro, de autenticación y de protección de contenidos), ingeniería de tráfico (teoría de grafos, teoría de colas y teletráfico) tarificación y fiabilidad y calidad de servicio, tanto en entornos fijos, móviles, personales, locales o a gran distancia, con diferentes anchos de banda, incluyendo telefonía y datos.

- Capacidad de construir, explotar y gestionar servicios telemáticos, utilizando herramientas analíticas de planificación, de dimensionado y de análisis.

- Capacidad de seguir el progreso tecnológico de transmisión, conmutación y proceso para mejorar las redes y servicios telemáticos.

En la parte de prácticas de laboratorio, parte principal de la asignatura con un 75% de la carga de trabajo del alumno, se trabaja de forma más concreta sobre los siguientes temas:

- Direccionamiento y conectividad IP: Establecer un esquema de direccionamiento, asignar direcciones IP y máscaras a los hosts y routers, y comprobar la conectividad entre puntos, mediante comandos de línea básicos o desde la propia interfaz gráfica del simulador.

- Encaminamiento: Comparar, elegir y configurar protocolos de encaminamiento, centrándose fundamentalmente en los protocolos de encaminamiento:

    - RIP[1] (*Routing Information Protocol*) (Hedrick, 1988) – Como protocolo de encaminamiento interno utilizado para encontrar la mejor ruta extremo a extremo a través de una red mediante el uso de un algoritmo de camino más corto basado en el número de saltos (Inc, Techopedia, 2019), se analiza su funcionamiento, cómo se actualizan las tablas de encaminamiento y las diferencias entre usar RIPv1 (obsoleta) y RIPv2[2].

    - OSPF[3] *(Open Shortest Path First)* (Moy, 1998) – Mediante el análisis de este protocolo de encaminamiento interno basado en estado del enlace (Nieto Leon, 2013), se profundiza en la definición de Sistemas Autónomos y el balanceo de tráfico para el cálculo y determinación de rutas.

- VLAN (*Virtual Local Área Network* - red de área local virtual): Crear e implementar redes lógicas independientes, dentro de una única red física. Se comprueba el intercambio de información, la interconexión existente y cómo es la estructura de la red.

[1] RFC 1058- https://tools.ietf.org/html/rfc1058 (último acceso: 24 de abril de 2019)
[2] RFC 2453- https://tools.ietf.org/html/rfc2453 (último acceso: 24 de abril de 2019)
[3] RFC 2328- https://tools.ietf.org/html/rfc2328 (último acceso: 24 de abril de 2019)

- SCE (Sistemas de Cableado Estructurado): Diseñar una red, siguiendo las especificaciones dadas para la realización de un proyecto de SCE, incluyendo la normativa legal relacionada con el diseño de redes y la implementación de dicha red.

1.3. CONTENIDO DE LA MEMORIA

Esta memoria está divida en varios capítulos. En el capítulo 2 afrontaremos, en primer lugar, lo que es un simulador de redes y su uso. A continuación, estudiaremos los distintos simuladores existentes en el mercado que cumplen las especificaciones expuestas. Para completar este capítulo se realiza una comparativa de todos los simuladores y una toma de decisiones explicando de forma razonada cuál se ha elegido finalmente.

El siguiente capítulo es el correspondiente al simulador seleccionado, GNS3, con una descripción más detallada del mismo, definiendo sus características, modo de instalación y funcionamiento, así como también la forma en que se pueden realizar las prácticas con los resultados conseguidos.

El último capítulo contiene las conclusiones y líneas futuras, explicando si la opción elegida cumple realmente con todo lo exigido en el planteamiento del problema y cómo podría mejorarse.

Capítulo 2 – Simuladores de Red

2.1. ¿QUÉ ES UN SIMULADOR DE RED?

Un simulador es una herramienta que permite la recreación de escenarios, evitando su construcción. Va a permitirnos demostrar si el diseño teórico es correcto y puede ser desarrollado, sometiéndole a todo tipo de pruebas para ver errores en el diseño y realizar las mejoras.

La anterior descripción de un simulador es muy general y sirve para definir todos los simuladores. Las siguientes definiciones son más técnicas y relacionadas con simuladores de redes de telecomunicaciones.

> *"Es una aplicación que permite al usuario/administrador de una red, diseñar un sistema de redes entre computadoras, switches, router, impresoras, servidores, etc. Todo esto se realiza en nuestro monitor haciendo conexiones de cables, e interconectándolos entre sí, para luego realizar una prueba virtual de la compatibilidad de nuestras conexiones. Además, es posible configurar de manera individual cada periférico."* (Blogspot.com, 2012)

> *"Es la imitación del funcionamiento de un sistema real durante un intervalo de tiempo. La simulación se basa en un modelo de la realidad. Además, la simulación también se puede usar como estrategia en la etapa de diseño, antes de que el sistema sea construido, o se puede usar en ambos casos a la vez, para predecir el efecto de un cambio y diseñar variantes de un sistema actual."* (oscarmanuuel, 2014)

Los simuladores de redes admiten la comprobación de los diseños, permiten el ahorro de costes (tiempo y dinero), evitando la construcción real de la red hasta que el diseño sea el correcto y esté validado. Otra ventaja de los simuladores es la capacidad de recrear escenarios complejos y de poder repetir la simulación hasta lograr el objetivo.

La ventaja de las repeticiones de las simulaciones se puede volver una desventaja cuando el modelo a diseñar es complejo y necesita un número de repeticiones excesivas para conseguir el objetivo, aumentando el coste por la cantidad de tiempo y trabajo invertido en las simulaciones.

2.2. DEFINICIÓN DE REQUISITOS

En esta etapa se establecen las bases principales para la selección de los simuladores que entren en el estudio.

Algunas características que buscamos para el simulador son:

- Código abierto (*Open Source*) - Se quiere evitar un pago de licencia y de sus futuras actualizaciones, así como también tener que estar a expensas de la caducidad de la licencia y depender de un fabricante. Se intenta que el gasto sea cero y que se garantice la posibilidad de un uso futuro de la herramienta.

- Posibilidad de implementar todas las técnicas y protocolos sobre los que se desarrollan las competencias de la asignatura. No es necesario que las prácticas sean exactamente las mismas que se realizan en la actualidad, pero sí que se puedan adecuar.

- Es preferible que funcionen en los sistemas operativos Windows y Linux, ya que la mayoría de los estudiantes usan alguno de estos dos sistemas. Un simulador con versiones para ambos sistemas será más accesible.

No son características únicas y no son excluyentes entre sí.

2.3. ESTUDIO DE SIMULADORES

El número de simuladores de redes existentes en la actualidad es extenso, y algunos no cumplirán los objetivos, pero no los descartaremos hasta haber estudiado sus requisitos y principales características. En concreto, éstos son los simuladores que vamos a estudiar en este trabajo:

- CISCO VIRL

- NS-2

- JIMSIM

- CORE

- NETSIMK
- KIVA NS
- CLOONIX
- MININET
- CISCO PACKET TRACER
- RIVERBED MODELER
- OMNET++
- NETSIM
- MARIONNET
- GNS3

2.3.1 CISCO VIRL

CISCO VIRL[4] (VIRTUAL INTERNET ROUTING LAB) es un simulador orientado a la certificación Cisco; es una plataforma de virtualización y organización de redes. Permite la implementación de modelos de redes existentes y planificadas en un entorno seguro. El acceso a este simulador en su versión de prueba se puede obtener desde la propia página del simulador.

VIRL ofrece una experiencia perfecta para el estudio de la tecnología CISCO, ya que el código de software es el de CISCO IOS, usando su propia gama de switches, routers, hubs…, dando una perfecta oportunidad de usar sus herramientas para aprender sobre las redes. Su uso está dirigido tanto a profesionales como a estudiantes del ámbito de las telecomunicaciones.

Las principales características generales que nos ofrece el simulador de CISCO VIRL son:

- Creación de modelos y escenarios de redes del mundo real y futuras.
- Generación automática de diferentes configuraciones.
- Visualización de numerosos protocolos.
- Uso de sistema operativo CISCO IOS, con sus equipos.

[4] http://virl.cisco.com y https://learningnetworkstore.cisco.com/virtual-internet-routing-lab-virl (último acceso: 16 de marzo de 2019).

- Posibilidad de conectar entornos virtuales y físicos para formar entornos de prueba y desarrollo, incluyendo máquinas virtuales y servidores de terceros.

Incluye una comunidad virtual que proporciona recursos, documentación, foros de discusión, recursos y ayuda para la solución de problemas.

2.3.2 NS-2

NS-2[5] es un simulador de eventos discreto dirigido a la investigación de redes. NS-2 proporciona un soporte sustancial para la simulación de TCP, el encaminamiento y protocolos de multidifusión a través de redes cableadas e inalámbricas (locales y satelitales).

NS-2 comenzó en 1989 como una variante del simulador de red REAL y ha evolucionado sustancialmente en los últimos años. Siempre ha incluido contribuciones sustanciales de otros investigadores, con numerosas colaboraciones de distintas universidades de todo el mundo.

Este simulador está programado en C y puede ser instalado en sistemas operativos Unix y Linux (Debian, Ubuntu). Para instalarse en Windows requiere de la aplicación Cygwin.

En 2005, apareció una nueva versión del simulador NS-2, denominado NS-3, el cual es incompatible con NS-2. Se partió de cero en la programación, usando el lenguaje de programación C++. Al igual que la versión anterior, fue desarrollado, fundamentalmente por instituciones académicas, principalmente universidades. En el desarrollo de NS-3 se consiguió que este nuevo simulador ofreciera un número mayor de eventos de simulación y mayores prestaciones.

2.3.3 JIMSIM SIMULATOR

J-Sim[6] es un simulador de código abierto conocido anteriormente como JavaSim. Es un entorno de simulación de composición basado en componentes. Su construcción se basó en la noción del Modelo de Programación de Componentes Autónomos.

El modelo de simulación que utiliza es un modelo de red generalizado de comunicación de paquetes. Dicho modelo define la estructura genérica de un nodo (router, host…) y los componentes de red genéricos, los cuales se pueden usar como clases de bases para implementar protocolos en varias capas.

[5] https://www.isi.edu/nsnam/ns/ Descarga del simulador: https://sourceforge.net/projects/nsnam (último acceso: 16 de marzo de 2019)

[6] Descarga del simulador: https://sites.google.com/site/jsimofficial/downloads (último acceso: 16 de marzo de 2019)

Está desarrollado enteramente en Java, lo que hace que el entorno sea neutral, extensible y reutilizable, independiente de la plataforma. Proporciona una interfaz de script para permitir la integración con diferentes lenguajes de programación.

Este simulador es dual al lenguaje de simulación de NS-2, pudiendo ser utilizadas algunas de las clases de uno en el otro mediante la importación de éstas.

Las características generales que tiene el simulador son las siguientes:

- Modelo de programación basado en componentes de acoplamiento flexible.
- Capaz de hacer los procesos de simulación en tiempo real.
- Implementación de un conjunto de protocolos de servicios integrados, diferenciados.
- Entorno de lenguaje dual que permite la configuración automática y la supervisión de línea.
- Implementación de clases de interfaz genéricas para la simulación.
- Realización parcial de la emulación de la red.

Asimismo, el programa ofrece paquetes en el que están incluidos diferentes ejemplos donde se puede ver el código de las simulaciones que se ejecuten (Chen, Ye, Guanghui , Chunyu , & Hwangnam, 2005).

2.3.4 CORE

CORE[7] (*Common Open Research Emulator*) es una herramienta para emular redes en una o más maquinas. Puede conectar estas redes emuladas a redes que se ejecutan en tiempo real. Consta de una GUI para dibujar topologías de máquinas virtuales ligeras y módulos de Python para la emulación de red de *scripting*.

Ha sido desarrollado por un grupo de investigación de tecnologías de redes que forma parte de la división de investigación y tecnología de Boeing. El laboratorio de Investigación Naval está apoyando un mayor desarrollo de este proyecto de código abierto.

Las principales características de CORE son:

- Eficiencia y escalabilidad, al tener el laboratorio de red en una carpeta ejecutable.

[7]https://nrl.navy.mil/itd/ncs/products/core
Descarga del simulador: https://downloads.pd.itd.nrl.navy.mi/core (último acceso: 16 de marzo de 2019)

- GUI fácil de usar.
- Configuración y control centralizados.
- Ejecuta aplicaciones y protocolos sin modificarlos.
- Conexiones en tiempo real a redes reales, mediante una configuración del hardware en bucle, y una distribución con múltiples simuladores de COREs.
- Altamente personalizable.

En la página principal de CORE se pueden ver distintas demostraciones y los diferentes enlaces donde descargar el simulador. (U.S. Naval Research Laboratory, 2010)

2.3.5 NETSIMK

NetSimK[8] es un simulador para la enseñanza y el aprendizaje de los routers Cisco. Surgió de la frustración con los productos de simulación de redes existentes. En el diseño de NetSimK, se buscaba un producto más barato, sencillo y versátil. Se desarrolló a lo largo de varios años con la ayuda de muchos miembros del personal de soporte de Cisco. Se está mejorando y actualizando continuamente, con más funciones de IOS que se van agregando en cada versión. Dichas mejoras provienen de los comentarios de los clientes, estudiantes, etc.

Las características comerciales que nos ofrece este simulador son:

- Los diseños de los escenarios son realistas.
- La interfaz es sencilla e intuitiva.
- Se centra en enseñar principios, sin necesidad de configurar opciones innecesarias de dispositivos.
- Implementa todos los requisitos relevantes del IOS para obtener las certificaciones de Cisco. Así como la recuperación de contraseña.
- Telnet entre dispositivos.
- Los dispositivos se pueden apagar y encender. Los datos que no se escriben en la memoria RAM no volátil se perderán, al igual que ocurre en un dispositivo real.
- La red puede tener tantos dispositivos como sea necesario, sin límite en ningún dispositivo.

[8] www.netsimk.com
Descarga del simulador: http://netsimk.com/DownloadFile.htm (último acceso: 16 de marzo de 2019)

Actualmente hay una versión de evaluación que es completamente funcional para la gente, y su disponibilidad es gratuita y disponible desde la propia web del simulador.

2.3.6 KIVA NS

KivaNS[9] (KIVA NETWORK SIMULATOR) es una aplicación gratuita y de código abierto basada en Java, que permite especificar diferentes esquemas de redes de datos y simular el encaminamiento de paquetes a través de estas redes.

KivaNS está orientada principalmente a la simulación del comportamiento del protocolo IP y especialmente al tratamiento de los datagramas y el encaminamiento de éstos por una red.

El objetivo principal del entorno es ayudar a diseñar y comprender el funcionamiento de las redes de datos, y en especial el encaminamiento de paquetes en la arquitectura TCP/IP, sin necesidad de la infraestructura real y de herramientas de análisis de tráfico. También es capaz de simular distintos tipos de errores en el funcionamiento de las redes (pérdida de paquetes, fallos en las tablas de encaminamiento…).

Está compuesto por partes implementadas en Java:

- Una API que ofrece un motor de simulación.
- Una completa interfaz gráfica que hace uso de la API de simulación.

El entorno funciona en múltiples sistemas operativos, como puede ser GNU/Linux o Microsoft Windows. La interfaz de usuario está formada por un conjunto de clases que se deben descargar y ejecutar en el equipo; con estas clases especificamos las diferentes topologías de redes, equipos de la red…. (Gallardo, Sebastián Melzi, & de Dios, s.f.)

2.3.7 CLOONIX

Cloonix[10] es una plataforma de simulación de routers y host de Linux. Se encarga fundamentalmente del encapsulamiento de las aplicaciones, hosts, y de la red. Está basada en KVM o UML. Se puede instalar en diferentes sistemas operativos como Debian, Fedora, OpenSUSE y sus distribuciones derivadas.

Cloonix ofrece a los estudiantes e investigadores la posibilidad de investigar varias tecnologías de Internet como el Sistema de Nombres de Dominio (DNS).

[9] Descarga del simulador: http://www.disclab.ua.es/kiva (último acceso: 16 de marzo de 2019)
[10] Descarga del simulador: http://www.clownix.net (último acceso: 16 de marzo de 2019)

Puede simular una red con varias máquinas virtuales reconfigurables en un solo ordenador, donde las máquinas virtuales pueden ser diferentes distribuciones de Linux.

Asimismo, ofrece el servicio de monitorización de las actividades de la red a través de Wireshark.

Las principales características de este simulador de red son:

- Herramientas NS basadas en GUI WM basada en KVM.
- Las máquinas virtuales y los clientes están basados en Linux.
- Servidores para el monitoreo de las máquinas virtuales y las actividades de red que conseguimos con Wireshark.

Proporciona una interfaz gráfica de usuario fácil de usar, además de un equipo de desarrollo activo, que actualiza la herramienta cada poco tiempo, respondiendo muy bien a las opiniones de los usuarios. (Maiti, 2018)

2.3.8 MININET

Mininet[11] es un emulador de red que crea una red de hosts virtuales, switches, controladores y enlaces. Los hosts de Mininet ejecutan el software de red estándar de Linux, y sus switches son compatibles con OpenFlow para que el encaminamiento sea personalizado. Son redes altamente flexibles y definidas por software.

Mininet apoya la investigación, el desarrollo, el aprendizaje, la creación de prototipos, las pruebas, la depuración y cualquier otra tarea que pueda beneficiarse de tener una red experimental completa en una computadora, pudiéndose descargar gratuitamente.

Las características que ofrece el simulador Mininet son las siguientes:

- Proporciona un banco de pruebas de redes tanto simples como complejas para el desarrollo de aplicaciones OpenFlow.
- Permite que múltiples desarrolladores concurrentes trabajen independientemente en la misma topología.
- Se puede desarrollar topologías personalizas arbitrarias, proporciona una API de Python directa y extensible para la creación y experimentación de redes.

Mininet proporciona una forma fácil de obtener el comportamiento correcto del sistema y en medida que lo admita el hardware el rendimiento de la red, también permite la experimentación con las topologías.

[11] Descarga del simulador: http://mininet.org/download/ (último acceso: 16 de marzo de 2019)

Las redes de Mininet ejecutan código real, incluidas las aplicaciones de red estándar de Unix/Linux, así como el kernel y la pila de redes reales de Linux (incluidas las extensiones de kernel que pueda tener disponibles, siempre que sean compatibles con los espacios de nombres de red).

Es un simulador muy útil, ofrece una amplia escalabilidad, es barato, fácil de instalar y se puede reconfigurar. Es capaz de ejecutar código real, sin modificar, y se conecta de manera sencilla a las redes reales ofreciendo un rendimiento interactivo.

El problema que tiene es que no puede exceder el uso de la CPU o el ancho de banda disponible en un solo servidor, y también que no puede ejecutar switches o aplicaciones OpenFlow que no sean compatibles con Linux. (Mininet Team, 2018)

2.3.9 PACKET TRACER

Packet Tracer[12], de Cisco, es una potente plataforma de simulación de redes que estimula a los alumnos a experimentar con el comportamiento de las redes y a formular preguntas sobre situaciones hipotéticas. Está pensada para ser utilizada como complemento a laboratorios con equipos físicos; los estudiantes pueden crear una red con un número casi ilimitado de dispositivos, lo que estimula la práctica, detección y solución de problemas. (Cisco, s.f.)

El simulador permite la configuración de dispositivos de red, así como la detección y la corrección de errores en sistemas de comunicación.

Las principales características generales que nos ofrece este simulador son:

- Es una herramienta útil para la enseñanza de Redes de Comunicación.
- Su interfaz es fácil de manejar, además de incluir una extensa documentación (documentos, tutoriales, ejemplos prácticos…) para el manejo del simulador.
- Permite ver el desarrollo por capas del proceso de trasmisión y recepción de paquetes de acuerdo con el modelo de referencia OSI.
- Puede realizar simulaciones de diferentes protocolos y hacer diagnósticos básicos de las conexiones entre los dispositivos del modelo de la red.

2.3.10 RIVERBED MODELER

Riverbed Modeler[13] es un simulador cuya principal función, como el resto de los simuladores estudiados, es la eliminación de pruebas reales para el análisis de las

[12] Descarga del simulador: www.netacad.com (último acceso: 16 de marzo de 2019)
[13] Descarga del simulador: www.riverbed.com (último acceso: 16 de marzo de 2019)

configuraciones diseñadas, consiguiendo comprobar los diseños y protocolos con el consiguiente ahorro de tiempo y dinero.

Riverbed Modeler fue desarrollado en 1984 en el Instituto de Tecnologías de Massachusetts (MIT) por investigadores y científicos. Está basado en la teoría de redes de colas e incorpora las librerías necesarias para facilitar el modelado de las topologías de red, soportando un amplio rango de tecnologías de red tipo LAN, MAN y WAN.

Es una poderosa herramienta que permite simular sistemas de comunicación y así evaluar las prestaciones de una red bajo diversas condiciones de simulación como: flujos variables, pérdida de paquetes, caída de enlaces…

Ofrece diferentes características comerciales que permite a los usuarios del simulador (Riverbed Technology, 2019):

- Desarrollar tecnologías y protocolos inalámbricos propietarios.
- Evaluar las mejoras a los protocolos basados en estándares.
- Probar y demostrar diseños de tecnologías en escenarios realistas antes de la producción.
- Aumentar la productividad de la I+D de la red y acelerar el tiempo de comercialización.
- Interpretar los resultados de la simulación fácilmente utilizando tablas y gráficos intuitivos.

Este software se ofrece de manera gratuita a las universidades, a través de una versión académica, Riverbed Modeler Academic Edition. Dicha versión no incluye todos los servicios que tiene la versión de pago. Además, es necesario el pago para conseguir las actualizaciones del simulador, por esta razón se intenta encontrar otro software que elimine estos problemas.

2.3.11 OMNET++

OMNet++[14] es un marco de simulación de red de eventos discretos, modular y orientado a objetos. Tiene una arquitectura genérica, y puede ser utilizado para el modelado de distintos problemas:

- Modelado de redes de comunicación, protocolos y colas.
- Validación de arquitecturas de hardware.

[14] Descarga del simulador: https://omnetpp.org/download/ (último acceso: 16 de marzo de 2019)

- Evaluación de aspectos de rendimiento de sistemas de software complejos.
- El modelado y la simulación de cualquier sistema que en el que el enfoque del evento discreto sea adecuado.

No es un simulador concreto, sino que proporciona la infraestructura y las herramientas para escribir simulaciones. Los modelos de simulación se ensamblan a partir de componentes reutilizables que se denominan módulos. Los módulos están programados en C++ y hacen uso de la biblioteca de simulación.

OMNet++ puede ejecutarse bajo varias interfaces de usuario. Las interfaces de usuario gráficas y animadas son muy útiles para fines de demostración y depuración, y las interfaces de usuario de línea de comandos son las mejores para la ejecución por lotes.

El simulador, las interfaces de usuario y las herramientas son altamente portátiles. Se prueban en los sistemas operativos más comunes. (Varga & OpenSim Ltd., 2016)

2.3.12 NETSIM

NetSim[15] fue creado para la simulación de varios sistemas Cisco. Al simular el funcionamiento del software de Cisco los usuarios pueden adquirir conocimientos y aprender fácilmente cómo se trabaja dentro de Cisco.

Netsim nos ofrece tres versiones de su simulador, dependiendo de las necesidades de cada usuario.

- Netsim pro.
- Netsim standard.
- Netsim academic.

Va a permitir conectar el simulador a hardware real e interactuar con las aplicaciones en vivo. Se puede probar el rendimiento de las aplicaciones a través de una red virtual, expandiendo la red o diseñando una nueva según las necesidades.

Ofrece una gran variedad de características generales, comunes para todas las versiones que ofrece Netsim:

- Permite la medición de parámetros como el rendimiento, el retardo y la tasa de pérdida de paquetes.
- Compatible con el uso del Wireshark.
- Se puede utilizar para emular una amplia gama de tecnologías.

[15] Descarga del simulador: https://www.tetcos.com/download.html (último acceso: 16 de marzo de 2019)

- Uso de máquinas virtuales y mediciones en tiempo real.
- Implementación de múltiples protocolos.

Es una alternativa rentable a los emuladores de hardware que tienen costes altos, requisitos de configuración complicados y escalabilidad limitada. (TETCOS, s.f.)

2.3.13 MARIONNET

Marionnet[16] es un laboratorio de redes virtuales, permitiendo a los usuarios definir, configurar y ejecutar redes informáticas complejas sin necesidad de una configuración física. Solo requiere una máquina GNU/Linux. Nació en abril de 2005 como una simple interfaz de texto para Netkit, desde entonces se ha ido completando.

El simulador ha ido evolucionando, alcanzando un estado estable y se está utilizando con éxito para redes de enseñanza en las universidades de todo el mundo.

Los principales rasgos que ofrece este simulador, que trabaja con distintos sistemas operativos son:

- Reconfiguración dinámica de la red.
- Compatibilidad binaria con el software GNU/Linux de nivel usuario que se ejecuta en máquinas virtuales.
- Posibilidad de utilizar el servidor host X para ejecutar una aplicación gráfica.
- Sistema de archivos de copia en escritura, lo que permite economizar el uso de espacio en disco.
- Dispositivo “Gateway” para conectar la red virtual a la red del host.
- Interfaz gráfica de usuario intuitiva con el diagrama de red actualizado dinámicamente.
- Tiene buen rendimiento en esquemas con redes complejas.

El objetivo principal es la enseñanza de redes de ordenadores en laboratorios universitarios. También se puede usar como emulación de redes con fines de prueba o desarrollo; es una herramienta de simulación bastante fácil de configurar, rápida incluso con las configuraciones más complicadas y ofrece la posibilidad de revertir los cambios del sistema de archivos en las máquinas virtuales, lo cual lo hace bastante flexible. (Loddo & Saiu, 2007-2019)

[16] Descarga del simulador: https://marionnet.org/site/index.php/en/documentation/installation?id=253 (último acceso: 16 de marzo de 2019)

2.3.14 GNS3

GNS3[17] es un simulador utilizado por miles de ingenieros de redes de todo el mundo para emular, configurar, probar y solucionar problemas de redes virtuales y reales. Permite la ejecución de pequeñas y grandes topologías alojadas en múltiples servidores o incluso en la nube…

Es un software de código abierto que se puede descargar y utilizar de forma gratuita.

La arquitectura de GNS3 consta de dos componentes de software:

1. El software GNS3-all-in-one (GUI) - es la parte cliente y la interfaz gráfica de usuario (GUI).
2. La máquina virtual GNS3 (VM) -hay que ejecutar la máquina virtual GNS3 localmente en el ordenador usando un software de virtualización. Uso recomendable en la creación de topologías más avanzadas.

Tiene numerosas características:

- Software libre, de código abierto y sin cuotas.
- No hay limitaciones en la cantidad de dispositivos compatibles.
- Admite múltiples opciones de conmutación.
- Admite todas las imágenes VIRL (Cisco); dichas imágenes deben ser proporcionadas por Cisco, copiadas desde el dispositivo físico o compradas a través de una licencia.
- Soporta entornos de múltiples proveedores.
- Se puede ejecutar con o sin hipervisores (gratuitos y de pago).
- Dispositivos descargables, gratuitos y optimizados, para la simplificación de la implementación.
- Es escalable.
- Soporte nativo para Linux sin la necesidad de software adicional.

Existe una comunidad de GNS3, de la que los usuarios se pueden beneficiar, para aprender los conceptos de tecnologías de redes. (Bombal & Duponchelle, 2019).

A continuación, se muestra la Tabla 2, donde se describen de una forma resumida las principales características de los simuladores.

[17] Descarga del simulador: https://www.gns3.com/software (último acceso: 16 de marzo de 2019)

CARACTERÍSTICAS / SIMULADORES	LICENCIA	SISTEMA OPERATIVO	VERSIÓN	DESARROLLADOR	IDIOMA	LENGUAJE DE PROGRAMACIÓN
PACKET TRACER	Propietario	Linux, Windows	7.2	Cisco	Multilenguaje	-
RIVERBED MODELER	Propietario, Académica renovable cada 6 meses	Windows 7 o XP (como mínimo)	18.1	Riverbed (anteriormente OPNET)	Multilenguaje	C++
OMNet++	Licencia pública académica	Multiplataforma Linux, Mac Os/X, Windows	5.4.1	Andras Vargas, Opensim Ltd.	Inglés	C++
NETSIM	Distintas versiones: Standard, Pro, Academic	Windows 98/me/net/2000/Xp	v.11	-	Multilenguaje	Python
MARIONNET	Software gratuito, distribuido bajo licencia GNU	GNU/Linux Windows o Apple necesario una máquina virtual	v.0.6	Jean-Vincent Loddo	Inglés, italiano, francés	Lenguaje de programación Ocaml, un poco de lenguaje C
GNS3	Software libre	Multiplataforma: Windows, Mac, Linux	2.1.11	Jeremy Grossman	Multilenguaje	Python
MININET	BSD Open Source	Necesario un sistema de virtualización (Virtual Box). Funciona en Windows, Mac OSX, Linux, Ubuntu	Mininet 2.3.0d4	Bob Lantz y Brandon Heller	Inglés, español, multilenguaje	Python

CARACTERÍSTICAS / SIMULADORES	LICENCIA	SISTEMA OPERATIVO	VERSIÓN	DESARROLLADOR	IDIOMA	LENGUAJE DE PROGRAMACIÓN
CLOONIX	Código abierto	Cualquier distribución de Linux, basada en Debian o Ubuntu. Además de máquinas virtuales	v.42.7	Agplv3	Inglés	Java
KIVA NS	Código abierto	Diferentes sistemas operativos GNU/Linux o Windows	v.1.6, existe una versión para Android	Universidad de Alicante. Pablo Gil, Francisco A. Candelas-Herías	Español	Java
NETSIMK	Versión de evaluación gratuita	Windows	v.1.11	Desarrollado por un tutor de Cisco	Inglés	C++
CORE	Open Source	Linux (Fedora)	v.4.8	Departamento de la armada de la Marina	Inglés	Módulos Python
JIMSIM	Código abierto	Windows, Linux, Unix	v.1.3	Departamentos de Ciencias de la Computación e Ingeniería de la universidad de West Bohemia Republica Checa	Multilenguaje (inglés y francés)	C++ y Python
NS 2 (TAMBIEN VERSION NS 3)	Licencia Pública GNU	Multiplataforma (Windows y Unix)	NS-2 versión 2.34 y NS-3 versión 3.14 (incompatibles)	Comunidad	Inglés	C++ y Python
CISCO VIRL	Software Propietario, orientado al aprendizaje	Windows (a partir de Windows 7) y con máquina virtual en Windows y Linux	v.1.5.154 Diferentes versiones dependiendo del paquete que quiera el usuario	Cisco Systems	Inglés	Desarrollado en Java

Tabla 2. Características de los simuladores.

2.4. ANÁLISIS DE PRESTACIONES DE LOS SIMULADORES DE RED

2.4.1 FUNCIONALIDADES BÁSICAS

Los simuladores deben ofrecer la posibilidad de implementar direccionamiento IP (con posibilidad de *subnetting* de longitud variable), configurar redes virtuales de área local y configurar diferentes protocolos de encaminamiento (RIP y OSPF).

2.4.2 DIRECCIONAMIENTO IP

El direccionamiento IP proporciona un mecanismo para la asignación de identificadores a cada dispositivo conectado a una red. El protocolo IP es parte de la pila de protocolos TCP/IP[18] (Socolofsky & Kale, 1991). Todo dispositivo IP debería implementar dicha pila de protocolos y, en general, tener una dirección IP asignada (las direcciones pueden ser estáticas o dinámicas y también las podemos distinguir entre direcciones públicas o privadas).

La dirección IP es un número, que se puede representar en diferentes formatos. Estas direcciones son jerárquicas, con un identificador de red que indica a qué red pertenece el dispositivo y un identificador de host, que identifica el dispositivo dentro de la red. El direccionamiento IP es diferente en función de la versión del protocolo IP; las direcciones IP versión 4 (IPv4) son de 32 bits y se representan en formato decimal, y las direcciones IP versión 6 (IPv6) son de 128 bits y se representan en formato hexadecimal. (Gonzalez, 2012)

El direccionamiento IP moderno es inconcebible sin el concepto de *subnetting*, el cual nos va a permitir una mejor organización de las redes y un mejor aprovechamiento del espacio de direccionamiento, que en IPv4 es muy limitado. El *subnetting* consiste en dividir una red en una serie de subredes lógicas independientes. Con la formación de subredes conseguimos que cada una de las subredes trabajen de manera independiente, haciendo que trabaje la red con mayor celeridad en la transmisión y recepción de datos.

En los simuladores de redes, el direccionamiento IP es una tarea sencilla que prácticamente todos ellos soportan. La única diferencia entre los simuladores en

[18] RFC 1180 https://tools.ietf.org/html/rfc1180 (último acceso: 31 marzo de 2019)

relación con este aspecto es la interfaz gráfica que nos ofrecen cada uno de ellos. En unos simuladores, la configuración de parámetros de direccionamiento (dirección IP y máscara de subred asociada -por ejemplo, 255.255.255.0 o longitud de prefijo de red extendido -por ejemplo /24) se realizará mediante línea de comandos, mientras que en otros mediante menús o pestañas para introducir los datos. En la Figura 1 se muestra un ejemplo de dos interfaces de configuración, tanto por la línea de comandos como gráficamente a través de menús desplegables.

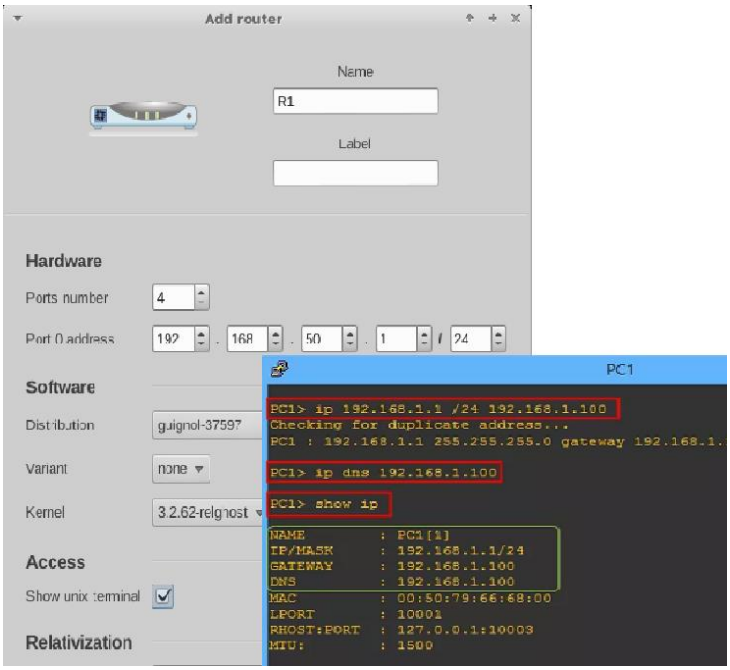

Figura 1. Configuración de direcciones IP.

2.4.3 RIP (Routing Information Protocol)

RIP (Malkin, 1998) es un protocolo de encaminamiento interno definido por la IEFT[19](*Internet Engineering Task Force*). Tiene una gran capacidad de interoperar con cualquier equipo de encaminamiento.

RIP es un protocolo poco sofisticado, comparado con otros protocolos de encaminamiento, pero su simplicidad y su amplia utilización hace que se siga utilizando.

Las principales características que tiene son:

[19] https://www.ieft.org/standards/rfcs/

- Protocolo de encaminamiento por vector distancia (intercambia periódicamente con los vecinos información de encaminamiento o alcanzabilidad).
- Utiliza como métrica el número de saltos; el camino mejor o más corto es el de menor número de saltos o enlaces en el camino.
- Número de saltos limitado para la selección de ruta, las rutas que tengan más de 15 saltos son rutas inalcanzables, lo que limita su utilización a redes de un diámetro máximo de 15 saltos.
- Se transmiten mensajes de actualización por defecto cada 30 segundos.

El funcionamiento es el siguiente: al iniciar la red o el protocolo, cada interfaz que está configurada con RIP manda un mensaje de solicitud a sus vecinos para que envíen sus tablas de encaminamiento completas (vectores de distancias) y cuando las recibe, evalúa los distintos caminos con el fin de quedarse con el más corto y actualiza así su tabla de encaminamiento. Después del proceso de inicialización, el intercambio de vectores de distancia se realiza periódicamente cada 30 segundos (sin necesidad de mensaje de solicitud).

Existen tres versiones de RIP, aunque la primera está obsoleta y se desaconseja su uso:

- RIP versión 1 o RIPv1, definida en RFC 1058.
- RIP versión 2 o RIPv2, definida en RFC 2053.
- RIPng, definida en RFC 2080, para IPv6.

En una de las prácticas de encaminamiento a implementar, se necesita que el simulador soporte las dos primeras versiones del protocolo, con el objetivo de poder comparar su funcionamiento.

RIPv1 solo funciona con el direccionamiento basado en clases o *classful* (direcciones de tipo A, B, y C), sin máscara de subred, y no admite la autentificación de mensajes de actualización.

La versión 2 de RIP es un protocolo sin clases (*classless*), admitiendo el uso de máscaras de subred y de *subnetting* de longitud variable (VSLM), CIDR y resumen de rutas. También tiene autentificación de mensajes y ayuda a confirmar que las actualizaciones vienen de fuentes autorizadas. Por último, admite el envío de mensajes de actualización por multidifusión (*multicast*), reduciendo el consumo de recursos frente a RIPv1 que realiza las actualizaciones por *broadcast*. (Muniz, 2016)

Comprobamos que no todos los simuladores de redes analizados son capaces de implementar las dos versiones del protocolo.

2.4.4 OSPF (Open Shortest Path First)

OSPF es un protocolo que surgió para solventar los problemas de rendimiento y escalabilidad de los protocolos de vector distancia. Es un protocolo de encaminamiento basado en Estado de Enlaces (*Link State),* donde cada router asigna un coste a sus enlaces. La información del estado del enlace se difunde a toda la red y a partir de esta información cada router es capaz de representar la topología completa de la red, calcular los caminos más cortos y construir su propia tabla de encaminamiento. El router es capaz de calcular el mejor camino mediante el uso del algoritmo de Dijkstra[20]. Usa el coste de cada enlace para el cálculo de la ruta más corta.

Las características más importantes de OSPF son: permite definir diferentes áreas de encaminamiento (encaminamiento jerárquico) y reducir así la sobrecarga de señalización. También permite realizar un balanceo de carga y la diferenciación por tipo de servicio.

La principal diferencia con el protocolo RIP, es que aquí se definen distintas áreas en la red y comprobamos que la convergencia del protocolo hacia rutas correctas y estables es más rápida, además de poder configurarlo para evitar la saturación a través del balanceo de carga. También es más escalable y puede trabajar con redes más grandes que RIP.

Se pretende implementar una práctica para analizar el protocolo OSPF y todas sus características. Este protocolo no es soportado por todos los simuladores de la lista, y para los que sí lo soportan, comprobamos que en algunos su configuración es más sencilla por su interfaz gráfica, mientras que en otros esta operación resulta más tediosa (línea de comandos).

2.4.5 VLAN

El diseño de VLAN (red de área local virtual o Virtual LAN) consiste en definir redes de área local en las que agrupamos los equipos de manera lógica independientemente de su arquitectura física. El objetivo es tener una o varias redes lógicas independientes, aunque estén dentro de la misma red física. En el diseño de VLANs es fundamental el concepto de enlace troncal (*trunk*), que permite trasportar por el mismo enlace físico, tráfico perteneciente a distintas VLANs.

Comprobamos que la mayoría de los simuladores soportan el diseño de VLAN, en algunos de una manera muy simple.

[20] Descripción del algoritmo. www.jariasf.wordpress.com (último acceso: 5 de abril de 2019)

Por tanto, en resumen, se puede decir que el simulador buscado tiene que, como mínimo, poder implementar direccionamiento IP, RIP (en sus dos versiones), OSPF y VLAN. Si alguno de estos objetivos no se puede realizar, el simulador es descartado. La Tabla 3 muestra un cuadro resumen de los simuladores y el soporte de estas funciones y protocolos.

OBJETIVO / SIMULADOR	SUBNETTING	RIPv1	RIPv2	OSPF	VLAN
GNS3	SI	SI	SI	SI	SI. Permite *trunk*
MARIONNET	SI	SI	SI	SI	SI. A nivel de puerto
NETSIM	SI	SI	SI	SI	SI. Permite *trunk*
OMNET++	SI	SI	SI	SI	SI. Tiene librerías para la implementación
RIVERBED MODELER	SI	SI	SI	SI	SI. Permite la segregación, *trunk* y limitación de estaciones
PACKET TRACER	SI	SI	SI	SI	SI. Permite Trunk
MININET	SI	SI	SI	SI	SI. Contiene un paquete con clases para realizar la simulación
CLOONIX	SI	SI	SI	SI	SI. Contiene un paquete con clases para realizar la simulación, pero no funciona correctamente.
CORE	SI	SI	SI	SI	SI. Necesario un dispositivo virtual
KIVA NS	SI	SI	SI	NO	NO
NETSIMK	SI	SI	SI	NO	SI. Permite *trunk*
JIMSIM	SI	SI	SI	SI	SI. Similar a la tecnología CISCO sobre VLAN
NS-2	SI	SI	SI	SI	SI. Ofrece una serie de ejemplos para realizar simulaciones de pruebas
CISCO VIRL	SI	SI	SI	SI	SI. Solo configurable para la capa L2

Tabla 3. Características principales.

2.5. FUNCIONALIDADES ESTADÍSTICAS

Otro aspecto importante es que sea posible comprobar y analizar los resultados obtenidos en las simulaciones de los ejercicios de las prácticas. Para ello, hay algunos simuladores que incluyen dentro del programa una utilidad para analizar y visualizar los resultados. En otros casos hay programas que son compatibles y capturan el tráfico que se genera en las simulaciones, permitiendo analizar el resultado de la simulación.

Estos programas o utilidades de los simuladores nos permiten ver el camino que toman los paquetes, comprobar el funcionamiento de los protocolos de encaminamiento, el funcionamiento de los enlaces (envío/respuestas de mensajes, saturación, reenvío, descartar), etc.

Wireshark es uno de esos programas, muy popular, de código abierto (gratuito), que permite analizar paquetes de datos que se transmiten en una red. Captura el tráfico generado y analiza los paquetes. Wireshark funciona con algunos de los simuladores estudiados, además es compatible con la mayoría de los sistemas operativos. Su funcionamiento es muy sencillo, y en algunos casos, lo puedes instalar en el mismo momento que la instalación del simulador, y en otros casos se puede ejecutar directamente desde el propio simulador.

En la Tabla 4 podemos comprobar que algunos de los simuladores estudiados tienen su propio analizador de tráfico y proporcionan estadísticas y gráficas de la simulación con los datos obtenidos en las capturas de tráfico, dando más opciones que con el uso de Wireshark.

Wireshark es el analizar más popular, y es el que se usa principalmente con GNS3. Nos muestra los paquetes que se transmiten por la red, y con la opción que tiene el simulador de gestión de enlace, podemos configurar los parámetros tales como la latencia y la pérdida de paquetes.

Algunos simuladores con licencia de software propietario incluyen dentro del programa su propio analizador, con estadísticas y graficas. También hay algunos que no nos van a ofrecer ninguna información del tráfico, mientras que otros solo nos dan datos estadísticos y no representaciones gráficas.

TRÁFICO / SIMULADOR	ANALIZADOR	ESTADÍSTICAS	GRÁFICOS
GNS3	Wireshark	Puede generar estadísticas con Wireshark	Tiene una opción de gráficos IO Graph de Wireshark (muestra el tráfico que pasa)
MARIONNET	Wireshark, ejecutado a través de línea de comando	Puede generar estadísticas con Wireshark	Tiene una opción de gráficos IO Graph de Wireshark (muestra el tráfico que pasa)
NETSIM	Tiene su propia interfaz gráfica para analizar el trafico	Tiene una pestaña en el programa donde puedes ver las estadísticas	-
OMNET++	Pcapdump: captura paquetes y los analiza con Wireshark	Tiene una clase que permite la recopilación de los datos del tráfico	Tiene una opción de gráficos IO Graph de Wireshark (muestra el tráfico que pasa)
RIVERBED MODELER	El simulador tiene un analizador incluido dentro del programa	El simulador puede calcular estadísticas de las simulaciones	El simulador muestra gráficos de los datos conseguidos
ACKET TRACER	Incluido en el simulador, se ve el envío de las tramas	Solo se puede ver las PDUs de la simulación	-
MININET	Wireshark, ejecutado a través de línea de comando	Puede generar estadísticas con Wireshark	Tiene una opción de gráficos IO Graph de Wireshark (muestra el tráfico que pasa)
CLOONIX	Wireshark	Puede generar estadísticas con Wireshark	Tiene una opción de gráficos IO Graph de Wireshark (muestra el tráfico que pasa)
CORE	Opción View en el simulador, hay que definir el tráfico	Mgen: genera el tráfico para estudio y estadísticas	-
KIVA NS	Solo puedes ver eventos de la simulación	Puedes ver los paquetes y luego calcular las estadísticas	-

TRÁFICO / SIMULADOR	ANALIZADOR	ESTADÍSTICAS	GRÁFICOS
NETSIMK	-	Solo distingue las PDUs	No
JIMSIM	-	-	-
NS-2	Xgrapf y Wireshark	Funciones estadísticas muy sencillas	Tiene una opción de gráficos IO Graph de Wireshark (muestra el tráfico que pasa)
CISCO VIRL	Tiene un módulo para analizar el tráfico	Tiene una clase que permite las estadísticas	-

Tabla 4. Analizadores.

2.6. OTRAS FUNCIONALIDADES AVANZADAS

En las simulaciones de redes se busca, además de un correcto funcionamiento del diseño, que ofrezca una calidad de servicio óptimo y sus costes sean los mínimos.

Hasta ahora hemos comprobado que casi todos los simuladores seleccionados cumplen las especificaciones requeridas; soporte de direccionamiento IP, configuración de los protocolos de encaminamiento RIP y OSPF y configuración de redes VLAN.

En la búsqueda de la eficiencia y la eficacia de las redes, algunos de los programas también permiten simular servicios relacionados con la QoS (*Quality of Service*). Los servicios o técnicas que están más relacionados con la asignatura son:

- MPLS *(MultiProtocol Label Switching)*[21] (Rosen, Viswanathan, & Callon, 2001) - intenta solucionar el problema entre establecer rutas predeterminadas y que sean altamente eficientes. Con MPLS, la primera vez que un paquete entra en la red se le asigna una etiqueta, lo cual es un identificador corto de longitud fija que identifica una clase de equivalencia de reenvío específica (FEC - *Forwarding Equivalence Class*). Cada router tiene una tabla que indica cómo manejar los

[21] RFC 3031, https://tools.ietf.org/html/rfc3031 (último acceso: 5 de abril de 2019)

paquetes de un tipo específico de FEC. Esto da a la red MPLS la capacidad de manejar paquetes con características particulares de manera consistente. Se consigue que paquetes que transportan tráfico en tiempo real, se les pueda asignar fácilmente a rutas de baja latencia en toda la red. Lo importante del uso de MPLS es la etiqueta que se le asigna e incluye la información adicional de cada paquete. La ventaja de usarlo es: escalabilidad, rendimiento y un mejor uso del ancho de banda; como desventajas tenemos que es costoso y que no todos los dispositivos lo implementan. (Wienberg & Johnson, 2018)

- IntServ (*Integrated Services*) – fue el primer intento en ofrecer garantías QoS a las redes IP (RFC 1633). Se basa en una arquitectura de reserva de recursos con el fin de satisfacer los requisitos de las aplicaciones en tiempo real. Para poder trabajar adecuadamente necesita un protocolo de señalización.

  RSVP (*Resource ReSerVation Protocool*)[22] (Braden, Zhang, Berson, Herzog, & Jamin, 1997)- es un protocolo de control de red que permite que las aplicaciones puedan obtener la calidad de servicio que sus flujos de datos puedan requerir, mediante la reserva de recursos. Está dentro de la capa de transporte y se apoya en la tabla de encaminamiento. Su funcionamiento consiste en la creación de una sesión donde se especifica la dirección del destino, un identificador y un puerto; cuando se activa la sesión, el router recibe un mensaje RSVP donde se especifica la reserva de recursos que se ha de realizar a lo largo de todo el camino; no todos los router tiene la capacidad de usar RSVP. El propio protocolo tiene capacidad de negociación de la Calidad de Servicio requerida. (Danysoft, 2016)

- DiffServ (*Differentiated Services*)[23] (Nichols, Blake, Baker, & Black, 1998) - está relacionado también con la calidad de servicio de las redes, principalmente en las de gran tamaño.

  Consiste en marcar paquetes IP mediante un código llamado DSCP (*Differentiated Services Code Point*) utilizado en la cabecera IP. Los routers y switches de la red pueden leer el campo y priorizan el tráfico indicado mediante técnicas de encolado del tráfico. (Rejón, 2016)

Dentro de los simuladores bajo estudio, tenemos algunos de ellos que ofrecen características de QoS. Algunos lo hacen de una manera muy sencilla y otros no tienen la capacidad de realizarlos completamente. La información se muestra resumida en el cuadro de la Tabla 5.

[22] RFC 2205, https://tools.ietf.org/html/rfc2205 (último acceso: 13 de abril de 2019)
[23] RFC 2474, https://tools.ietf.org/html/rfc2474 (último acceso: 13 de abril de 2019)

QoS / SIMULADOR	MPLS	IntServ + RSVP	DiffServ
GNS3	Permite incluirlo en la configuración de los protocolos de encaminamiento	Define el ancho de banda	Sí
MARIONNET	No hay información disponible actualmente	No hay información disponible actualmente	No hay información disponible actualmente
NETSIM	Varias versiones de implementación	No hay información disponible actualmente	No
OMNET++	Tiene un módulo de implementación	Incluye un modelo	Implementa una clase que implementa el servicio
RIVERBED MODELER	En versiones superiores a la 1.1	Se usa para comparativas	Se usa para el control de tráfico
PACKET TRACER	No hay información disponible actualmente	Se podría configurar manualmente en algún router	Implementa DSCP, pero configurado de manera manual
MININET	No hay información disponible actualmente	Lo implementa. Pero su uso es fundamentalmente es para el estudio del tráfico.	Se puede implementar
CLOONIX	No hay información disponible actualmente	Mediante una configuración manual	No hay información disponible actualmente
CORE	No hay información disponible actualmente	No hay información disponible actualmente	No hay información disponible actualmente
KIVA NS	No hay información disponible actualmente	No	No

QoS / SIMULADOR	MPLS	IntServ + RSVP	DiffServ
NETSIMK	No hay información disponible actualmente	No hay información disponible actualmente	No hay información disponible actualmente
JIMSIM	No hay información disponible actualmente	No hay información disponible actualmente	No
NS-2	Con un paquete software	Con un paquete software	Módulo que implementa el servicio
CISCO VIRL	Sí	No hay información disponible actualmente	A través de circuitos virtuales

Tabla 5. Otras características.

Podemos comprobar que la mayoría de los simuladores de licencia propietaria ofrecen los servicios de QoS, y dentro de los de código abierto podemos destacar GNS3. Pero no es una información que esté muy disponible.

Otras características destacables, que se han visto durante el análisis de los simuladores, son:

- Máquinas virtuales: su uso hace que el rendimiento sea mayor, al no tener que instalar todos los elementos en el ordenador.
- Simuladores orientados a titulaciones o estudios: algunos de ellos sirven principalmente para conseguir titulaciones o certificaciones específicas de fabricantes, esto hace que su uso no sea el idóneo para nuestras especificaciones.
- Implementación de otros protocolos en las diferentes capas:
  - Capa de enlace- Ethernet, Frame Relay, ATM.
  - Capa de transporte- intercambio de datos entre sistemas finales.
  - Capa de aplicación- HTTP, Telnet…
- Posibilidad de la encriptación de mensajes.

- Implementación de otras redes, como son: las redes móviles ad-hoc, redes *Wireless*.

Estas características, técnicas o protocolos que ofrecen los simuladores de redes; algunos ofrecen todas y otros solo algunas, son aspectos a tener en cuenta a la hora de elegir el simulador más adecuado. Es importante que el simulador soporte todos los objetivos de la asignatura; mientras que el resto de las características es bueno que el programa las soporte para que el mismo sea utilizable para objetivos futuros de ésta u otras asignaturas.

Con toda esta información, ya se tienen los datos suficientes para hacer una valoración de cada simulador, y comparar todas sus características.

2.7. ANALISIS COMPARATIVO DE LOS SIMULADORES

2.7.1 DESCRIPCIÓN

El objetivo de este apartado es disponer de una perspectiva comparativa de todos los simuladores estudiados, que permita decidir cuál de ellos debería ser seleccionado para su uso en la asignatura de "Laboratorio de Diseño y Configuración de Redes".

Los criterios que se han empleado para hacer el análisis comparativo son los siguientes:

- Orientación del simulador - comercial, académico, o de investigación.
- Tipo de licencia- libre o propietaria.
- Sistema Operativo- si solo trabaja con uno o con varios.
- Curva de aprendizaje y uso del simulador.
- Tráfico- generación, calidad.
- Mantenimiento, confiabilidad y escalabilidad.
- Actualizaciones.

2.7.2 ORIENTACIÓN DEL SIMULADOR

Los desarrolladores de programas, en general y cada día más, se van ajustando a las necesidades que les pide la sociedad. Es más, en muchos casos se programa bajo demanda, con las especificaciones que el cliente necesita.

Algunos de los simuladores estudiados tienen distintas versiones, unas orientadas hacia la investigación y la educación, y otras con una visión comercial. En algunos casos, la versión educativa no está completa y no ofrece todos los servicios que da la versión de pago. También hay simuladores que tienen versiones de evaluación, con una duración determinada de uso, de forma que al expirar el plazo no se puede seguir trabajando con el programa, salvo que adquieras la versión comercial.

En relación con los simuladores estudiados, algunos como Riverbed Modeler, NETSIM, NS-2 y Omnet++ nos ofrecen una versión de pruebas con una duración determinada. Packet Tracer está especialmente orientado al ámbito educativo, ofreciendo incluso un curso gratuito con prácticas propias. Por otra parte, CISCO VIRL está dirigido

principalmente a la obtención de la certificación CCNA[24]. Core es un simulador desarrollado por Boeing, para la investigación y desarrollo de Boeing con la ayuda del Laboratorio de Investigación Naval. El resto no tienen una orientación específica, su uso es general.

2.7.3 TIPO DE LICENCIA

Existen dos tipos de licencias principalmente:

- Software propietario.
- Software libre.

El software propietario lo podemos definir como software no libre, a cuyo código fuente el usuario no puede acceder, teniendo un uso limitado, sin posibilidad de modificar ni redistribuir. Para acceder a dichos programas, el usuario tiene que pagar por una licencia, ya que las compañías poseen los derechos de autor sobre su código.

En oposición al software propietario, está el software libre, que se está popularizando, y que permite que cualquiera con conocimientos lo pueda modificar y redistribuir. Actualmente estos softwares están siendo desarrollados por pequeñas empresas o grupos de usuarios, cuya participación activa contribuye a que el sistema mejore antes y con mayor confiabilidad.

En este caso, se prefiere un software que no genere un gasto, y sea independiente de las decisiones tomadas por las empresas propietarias, siendo, por tanto, la mejor opción el uso de **software libre**. Además, estos programas son más accesibles para los alumnos. Algunos de los simuladores estudiados tienen licencia de propietario, pero se puede conseguir una versión de prueba o limitada en tiempo, como Riverved Modeler, Packet Tracer, Cisco Virl... En la versión libre tenemos, por ejemplo, GNS3 y Marionnet. Existen algunos que son mixtos: tienen una versión libre (normalmente orientada hacia la educación y con menores prestaciones) y una versión de pago, como NetSim, Riverved Modeler.

2.7.4 SISTEMA OPERATIVO

Los simuladores de redes estudiados trabajan en los siguientes sistemas operativos:

- Windows- en las diversas versiones que existen en el mercado.

[24] CCNA- Cisco Certified Network Associate.

- GNU/Linux- y sus versiones.
- Mac.

La accesibilidad al programa es fundamental, ya que se busca que el alumno, además de poder realizar las prácticas en el laboratorio de la asignatura, puede afianzar sus conocimientos fuera del mismo. Por ello, el sistema operativo Windows es el más adecuado, ya que la mayoría de la población lo tiene instalado en sus ordenadores y soporta la mayoría de los simuladores estudiados. En la actualidad, el uso del sistema Linux está aumentando, pero es más complicado, ya que requiere la realización de una partición del disco duro o la instalación de una máquina virtual donde correría el sistema operativo. La opción de un simulador que funcione solo en el sistema Mac está completamente descartada, ya que el elevado coste que tienen los equipos de esta marca hace que no sea viable.

Entre los simuladores de la lista, hay varios que funcionan en distintos sistemas operativos, ofreciendo versiones para cada uno de ellos, como por ejemplo GNS3 y OMNet++.

2.7.5 CURVA DE APRENDIZAJE Y USO DEL SIMULADOR

El uso del simulador por parte de los alumnos va a ser progresivo. La metodología que se sigue es tener una o dos sesiones de laboratorio para familiarizarse con el programa y luego se continúa aplicando los conocimientos teóricos dados en la asignatura a través de ejemplos prácticos.

Una de las características que buscamos en el nuevo simulador es que su funcionamiento sea sencillo y fácil de aprender a usarlo. Por ejemplo, Riverbed Modeler es un programa con un entorno gráfico sencillo y cómodo; su interfaz gráfica hace que sea muy intuitiva, encontrando todas las funciones necesarias con rapidez; en contra, Marionnet nos obliga a tener conocimientos de Linux y a realizar las configuraciones y comprobaciones mediante línea de comando.

Algunos de los simuladores, como es el caso de Kiva Ns, aunque su uso es fácil y su aprendizaje es rápido, solo permiten simular una pequeña parte de los conceptos que se imparten en la asignatura.

En general los simuladores de licencia propietaria tienen una interfaz gráfica mucho más potente que los de software libre. En este grupo podemos destacar Riverbed Modeler y Packet Tracer, que nos ofrecen una serie de pestañas con apartados para las configuraciones. Dentro de los programas de software libre tenemos GNS3, con una apariencia gráfica similar a la de Packet Tracer, aunque su número de pestañas es inferior. Generalmente la apariencia es pareja en todos los simuladores, tienen una

parte central donde realizar el diseño, y en los laterales las funciones que se pueden hacer, y los elementos que puedes usar.

Algunos simuladores ofrecen la posibilidad de configuración de manera manual, indicando específicamente cada dirección, máscara, pasarela, etc. GNS3 y Marionnet son dos de esos simuladores que permiten esta configuración manual, que, aunque es más laboriosa, porque en la mayoría de los casos se realiza mediante consola, puede ayudar al alumno a aumentar sus conocimientos sobre redes.

2.7.6 GENERACIÓN DE TRÁFICO

En la actualidad, la sociedad quiere que sus redes de comunicación sean más eficaces y eficientes, el tráfico que circula por ellas ha aumentado y se busca que este aumento de carga de tráfico sea gestionado de una manera óptima. Un problema que puede aparecer y que se tiene que controlar es el de la congestión de la red, lo que puede originar una pérdida de datos o un retardo de estos.

La solución para la gestión del tráfico hay que plantearla desde el diseño. En la fase de diseño tenemos que hacer una previsión de cuánto tráfico debe ser capaz de gestionar la red, y cuando elijamos los equipos debemos seleccionar aquellos que cumplan las necesidades establecidas. Con los simuladores podemos prever todos estos problemas y atajarlos antes de la implantación real de la misma. Nos facilitan la elección de equipos de interconexión que sean capaces de trabajar con una gran cantidad de datos, así como la de servicios o protocolos que permitan negociar, clasificar y conformar el tráfico entre enlaces, tanto punto-a-punto como extremo-a-extremo.

Algunos simuladores incluyen tecnologías que permiten controlar y priorizar el tráfico, como RSVP y MPLS. Gns3, Riverbed Modeler y Omnet++ están entre ellos.

Muchos también incluyen otras características de filtrado de tráfico, como el control mediante listas de acceso (ACLS), para permitir o denegar algún tipo de tráfico. Packet Tracer permite esta opción.

Algunos simuladores como GNS3 y Riverbed Modeler permiten configurar el valor del *jitter* en los enlaces (introducir retardo), pérdida de paquetes… Para comprobar estas características que introducimos en la configuración de los enlaces de las redes usamos los analizadores de tráfico. Los analizadores nos permiten analizar el contenido de los paquetes que circulan por la red, así como el tiempo que transcurre entre su salida y la llegada al destino. Algunos con opciones más avanzadas ofrecen representaciones gráficas del tráfico que circula por los enlaces.

2.7.7 MANTENIMIENTO, CONFIABILIDAD, ESCALABILIDAD

El mantenimiento de los programas es una característica importante. El funcionamiento correcto de los simuladores puede ahorrar tiempo y dinero. Los desarrolladores deben estar renovando los equipos de simulación según aparezcan en el mercado, así como los sistemas de transmisión. El mantenimiento en software libre es muy interactivo; el usuario puede enviar al desarrollador el problema que ha visto y en ocasiones hasta le puede enviar la solución para que la aplique.

Entre los simuladores estudiados hay algunos que ya tienen una opción de nube (*Cloud*), la cual permite almacenar las configuraciones y los datos conseguidos, de una manera accesible.

Hay versiones o parches en la red, que consiguen una mayor escalabilidad del programa aumentando las topologías que se pueden implementar, así como servicios que antes no daban. Omnet++ es un simulador que está basado en clases; si el usuario instala el paquete completo de clases obtendrá todos los servicios posibles que ofrece el simulador; dependiendo del número de clases que se instale los servicios que ofrezca serán distintos.

2.7.8 ACTUALIZACIONES

Usar una versión obsoleta de un simulador, va a suponer que no tengamos un resultado óptimo del estudio que estamos realizando. Las compañías deben actualizar sus programas con los nuevos avances que se produzcan en ellos.

Algunos programas crean actualizaciones que solo afectan a una parte del programa, por ejemplo, actualizaciones de protocolos, pudiendo encontrar una gran variedad de actualizaciones de algunos simuladores. Omnet++, al estar basado en clases, nos ofrece las actualizaciones mediante una amplia gama de clases, que puedes instalar según sea necesario.

Las actualizaciones, en algunos casos, las tiene que buscar el usuario e instalarlas; mientras que, en otros, el programa nos indica que hay una nueva versión disponible; GNS3 es un ejemplo de estos últimos.

En los simuladores de licencia de propietario las actualizaciones o versiones mejoradas del mismo suelen ser de pago. Por ejemplo, Packet Tracer tiene una versión básica gratis, pero para utilizar su versión completa hay que pagar la licencia de uso. En el listado de simuladores que estamos comparando, algunos de ellos tienen actualizaciones frecuentes, entre ellos GNS3, Marrionet, Mininet, pero tenemos otros como Kiva Ns y JimSim que se han quedado obsoletos y no cumplen los requisitos que se buscan.

2.8. SELECCIÓN DEL SIMULADOR

Teniendo en cuenta todo lo expuesto en los apartados anteriores, ya podemos tomar una decisión a la hora de elegir el simulador que mejor se ajusta a nuestras necesidades. Todos los simuladores estudiados (GNS3, Omnet++, Cisco Virl, …) son programas que tienen pros y contras para su elección.

Durante todo este capítulo, se ha expuesto toda la información relacionada con los simuladores estudiados, explicando sus características, así como los requisitos necesarios para poder implementar las prácticas de la asignatura. En este apartado se va a realizar una breve comparación de los simuladores, teniendo en cuenta dichos requisitos, llegando finalmente a la elección del más adecuado para los objetivos de la asignatura.

En primer lugar, hay que tener en cuenta hacia dónde va orientado su uso; si su utilización es educativa o profesional, algunos de ellos sirven para obtener una certificación como Cisco Virl o Netsim.

Un punto importante y la base de la realización de este proyecto, es encontrar un simulador que no fuera de licencia propietaria (lo que nos haría descartar Riverbed Modeler y Cisco Virl, entre otros), es decir, buscamos software libre (como GNS3 y Cloonix) para evitar que la universidad tenga que pagar una licencia y estar dependiendo de un software comercial y su futuro incierto. En el estudio se han analizado los dos tipos de simuladores, propietario y libre, así como algunos que nos dan una versión libre con menos funcionalidades que la versión de pago, como Netsim y Omnet++. Al final se ha llegado a la conclusión de que sea uno de software libre, ya que también cumplen con los requisitos especificados.

Dentro de la opción de software libre, hay que elegir el sistema operativo con el que se quiere trabajar. La principal opción es el uso de Windows como sistema operativo, ya que este sistema es el más usado entre los alumnos de la asignatura. Además, queremos un simulador que no tenga grandes requisitos técnicos y se pueda instalar fácilmente en cualquier tipo de ordenador. En los laboratorios de la universidad no hay ningún problema a la hora de la elección de sistema operativo, ya que tienen instalados tanto Windows como Linux.

El programa tiene que ser capaz de implementar todos los conceptos que se imparten en la asignatura, sin ser necesario reproducir las prácticas de forma idéntica, porque se podrían adecuar a las características del simulador. Esto nos lleva a descartar los simuladores que no son capaces de reproducir todas las prácticas (como Kiva, NS y Netsik). También es deseable que sea fácil de usar, que tenga una interfaz gráfica intuitiva, y que no se pierda mucho tiempo en su aprendizaje. Como una opción a mayores, la mayoría de los programas libres ofrecen la configuración de los elementos

de la red a través de línea de comandos, que ayudaría a complementar los conocimientos de los alumnos.

El análisis de tráfico y de resultados es fundamental; permite comprobar el alcance de las simulaciones, que el diseño es correcto, que no hay pérdida de datos... Algunos de los simuladores tienen esta función dentro del programa (Riverbed Modeler) y otros como GNS3, Marrionet, Mininet y OMNet++, trabajan con Wireshark, que es un programa compatible con la mayoría de ellos. Este programa de captura de tráfico también es de software libre, por lo que no hay ningún problema en obtenerlo; su punto débil es que no ofrece funciones de estadística y representación gráfica que, aunque no son imprescindibles para la asignatura objeto de este estudio, sí que serían útiles para otras asignaturas.

En el sector de las telecomunicaciones, los cambios en protocolos, equipos y técnicas se producen en espacios de tiempos relativamente cortos; por eso el simulador debe tener actualizaciones casi a la misma velocidad que se producen los cambios. GNS3 nos ofrece este servicio de actualización, avisando con un mensaje emergente al abrir el simulador. Las actualizaciones las podemos conseguir desde la página oficial de GNS3.

Por último, como requisitos adicionales a los necesarios para la asignatura, se han incluido tecnologías como MPLS, RSVP, DiffServ y, en general, técnicas de QoS. Su disponibilidad hace que la valoración del simulador aumente, aún sin ser un punto fundamental a la hora de la elección.

Después de estudiar, analizar y comprobar todos los puntos expuestos durante este capítulo, los programas que cumplían más objetivos eran Riverbed Modeler, GNS3, OMNet++, Marionnet y Nestim. Cada uno de ellos tiene puntos positivos y puntos negativos. En nuestro caso, Riverbed Modeler y Netsim los descartamos por tener que pagar por sus licencias, así como OMNet++, ya que, aunque su versión académica está bastante bien, es preferible evitar la dependencia comercial. Entre los otros dos que quedan al final, se ha descartado Marionnet por tener que trabajar en Linux. Por tanto, el simulador elegido finalmente es **GNS3**.

GNS3 cumple casi todos los objetivos que nos marcaron al inicio del proyecto; la descarga desde su web (www.gns3.com) es fácil , es capaz de trabajar en distintos sistemas operativos, su interfaz gráfica es muy intuitiva y se pueden implementar las prácticas de la asignatura adecuándolas al simulador. Permite la configuración de equipos a través de la línea de comandos, y es compatible con Wireshark. Por último, nos permite trabajar con técnicas y servicios relacionados con la calidad, gestión y eficiencia de las redes. Este simulador no es perfecto, como veremos, y va a costar un tiempo acostumbrarse a él, después de trabajar con Riverbed Modeler, pero es una buena solución de compromiso para el problema abordado al inicio del proyecto.

Capítulo 3 – GNS3

3.1. DESCRIPCION

GNS3 es el simulador elegido para intentar sustituir a Riberved Modeler en la asignatura "Laboratorio de Diseño y Configuración de Redes", ya que nos va a permitir realizar el diseño de diferentes topologías de red y hacer las simulaciones correspondientes a las prácticas de la asignatura.

Figura 2. GNS3

GNS3 es utilizado por ingenieros, estudiantes e investigadores para la emulación, configuración y la gestión de problemas que puedan ocasionar en sus diseños de redes reales o virtuales. Las topologías que se pueden implementar van desde pequeñas redes en las que intervienen unos pocos elementos, hasta grandes topologías de red con servidores y dispositivos alojados en la Cloud.

GNS3 lo podemos descargar desde su propia página web: http://gns3.com o desde alguna página que nos ofrece un enlace para su descarga como: http://telectronika.com, http://sourceforge.net.

Es un programa de software libre, desarrollado por una comunidad compuesta por ingenieros, arquitectos, estudiantes, empresas que participan en su mejora con cambios y actualizaciones. Su uso es muy amplio, lo mismo lo utiliza un estudiante para obtener la certificación Cisco CCNA o lo usa una empresa para la mejora de un proyecto con el fin de reducir sus costes.

Ha ido evolucionando con los años, al principio su uso estaba orientado hacia dispositivos Cisco, pero ahora también es capaz de implementar dispositivos de otros proveedores.

3.1.1 VENTAJAS E INCOVENIENTES

Dentro de las ventajas que nos ofrece GNS3 podemos destacar las siguientes:

- Es un software libre y de código abierto, la comunidad de GNS3 interviene en sus cambios.
- No hay cuotas, los usuarios no tienen que abonar por su uso.
- Es compatible con los sistemas operativos Linux y Windows, la única limitación son las características de los dispositivos donde se instalen.
- Admite múltiples opciones de conmutación (elementos de la capa 2).
- Trabaja de manera local o de manera virtual, el uso de máquinas virtual es posible (VirtualBox, VMware Workstation).

Por otra parte, entre sus principales inconvenientes cabe destacar:

- Las imágenes de Cisco deben ser suministradas por el usuario; existen distintas páginas donde descargar las imágenes o desde el Marketplace que tiene el propio simulador.
- El paquete no es independiente, requiere una instalación local en el ordenador.
- Hay que tener cuidado a la hora de la instalación, porque puede verse afectada por la configuración del firewall, antivirus o políticas de distintos programas.

Asimismo, podemos destacar como características adicionales de GNS3:

- Simulaciones en tiempo real, sin necesidad de tener un hardware de red.

- Es capaz de trabajar con equipos de distintos proveedores sin ningún problema.
- Creación de mapas dinámicos antes de construir las redes con el fin de reducir el tiempo de toma de decisiones y poder ponerlo en producción lo antes posible.
- Topología y laboratorios personalizados para la certificación de redes, siendo una herramienta recomendable para aspirantes que busquen su certificación.
- Permite tener un laboratorio de pruebas en casa, y permite conectar GNS3 a cualquier red real directamente.

3.2. INSTALACIÓN DE GNS3

La instalación del simulador GNS3 es muy sencilla. En la red puedes encontrar un gran número de páginas donde te explica como instalarlo[25], pero desde la web de GNS3 también se te ofrece este servicio. El primer paso es que el ordenador donde lo vayamos a instalar cumpla los requisitos de sistema operativo:

- Windows desde la versión 7 (64 bits) en adelante, Windows Server desde la versión de 2012 (64 bits) y sucesivas.
- Mac OS X Mavericks desde la versión 10.9.
- Linux.

En lo relacionado con el hardware, el procesador debe tener al menos 2 núcleos y ser capaz de realizar virtualizaciones y extensiones de virtualizaciones. Pero lo más importante es contar con un mínimo de 4 GB de memoria RAM, almacenamiento de disco duro de al menos 1 GB y un almacenamiento adicional para la instalación de las imágenes de los equipos. Cumpliendo con estos requisitos la instalación se realizará correctamente.

Es importante definir el método de trabajo que se va a usar antes de la instalación. GNS3 nos permite trabajar en un *modo local* o a través de *máquina virtual.* El aprendizaje y la instalación es más sencilla en modo local, ya que, para realizar configuraciones básicas, solo haría falta routers IOS de Cisco y una instalación local (Dynamips). El uso de la máquina virtual se recomienda cuando el usuario ya ha trabajado previamente con el simulador; y permite mejorar el rendimiento, y la optimización y ofrece un mayor aprovechamiento del programa GNS3.

[25] Descarga: https://telectronika.com/descargas/gns3/ (último acceso: 26 de octubre de 2019)
https://sourceforge.net/projects/gns-3/ (último acceso: 26 de octubre de 2019)

La instalación comienza con la descarga del programa desde la web de GNS3: *https://gns3.com/*. Es necesario que el usuario se registre en la comunidad GNS3, donde estará informado de las actualizaciones y cambios que se hagan en el programa y de los servicios que ofrece. A continuación, hay que seleccionar el sistema operativo, y aceptar las condiciones de la licencia (ver Figura 3). Después de estos pasos descargamos el paquete GNS3.

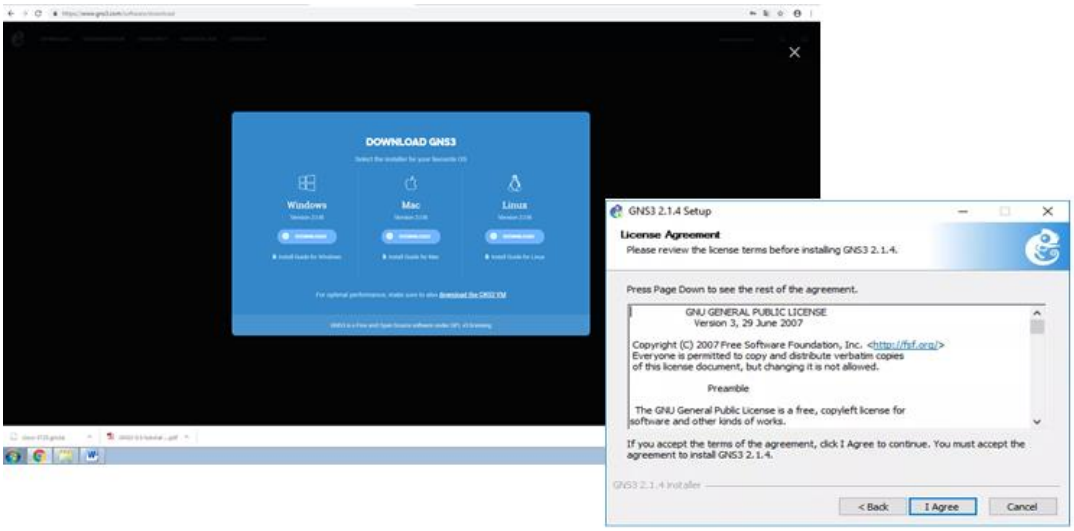

Figura 3. Términos de licencia GNS3.

El paquete tiene una serie de componentes que son necesarios e indispensables para la instalación, y otros que los podemos sustituir con algún otro programa que tengamos instalado o son opcionales. La Tabla 6 indica todos los elementos o componentes que son requeridos, los recomendados y los opcionales, para la instalación de GNS3, así como una descripción de la función de cada uno de estos paquetes.

La ubicación de la instalación del programa es determinante. De manera predeterminada la instalación se ubica en *C:/Program Files/GNS3.* Para completar la instalación continuamos pulsando "siguiente" en las diferentes pantallas que nos van apareciendo hasta llegar al final de la instalación. Si el sistema operativo no es Windows:

- **Linux**: desde la propia página GNS3 hay un enlace de descarga del programa y se incluye un documento donde se explican todos pasos que tienes que seguir para poder usarlo[26].
- **MacOSX**: como en los anteriores sistemas, desde la página de GNS3 se puede descargar, y también hay un documento con la explicación de instalación[27].

[26] https://docs.gns3.com/1QXVIihk7dsOL7Xr7Bmz4zRzTsJ02wklfImGuHwTlaA4/index.html (último acceso: 16 mayo de 2019)

[27]https://docs.gns3.com/1MlG-VjkfQVEDVwGMxE3sJ15eU2KTDsktnZZH8HSR-IQ/index.html (último acceso: 16 mayo de 2019)

REQUISITO / COMPONENTE	NECESARIO PARA LA INSTALACIÓN	DESCRIPCIÓN
WINCAP	Requerido	Necesario para conectar Gns3 a la red. Permite la comunicación de los proyectos con el exterior
NPCAP	Opcional	Reemplaza a Winpcap, mejora sus capacidades
WIRESHARK	Recomendado	Permite la captura de tráfico entre nodos
DYNAMIPS	Requerido	Necesario para la instalación local de Gns3. No es necesario si se usa máquina virtual
QEMU	Opcional	Emulador de computadora usado para Linux
VPCS	Recomendado	Emulador de computadora que admite comandos básicos
CPULIMIT	Opcional	Evita que Qemu use 100% de la CPU
GNS3	Requerido	Núcleo. Necesario para la instalación
TIGHTVNC VIEWER	Recomendado	Cliente VNC para conectarse a las interfaces gráficas de usuario del dispositivo
SOLAR-PUTTY	Recomendado	Consola de aplicación por defecto
VIRT-VIEWER	Recomendado	Visualizador de máquinas virtuales

Tabla 6. Elementos del paquete GNS3. (telectronika.com, s.f.)

3.2.1 MÁQUINA VIRTUAL

Si la opción elegida de uso del simulador es mediante el uso de la máquina virtual, en la página oficial de GNS3 dispone de un manual de instalación y enlace para la descarga de las máquinas (Coleman & Duponchelle, 2019).

El uso de máquinas virtuales permite conseguir una mayor velocidad en las simulaciones y es compatible con la virtualización anidada. GNS3 trabaja normalmente con las

siguientes máquinas virtuales: VMware WorkStation Pro, VMware Fusion y VMware Player (algunas son de licencia libre y otras no).

El asistente nos permite trabajar con una configuración de máquina virtual (VM) local, en la Figura 4 vemos la ventana donde nos dan las opciones; en la configuración de máquina virtual elegimos la *Run modern IOS.*

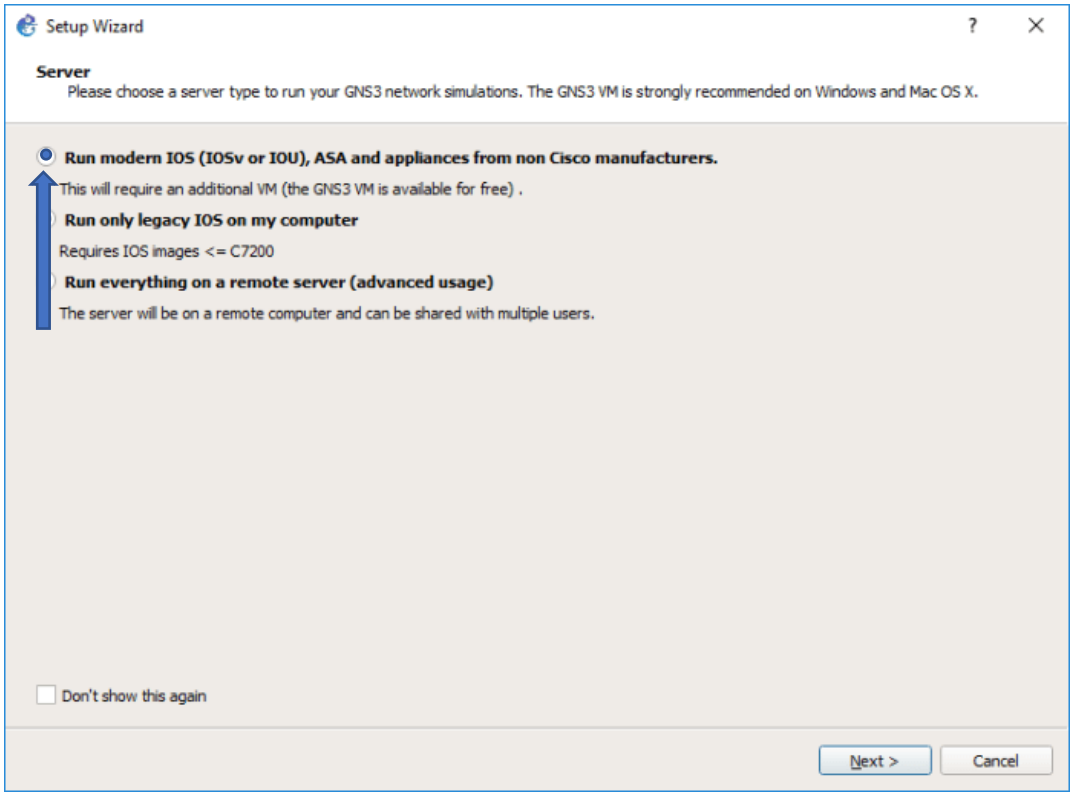

Figura 4. VM Local en GNS3.

En la configuración de los ajustes del servidor local, hay que definir la dirección de *loopback* 127.0.0.1 y el puerto que se va a usar. Si no está bien configurado, aparecerá una excepción en la pantalla. Después de la configuración, es recomendable actualizar para comprobar que la importación de la máquina virtual se ha realizado correctamente. Al arrancar la máquina virtual, nos mostrará la dirección de la máquina que vamos a usar en las simulaciones, y tenemos que definir los dispositivos virtuales y el tipo de servidor (local o virtual).

En la Figura 5, podemos ver todas las opciones de dispositivos que podemos instalar, desde *IOS router* a un *hub genérico*. También hay que indicar si vamos a trabajar de manera local en nuestro ordenador o con una máquina virtual.

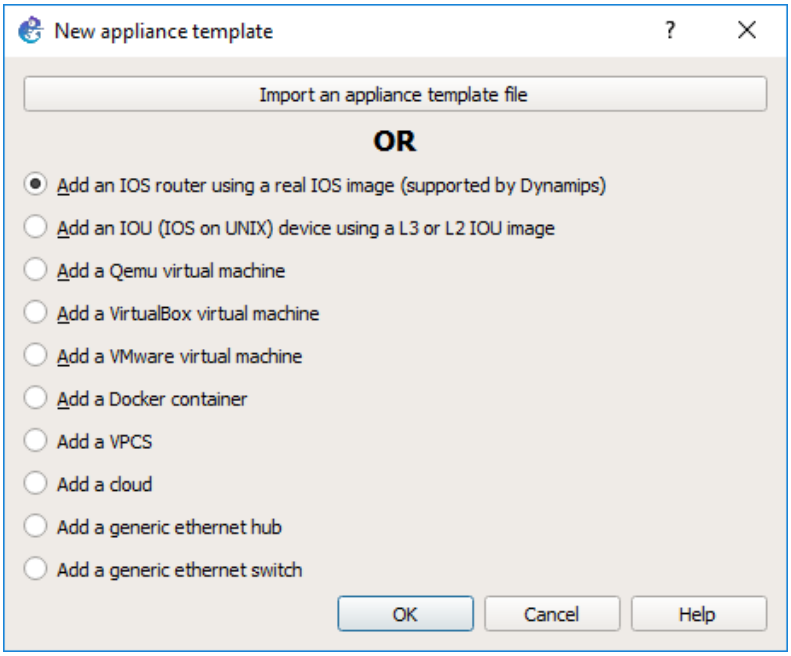

Figura 5. Añadiendo dispositivos en GNS3.

Las imágenes de los dispositivos las deberemos tener todas en una carpeta; cuando descargas dichas imágenes todas están comprimidas, por lo que las debemos descomprimir antes de usarlas. Cuando instalamos un nuevo dispositivo, como se muestra en la Figura 6, tenemos que indicarle el nombre con el que lo vamos a usar y la plataforma.

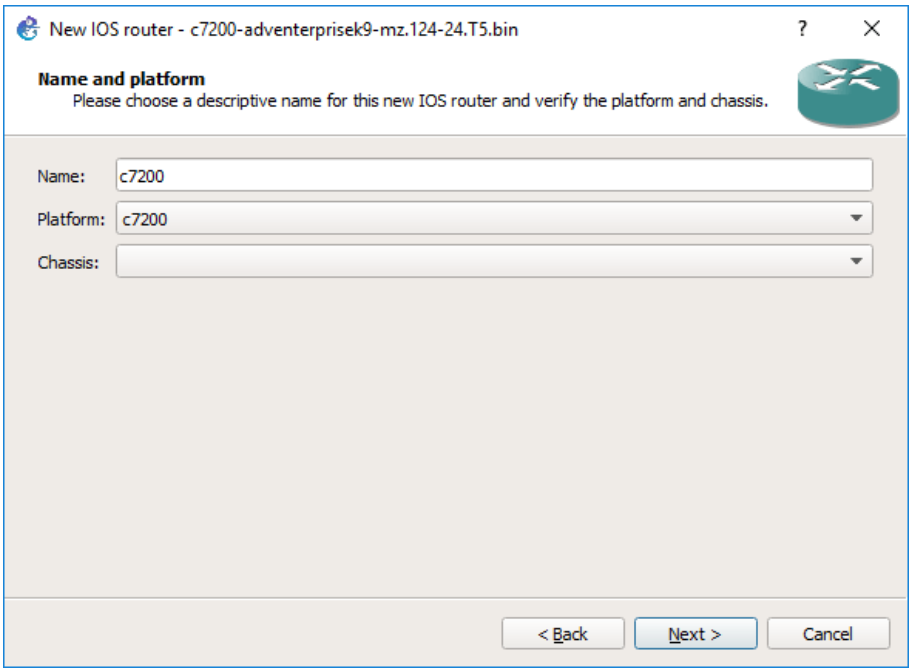

Figura 6. Añadiendo un router en GNS3.

Para finalizar, debemos determinar los parámetros de uso del dispositivo, como cuánta RAM necesita, los adaptadores de red WIC o el valor de IDLE-PC para el rendimiento óptimo. Con el servidor local configurado, si vamos a la opción *Preferences* dentro del menú *Dynamips,* comprobamos la configuración; y ya solo queda la creación del *Nuevo Proyecto*, pero esta parte se explicará en el siguiente apartado.

3.2.2 CONFIGURACIÓN LOCAL

La opción de *Run only legacy IOS on my computer,* indica que vamos a trabajar de manera local sin máquina virtual; la Figura 7 es igual que la Figura 4, pero en este caso debemos cambiar la elección para configuración local.

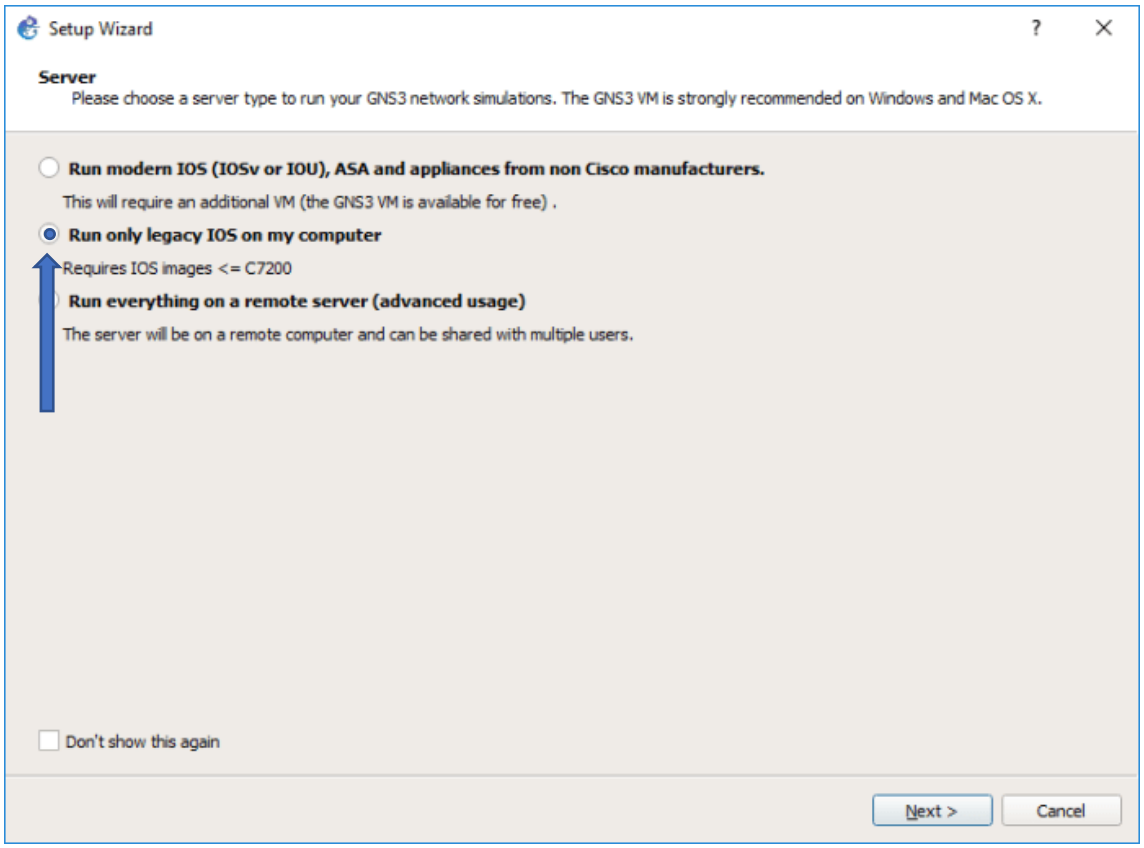

Figura 7. Servidor local en GNS3.

Los primeros pasos son iguales a los que hay que realizar con la máquina virtual, así como la definición de características y rendimiento de los dispositivos. Al finalizar toda esta configuración ya se puede comenzar a realizar simulaciones creando un *Nuevo Proyecto.*

Después de configurar la manera de trabajar (máquina virtual o local), es importante la configuración de las preferencias. Lo haremos una vez, y no será necesario cambiarlo en

las siguientes simulaciones. En el menú de *Preferencias* definimos los aspectos generales del programa, idioma, comando del terminal (telnet…), directorio de almacenamiento, ajustes gráficos…. Un submenú de *Preferencias* es *Dynamips,* que emula las plataformas de los routers Cisco y las serie que tiene, y ejecuta las imágenes IOS. Como se ha comentado anteriormente, las imágenes de Cisco las encontramos comprimidas, por lo que debemos descomprimirlas y ubicarlas en la ruta por defecto de GNS3 *C:\Program Files\Dynamips\images,* aunque también podemos instalarlas en cualquier directorio. En las versiones actuales de GNS3, las imágenes de los dispositivos Cisco no están incluidas. Tenemos que descargar las imágenes de distintas páginas que hay en Internet que nos ofrecen las imágenes de los dispositivos.

3.3. INTERFAZ GRÁFICA

La interfaz gráfica que nos ofrece GNS3 es fácil de usar e intuitiva para los usuarios, no tiene menús complicados. La pantalla principal está dividida en zonas, tal y como se muestra en la Figura 8.

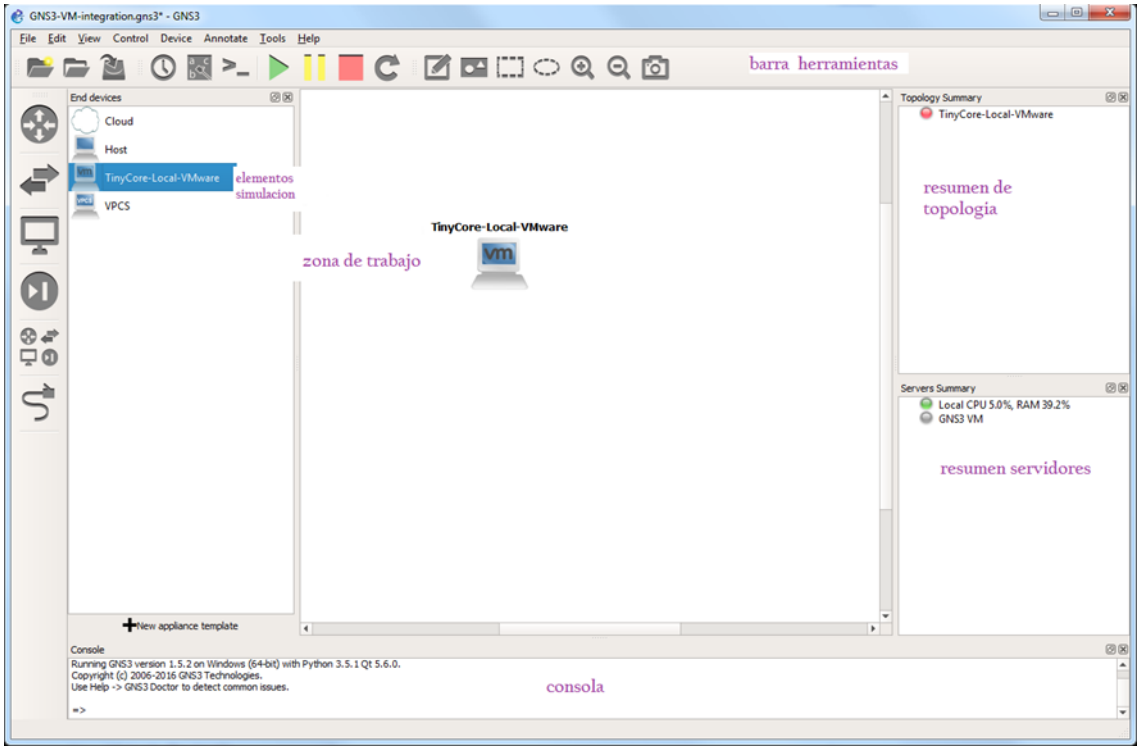

Figura 8. Interfaz gráfica.

En la interfaz se distinguen seis zonas de trabajo:

- *Barra de Herramientas*- parte superior de la pantalla, se corresponde con las operaciones básicas de la simulación (abrir, guardar, arrancar, captura de pantalla…).
- *Los elementos de la simulación* (dispositivos)- lateral izquierdo de la pantalla, donde están todos los elementos que necesitamos en la simulación (routers, hubs, switches, elementos de interconexión de equipos).
- *Zona de trabajo*- parte central de la pantalla, donde colocamos el diseño de la simulación (equipos, interconexiones…); para situar elementos en esta zona solo tenemos que elegirlos de la parte de elementos de simulación, luego arrastrarlos a la zona de trabajo y soltarlos.
- *Consola*- zona inferior de la pantalla, mediante la cual introducimos los comandos de configuración de los diferentes elementos de red.
- *Resumen de topología*- parte lateral derecho, zona superior; nos muestra cómo están los dispositivos. Unos leds indican el estado en el que se encuentran (funcionando, arrancando, apagado).
- *Resumen de servidores*- situada debajo de la zona de Resumen de topología, indica los servidores en uso y cómo se encuentran.

Estas zonas son configurables por el usuario, no es necesario que estén todas visibles, se pueden cerrar en cualquier momento.

3.4. METODOLOGÍA GENERAL PARA LA REALIZACIÓN DE LAS PRÁCTICAS

Vamos a seguir una metodología a la hora de realizar las diferentes prácticas de "Laboratorio de Diseño y Configuración de Redes". Algunos de los puntos son comunes para todas ellas, como la creación del proyecto, la simulación y la comprobación. Los cinco pasos en los que se basa el método de trabajo son:

- Creación de un proyecto nuevo.
- Diseño del esquema de red.
- Configuración y creación del tráfico de red.
- Simulación.

- Comprobación y análisis de resultados.

Creación de un Proyecto

En el momento de la creación del proyecto, tenemos que indicar la carpeta donde queremos almacenar nuestra simulación. En general, esta dirección es la indicada cuando hemos instalado GNS3, en este caso: *C:\Users\gemma\GNS3\projects\nombre_del_proyecto.*

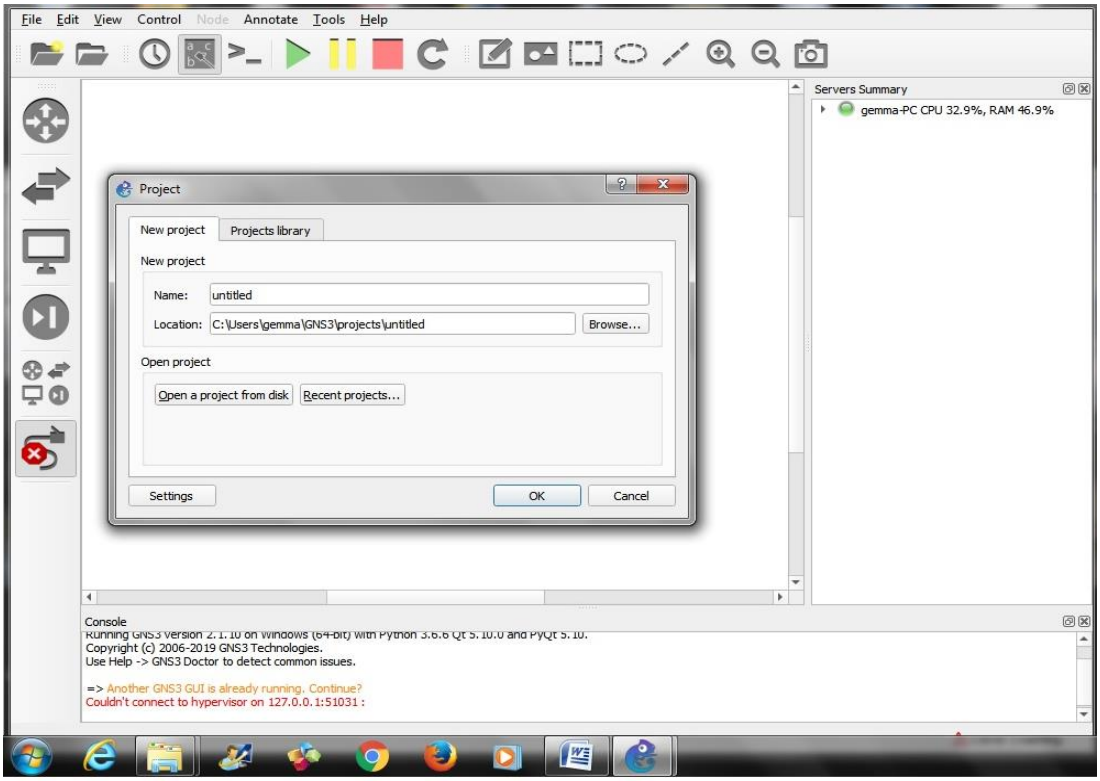

Figura 9. Creación de proyecto GNS3.

En la Figura 9 se ve el cuadro de diálogo donde indicamos el nombre que vamos a poner a nuestra simulación y la ubicación donde se creará la carpeta donde se instalarán todos los archivos relacionados con el diseño. Si no queremos un proyecto nuevo también nos da la opción de abrir desde un elemento externo.

Diseño del esquema de red y Configuración y Creación del tráfico de red

Después de la creación de proyecto, ya tenemos todo preparado para empezar con el diseño de red, arrastrando los componentes a la *zona de trabajo* e interconectándolos posteriormente. El diseño y la configuración de los dispositivos de red depende del problema concreto que se quiera analizar.

Comprobación del diseño

El tercer punto en común en todas las prácticas es la simulación. Una vez realizado el diseño de la red, tenemos que comprobar que todos los dispositivos están conectados correctamente. Para ello en la *barra de herramientas* veremos un triángulo de color verde, que pulsaremos y si todos los enlaces aparecen en verde, esto indicará que todos los elementos están bien conectados y a partir de ahí ya podemos configurar cada uno de ellos.

Simulación y visualización de los resultados

Al finalizar la simulación, un objetivo de las prácticas es la realización de pruebas para comprobar el funcionamiento de la red, descubriendo los posibles fallos en el diseño o si cumple las especificaciones que se indican al inicio de las prácticas. Para ello contamos con diferentes programas que podemos instalar cuando instalamos el paquete de GNS3, que aparecen en la Tabla 5, como Wireshark, con el que podemos capturar el tráfico que se genera en los enlaces. Cuando pichamos con el ratón en el enlace, podemos indicar fallos o retrasos, como se muestra en la Figura 10. También con los comandos de la consola vemos los resultados de la conectividad.

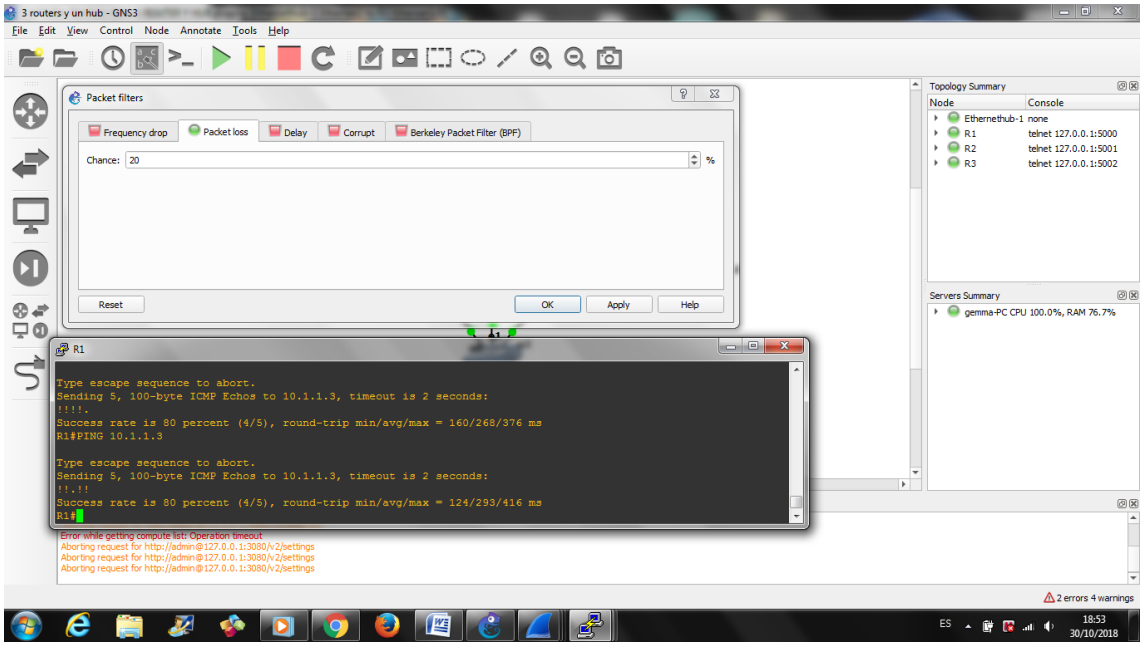

Figura 10. Comprobación de la simulación.

El diseño y la configuración y la generación del tráfico son dos puntos que son distintos en cada una de las prácticas, dependiendo del servicio o protocolo que se vaya a simular.

3.5. TOPOLOGÍAS BÁSICAS (PRIMEROS PASOS CON GNS3)

Antes de empezar con el diseño de las prácticas, el usuario debe tener unas nociones básicas de cómo funciona GNS3.

Una de las desventajas que tiene el simulador es que, al no tener licencia propietaria, no proporciona las imágenes de los componentes para realizar las simulaciones, pero GNS3 ofrece múltiples formas de emular IOS. En versiones antiguas GNS3 usa y mantiene Dynamips, pudiendo ejecutar imágenes IOS sin modificarlas. Tenemos que buscar las imágenes en diversas páginas que puedes encontrar en Internet donde te las puedes descargar como, por ejemplo: *www.telectronika.com/descargas/cisco-imagenes-ios-para-gns3-dynamips-y-vm.* Para su uso hay que tener en cuenta la memoria interna que necesita la imagen para que funcione a máxima capacidad y un valor propio del simulador denominado IDLE-PC que ayudará a Dynamips a consumir menos CPU. Los dispositivos más utilizados son los componentes Cisco de las series C2600, C3620, C3640…

Otra manera de instalar dispositivos es desde el *MarketPlace* que ofrece GNS3, donde podemos encontrar imágenes de IOSvL2, ASA… que se usan con la máquina virtual. Estos dispositivos los tenemos que descargar en el directorio *appliances* dentro del directorio general de GNS3.

CONFIGURACIÓN DE UN ROUTER

En todos los ejemplos y configuraciones que vamos a realizar desde ahora hasta el final del proyecto, se van a utilizar imágenes de IOS de Cisco, ya que son las más utilizadas y las más sencillas de conseguir para el usuario.

Para configurar un router, tenemos que iniciar el procedimiento desde consola, colocando el puntero del ratón, en el router a configurar y pulsando el botón derecho elegiremos la opción correspondiente a la consola del router. La primera sentencia que tenemos que introducir es:

- `Config Terminal`, `Config T` Ó `Conf T` (cualquiera de las tres opciones es válida para el inicio de la configuración).

Lo primero a configurar serán las interfaces de red. Un router tendrá varias interfaces, que pueden ser de distintos tipos. Cuando se configura cada una de ellas, tenemos que indicar el tipo y el identificador de la interfaz a configurar mediante el comando `Interface (Ethernet, Fast, Serial)` o `int eX/X`. Por ejemplo, con el siguiente comando iniciaríamos la configuración de la interfaz ethernet E1/0 del router R2:

- `Interface EX/X`

La configuración básica de la interfaz consiste en asignar la dirección IP y máscara que vamos a utilizar en la interfaz:

- `IP ADDRESS X.X.X.X X.X.X.X`

A continuación, se debe activar la interfaz, con el comando `NO SHUTDOWN`, o la versión abreviada `no shut`:

- `NO SHUTDOWN`

Salimos de la configuración de la interfaz con `EXIT`

- `EXIT`

Por último, para que todo se quede almacenado y no tengamos que configurarlo en cada simulación, escribimos el comando `WR (R2#WR)`, que guarda la configuración.

A continuación, se muestra un ejemplo

```
R2#Conf T
R2(config)#INT E1/0
R2(config-if)#IP ADDRESS 10.1.1.2 255.255.255.0
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#exit
```

Si todos los pasos se realizan correctamente, veremos un mensaje `Building configuration... [OK]`

CONFIGURACIÓN DE UN SISTEMA FINAL

Los ordenadores (o PCs) son elementos finales de las redes, por lo que hay que configurar su dirección IP, máscara de red y la dirección de la salida por defecto (o *gateway*). En la configuración, podemos usar dos formas diferentes, pudiendo poner la máscara tanto en formato decimal como en notación prefijo, tal y como se muestra en los siguientes dos ejemplos:

```
PC1: 192.168.100.126/26 gateway 192.168.100.65
PC1: 192.168.100.126 255.255.255.192 gateway 192.168.100.65
```

En algunas versiones se puede omitir el comando `Gateway`.

CONFIGURACIÓN DE UN ROUTER COMO UN PC

GNS3 nos da la opción de configurar un router con un elemento final de red. En la Figura 11, vemos dos routers: el router de la izquierda tiene la apariencia de un PC (pc-router) y el de la derecha tiene una apariencia clásica de un router.

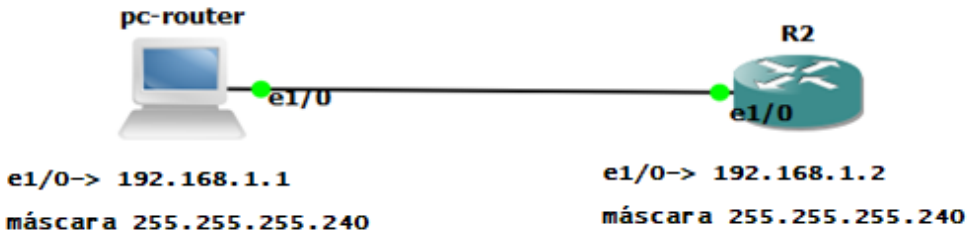

Figura 11. Configuración de un router como un PC.

Para usar el router como pc-router, lo primero es cambiar el icono de router por uno de ordenador. Después de este cambio, realizamos la configuración del PC desde consola.

- `Conf T-` indicamos que vamos a configurar el dispositivo desde el terminal.
- `pc-router(config)#no ip routing-` con este comando desactivamos la función de encaminamiento del dispositivo para que trabaje como un ordenador.

El resto de la configuración es como si configuráramos un router, indicando el primer lugar que vamos a configurar desde el terminal y lo primer es indicar que no queremos que el dispositivo trabaje con la función de encaminamiento, y lo siguiente definir la interfaz que usaremos. Los últimos pasos son definir la dirección IP y la máscara, así como la dirección de salida por defecto. Todo este procedimiento se muestra a continuación para el ejemplo de `pc-router`:

```
pc-router#Conf T
pc-router(config)#no ip routing
pc-router(config)#int e1/0      //indicamos la interfaz
pc-router(config-if)#ip address 192.168.1.1 255.255.255.240
pc-router(config-if)#ip default-network 192.168.1.0
pc-router(config-if)#no shut
pc-router(config)#exit
```

Con estas sentencias ya tendríamos la configuración completa y no solo nos quedaría guardar la configuración usando `WR`.

Éstos son los pasos que se deben seguir para conseguir usar un router por un ordenador. Es una configuración sencilla que nos puede ser útil en un momento dado.

CONFIGURACIÓN DE UN ROUTER COMO UN SWITCH

En GNS3 los switches tienen unas funcionalidades limitadas, por lo que tenemos que conseguir que un dispositivo realice todas las funciones que ofrece un switch. Los routers Cisco de la gama 3700 permiten que los configuremos como un switch, ya que cuentan con un módulo NS-16ESW. Para realizar esta operación haremos lo siguiente:

- Instalación de la imagen del sistema operativo IOS del Router 3725. En la configuración tenemos que activar el módulo NM-16ESW, en la Figura 12 podemos ver cómo definimos los aspectos generales del switch.

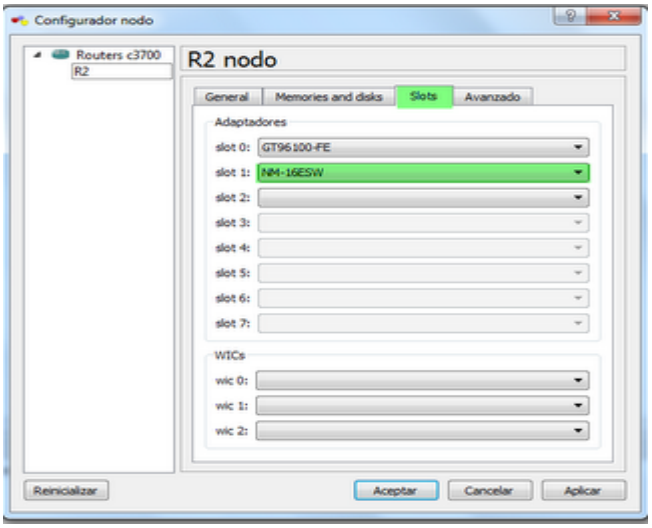

Figura 12. Configuración del Módulo NM-16ESW.

- Como en el caso de la configuración del PC como Router, podemos cambiar la apariencia en el menú `Edit` con la opción `Symbol Manager`, eligiendo un icono de switch.

- El switch es configurado con 16 puertos con un rango de Fastethernet 1/0 hasta el 1/15. Hay que activar los 16 puertos.

- Desde consola configuramos el switch con los siguientes comandos:

  - `R1#show ip interface brief` -ver los puertos que tenemos activos, los cuales aparecen en la columna Status.

  - `R1#config T` -configuración del terminal.

  - `R1(config)#hostname sw1` -definición de un nombre para el dispositivo

  - `sw1(config)#interface rango FastEthernet 1/0 -15` -rango de la interfaces activas.

  - `sw1(config-if-range)#no shut` -activación de las interfaces.

  - `sw1(config-if-range)#switchport mode trunk` -configuración de una interfaz como enlace troncal, por donde vamos a dejar pasar todas las VLANs.

- Salimos de la configuración con el comando `Exit`, y guardamos con `WR`. Con esta operación ya estaría configurado el switch.

EJEMPLO SENCILLO: Conexión de un Router y un PC

En la Figura 13 se muestra una red sencilla, la cual consistirá en la conexión de un router y un PC, configurando cada uno de los dos dispositivos, comprobando la conectividad entre ellos y viendo los resultados.

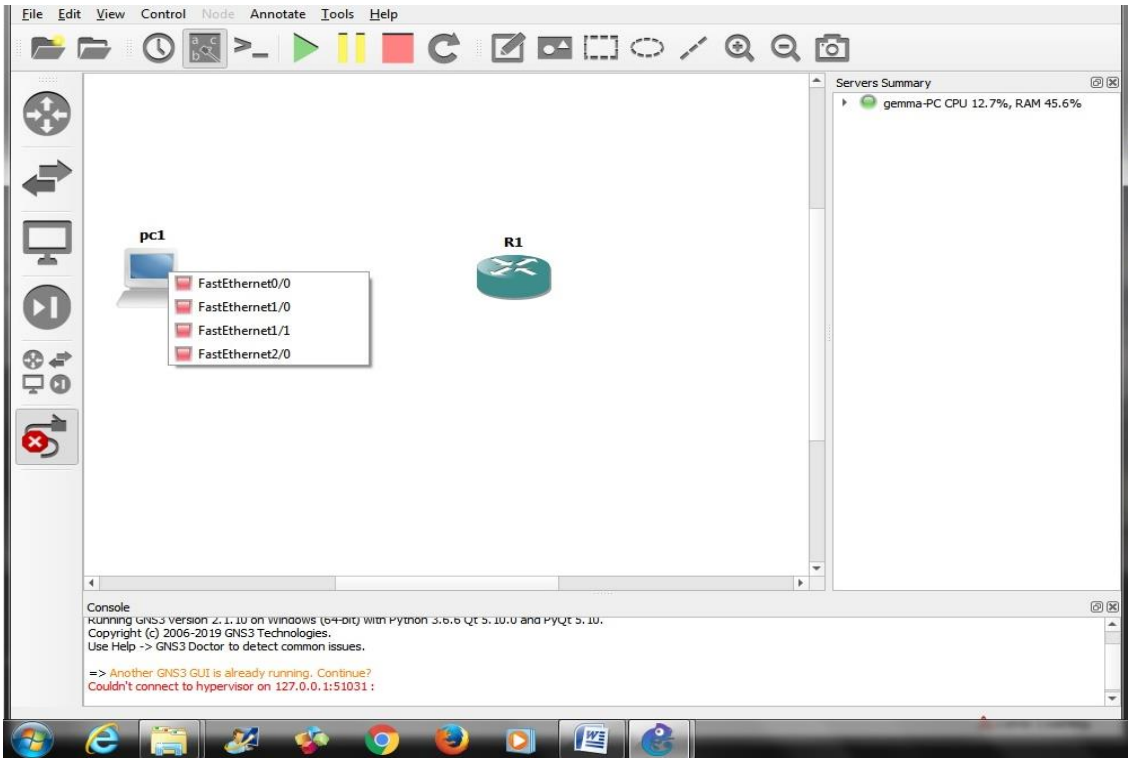

Figura 13 Conexión router y Pc.

Comenzamos colocando los elementos que intervienen en la configuración en la zona de trabajo. Colocamos un router y un pc, elegimos uno dentro de la lista que nos aparecen y lo arrastramos a la zona. Lo siguiente es la conexión entre los dispositivos. En la columna de la izquierda, donde están los dispositivos, el último icono corresponde a elementos de conexión (cables). Una vez pinchado el icono, arrastramos el puntero desde el origen al destino, eligiendo la interfaz que vamos a usar en cada uno de ellos y su tipo de conexión. Cuando los tenemos unidos, pulsamos el botón del **Play** (triángulo verde, en la parte superior de la pantalla).

Para que la configuración esté acabada, tenemos que configurar los dos dispositivos. Pulsando el botón derecho del ratón sobre el icono del router o pc, nos aparece un menú. Elegiremos la opción consola y desde allí; configuraremos el PC y el router de la forma especificada anteriormente:

- Configuración del PC:

```
PC1: 192.168.100.126 255.255.255.192 gateway 192.168.100.65
```

- Configuración del Router:

```
R1#Config Terminal
R1(config)#INT E1/0
R1(config-if) #IP ADDRESS 192.168.100.127 255.255.255.192
R1(config-if) #NO SHUT
R1(config-if) #EXIT
R1#WR
```

Para finalizar este pequeño ejercicio solo nos quedaría comprobar que hay tráfico entre origen y destino, usando PING o algún comando similar.

3.6. PRÁCTICAS DE LABORATORIO

3.6.1 DIRECCIONAMIENTO IP

La primera práctica que se plantea en la asignatura "Laboratorio de Diseño y Configuración de Redes" es el diseño del esquema de *direccionamiento IP* y con ello la realización de *subnetting.*

El objetivo de esta primera práctica es la configuración y direccionamiento de una red IP, con la utilización de los dispositivos de red (PC, routers, switches…), y la realización de *subnetting* con direcciones de tipo C (*subnetting*- subdivisión de la red en varias redes de distinto tamaño).

Se diseña una red compleja, utilizando diferentes dispositivos de red (PCs, routers, switches…) y se realiza un *subnetting* partiendo de una dirección de tipo C (esta clase se distingue por el rango de direcciones 192.0.0.0-223.255.255.255), donde solo se alberga el último octeto para las direcciones de hosts y para hacer *subnetting* (IONOS España, 16).

EJEMPLOS RELACIONADOS CON EL DIRECCIONAMIENTO IP

El simulador GNS3 permite configurar el direccionamiento IP de una red. En relación con el simulador Riverbed Modeler, el diseño de red, para la práctica de *subnetting* será más sencillo (el número de elementos será menor), pero se conseguirán los objetivos establecidos en la práctica.

Ejemplo 1

La configuración de este ejercicio consta de la conexión de 3 ordenadores a un router. Este diseño es sencillo y podemos ver como se realiza el *subnetting* con el simulador.

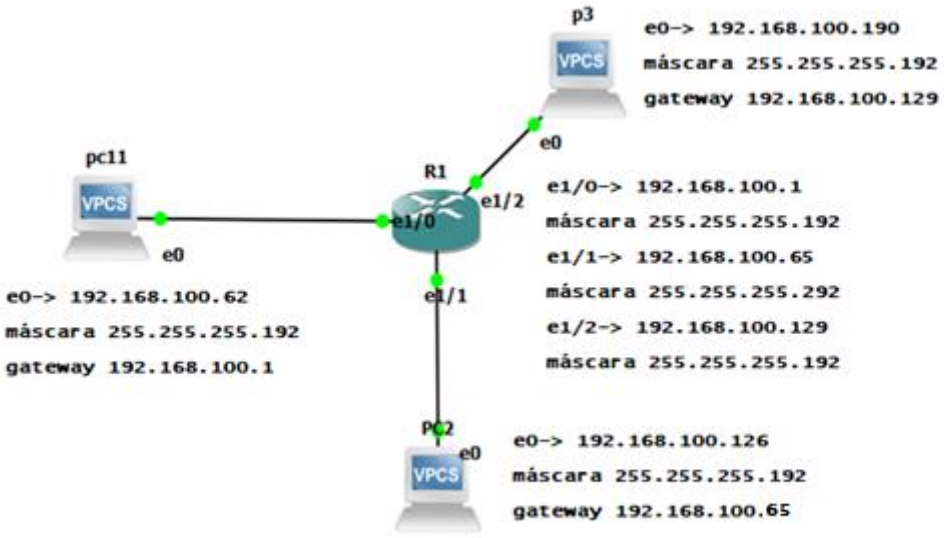

Figura 14. Ejemplo 1 - Direccionamiento IP.

En el esquema se pueden ver los 3 ordenadores identificados con los nombres: `pc11`, `p3` y `PC2` y el router con el nombre de `R1`. Al lado de cada elemento (ordenadores) podemos ver la dirección IP que se les ha asignado, así como la máscara correspondiente a esa dirección; en el caso del router se muestran la direcciones de cada una de las interfaces que intervienen en el diseño.

COMPONENTE	DIRECCIÓN IP	MÁSCARA	INTERFAZ
ORDENADOR 1 (PC11)	192.168.100.62	255.255.255.192	e0
ORDENADOR 2 (PC2)	192.168.100.126	255.255.255.192	e0
ORDENADOR 3 (P3)	192.168.100.190	255.255.255.192	e0
ROUTER 1 (R1)	192.168.100.1	255.255.255.192	e1/0
	192.168.100.65	255.255.255.192	e1/1
	192.168.100.129	255.255.255.192	e1/2

Tabla 7. Ejemplo 1- Direccionamiento IP configuración.

La Tabla 7 indica las direcciones que intervienen, con la dirección IP, máscara y la interfaz a la que se conecta en cada caso. Una vez definido el esquema de diseño y la información correspondiente a la configuración de cada uno de los equipos, trasladamos toda esta información al simulador.

Después de dibujar, conectamos los equipos activando la simulación. Si las conexiones son correctas, veremos unos puntos verdes que indican que la conexión es correcta; si hay alguno en rojo, la conexión no está bien realizada; y si es amarillo/anaranjado, es que se está reiniciando (hay que esperar unos segundos). Todo en verde, nos indica que ya podemos empezar con la configuración de cada componente.

El siguiente paso es la configuración de los equipos. Abrimos la consola de cada uno de los elementos que intervienen en el diseño. Lo primero son los ordenadores:

```
P3> ip 192.168.100.190/26 192.168.100.129

Pc11> ip 192.168.100.62/26 192.168.100.1

PC2> ip 192.168.100.126/26 192.168.100.65
```

Para la configuración de los ordenadores, se pueden usar las dos opciones explicadas en el apartado de *Configuración de un sistema final*, introduciendo la máscara tanto con el formato decimal como en notación prefijo.

El router tiene más pasos en la configuración antes de que todo funcione completamente. Desde la consola, lo primero es indicar que vamos a configurar el terminal.
Indicamos la interfaz que vamos a configurar e introducimos la dirección IP y la máscara, y guardamos la configuración con el comando `WR` después de Exit.

```
R1#CONF T
R1(config)#INT E1/0
R1(config-if)#IP ADDRESS 192.168.100.1 255.255.255.192
R1(config-if)#NO SHUT
R1(config-if)#EXIT
R1#WR            // guardar el resultado conseguido
Building configuration... [OK]
```

Repetimos este conjunto de operaciones para las otras dos interfaces que tienen conexión, interfaz E1/1:

```
R1#CONF T
R1(config)#INT E1/1
R1(config-if)#IP ADDRESS 192.168.100.65 255.255.255.192
R1(config-if)#NO SHUT
R1(config-if)#EXIT
```

Y para la interfaz E1/2:

```
R1#CONF T
R1(config)#INT E1/2
R1(config-if)#IP ADDRESS 192.168.100.129 255.255.255.192
R1(config-if)#NO SHUT
R1(config-if)#EXIT
```

No hay que olvidarse de guardar la configuración con `WR` cada vez que se haga una nueva configuración o una modificación de una configuración existente.

En este momento ya está todo correcto para empezar a realizar pruebas con la red. Si queremos comprobar cuáles son las interfaces que están activas y su dirección, se usa el comando `Show ip interface brief`, desde la consola del router. La salida del comando es la siguiente:

```
R1#show ip interface brief
 Interface        IP-Address         OK?   MethodStatus             Protocol
 FastEthernet0/0  unassigned         YES   unset administratively down down
 Ethernet1/0      192.168.100.1      YES   manual up                    up
 Ethernet1/1      192.168.100.65     YES   manual up                    up
 Ethernet1/2      192.168.100.129    YES   manual up                    up
 Ethernet1/3      unassigned         YES   unset administratively down down
 FastEthernet2/0  unassigned         YES   unset administratively down down
 FastEthernet2/1  unassigned         YES   unset administratively down down
 R1#
```

Vemos cuáles son las interfaces que están activas, si están configuradas correctamente, y la dirección IP que se les ha asignado. También se muestra el protocolo de encaminamiento activo y el modo de configuración empleado (en este caso ha sido manualmente). Las interfaces no configuradas aparecen como **unassigned** (no asignadas).

Para comprobar la conectividad entre puntos, realizamos un **Ping.** Vamos a ver la conexión entre el ordenador PC2 y la interfaz del router al que está conectado (dirección IP 192.168.100.65).

El comando **Ping** nos va a permitir ver si existe conectividad entre puntos, en este caso vamos a comprobar la conectividad desde el ordenador PC2 a la dirección 192.168.100.65. Esta comprobación es igual para el resto de las conexiones.

```
PC2> ping 192.168.100.65
84 bytes from 192.168.100.65 icmp_seq=1 ttl=255 time=33.002 ms
84 bytes from 192.168.100.65 icmp_seq=2 ttl=255 time=4.000 ms
84 bytes from 192.168.100.65 icmp_seq=3 ttl=255 time=7.001 ms
84 bytes from 192.168.100.65 icmp_seq=4 ttl=255 time=5.000 ms
84 bytes from 192.168.100.65 icmp_seq=5 ttl=255 time=11.000 ms
```

Hay conectividad entre el ordenador PC2 y el router, como comprobamos con el resultado del comando PING. Nos muestra los 5 intentos de conexión entre los extremos. Si lo que queremos es comprobar la conectividad entre ordenadores hacemos el PING al PC11; si la configuración es correcta obtendremos:

```
PC2> ping 192.168.100.62
192.168.100.62 icmp_seq=1 timeout
192.168.100.62 icmp_seq=2 timeout
84 bytes from 192.168.100.62 icmp_seq=3 ttl=63 time=16.001 ms
84 bytes from 192.168.100.62 icmp_seq=4 ttl=63 time=23.001 ms
84 bytes from 192.168.100.62 icmp_seq=5 ttl=63 time=13.001 ms
```

Pero si no está bien configurado, al hacer la prueba, tendremos este resultado:

```
PC2> ping 192.168.100.62
192.168.100.62 icmp_seq=1 timeout
192.168.100.62 icmp_seq=2 timeout
192.168.100.62 icmp_seq=3 timeout
192.168.100.62 icmp_seq=4 timeout
192.168.100.62 icmp_seq=5 timeout
```

El tiempo se agota, y no se consigue la conectividad entre los dos puntos.

Ejemplo 2

En este segundo ejemplo, como nos muestra la Figura 15, todos los elementos utilizados en el diseño son routers de la serie de Cisco C7200, y uno de ellos lo configuramos como si fuera un PC, para practicar el uso de router como otros dispositivos. En este caso el R1 se configurará como un ordenador, un dispositivo final. La apariencia la hemos cambiado por el símbolo Computer. GNS3 permite convertir equipos en otros dispositivos, como se ha visto anteriormente, un router según lo configures puedes ser un PC, Switch o Router. En este ejemplo, se configura un router Cisco de la serie C7200 como PC.

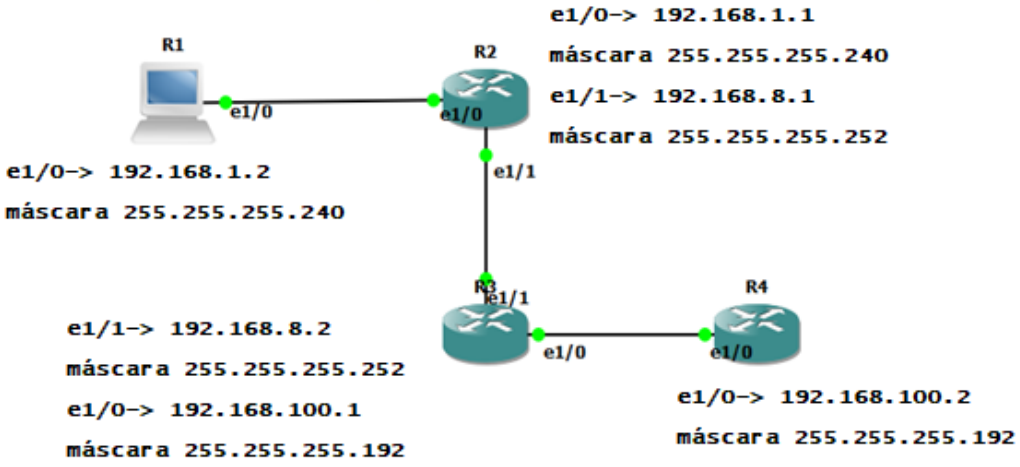

Figura 15. Ejemplo 2. Direccionamiento IP.

La configuración de este ejemplo es similar a la configuración hecha anteriormente, hay que definir las interfaces de cada router y sus direcciones IP, así como la configuración del elemento final de la red (PC con el nombre de R1). En la Tabla 8 se muestra un resumen de toda la información relacionada con el ejemplo.

COMPONENTE	DIRECCIÓN IP	MÁSCARA	INTERFAZ
R1 (Router como PC)	192.168.1.2	255.255.255.240	e1/0
R2 (router)	192.168.1.1	255.255.255.240	e1/0
	192.168.8.1	255.255.255.252	e1/1
R3 (router)	192.168.8.2	255.255.255.252	e1/1
	192.168.100.1	255.255.255.192	e1/0
R4 (router)	192.168.100.2	255.255.255.192	e1/0

Tabla 8. Ejemplo 2- Direccionamiento IP configuración.

Al finalizar la configuración de todos los dispositivos, podemos comprobar que todo funciona correctamente y que la configuración es la adecuada. El comando **SHOW** da información sobre distintos elementos/ funcionalidades de la red: interfaces, versiones, estado… un ejemplo de su uso lo hemos visto ya anteriormente: **R1#show ip interface brief.** Otras pruebas que se pueden realizar son mediante un **TRACEROUTE** (determina la ruta que toma un paquete para alcanzar su destino). Un ejemplo de Traceroute relacionado con el ejemplo sería:

```
R4>traceroute 192.168.1.2.
```

Ejemplo 3

En este tercer ejemplo, partimos de la Figura 16 una red de 3 Routers (R1, R2, R3) conectados a un Hub (Ethernethub-1).

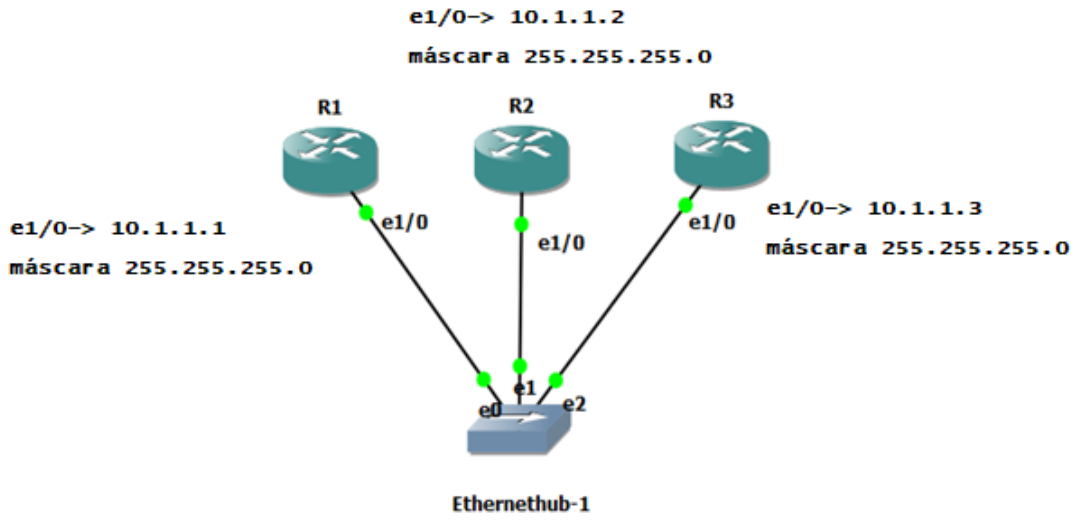

Figura 16. Ejemplo 3. Direccionamiento IP.

Las direcciones de configuración las vemos en la Tabla 9.

COMPONENTE	DIRECCIÓN IP	MÁSCARA	INTERFAZ
R1 (router)	10.1.1.1	255.255.255.0	e1/0
R2 (router)	10.1.1.2	255.255.255.0	e1/0
R3 (router)	10.1.1.3	255.255.255.0	e1/0

Tabla 9. Ejemplo 3- Direccionamiento IP configuración.

Los pasos para la configuración del Router 2 son los siguientes:

```
R2#CONF T
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#INT E1/0
R2(config-if)#IP ADDRESS 10.1.1.2 255.255.255.0
R2(config-if)#NO SHUT
R2(config-if)#EXIT
R2(config)#
R2(config)#EXIT
R2#WR
Building configuration...[OK]
R2#
```

Los pasos para configurar los otros dos routers serían equivalentes.

El comando **SHOW IP INTERFACE BRIEF** muestra un resumen con la información del uso de cada interfaz que dispone el dispositivo, en este caso del Router 2:

```
R2#SHOW IP INTERFACE BRIEF
Interface          IP-Address   OK?  Method Status                Protocol
FastEthernet0/0    unassigned   YES  unset administratively down  down
Ethernet1/0        10.1.1.2     YES  manual up                    up
Ethernet1/1        unassigned   YES  unset administratively down  down
Ethernet1/2        unassigned   YES  unset administratively down  down
Ethernet1/3        unassigned   YES  unset administratively down  down
FastEthernet2/0    unassigned   YES  unset administratively down  down
FastEthernet2/1    unassigned   YES  unset administratively down  down
R2#
```

Viendo el resumen comprobamos que tenemos activa la interfaz que hemos configurado en el Router 2 y el resto de las interfaces que tiene no están asignados ni

activas para su uso. Si hacemos una prueba de conectividad entre un router y otro con el comando **`PING`** comprobamos que todo funciona correctamente:

```
R1#PING 10.1.1.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 184/270/384
ms
R1#
```

Wireshark nos permite ver los paquetes que se transmiten entre los dos puntos del enlace; veremos algo similar a lo que se muestra en la Figura 17 cuando lo ejecutemos. La información que nos enseña es el origen y el destino con sus direcciones IP, el tipo del protocolo e información adicional.

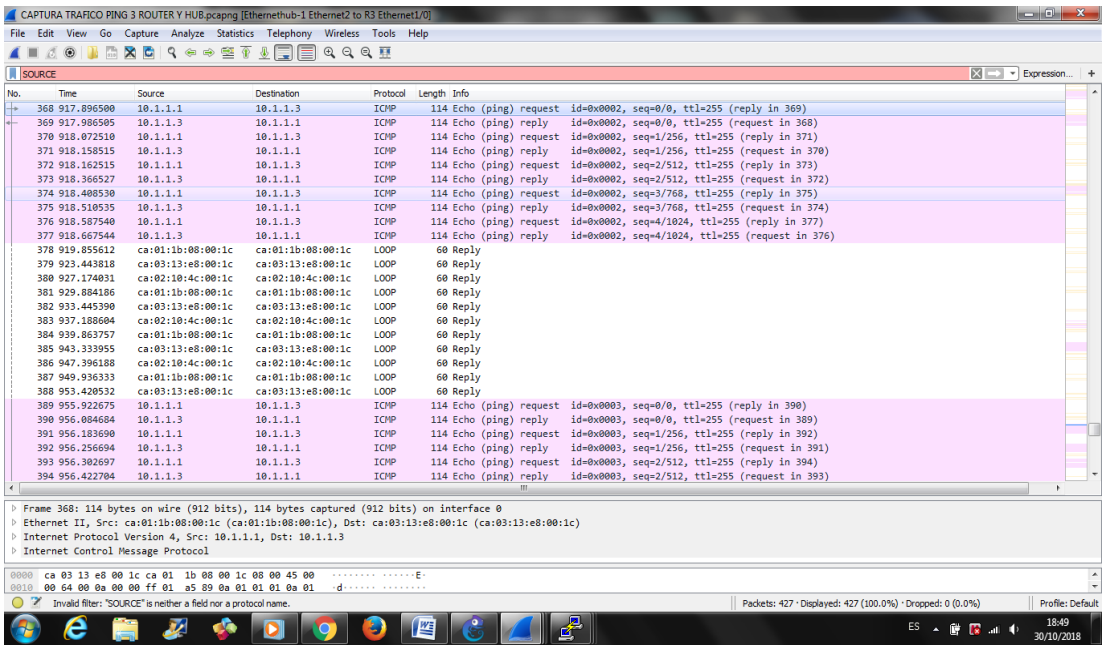

Figura 17. Captura de tráfico usando WIRESHARK.

Otra de las opciones que nos permite el simulador es filtrar paquetes. Para tener información estadística de los mismos, dentro de las opciones que podemos definir tenemos:

- *Frequency Drop*- variación de envío de paquetes.
- *Packet Loss*- pérdida de paquetes, en el envío.
- *Delay*- retraso en el envío.
- *Corrupt*- definir que alguno de los paquetes que se transmite tenga alguna parte errónea.

- *Berkeley Packet Filter (BPF)*-envío y recibo de paquetes de capa de enlace sin procesar.

Estos parámetros los definimos desde GNS3, como se puede observar en la Figura 18, donde se ven las pestañas en las que se pueden introducir los valores de la simulación.

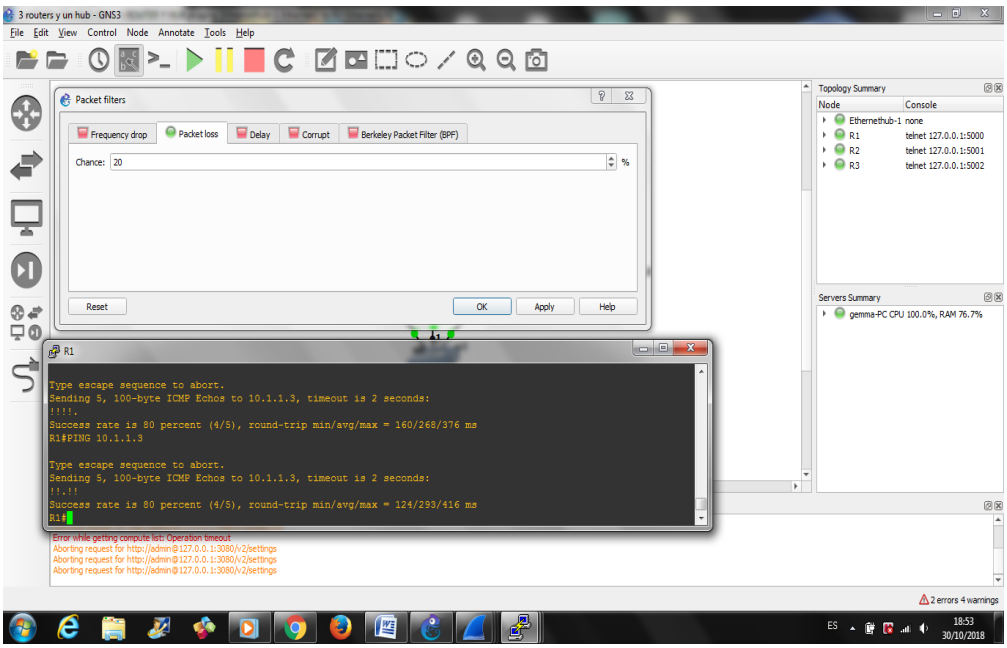

Figura 18. Pérdida de paquetes.

Todos estos ejemplos nos permiten ver la funcionalidad que tiene el simulador GNS3 para el direccionamiento IP, y la realización de *subnetting*. Se ha comprobado que se puede elaborar una práctica similar a la que se pide en la asignatura, asignando direcciones IP a los distintos componentes que intervienen en la red, asignando las direcciones de manera manual, así como sus máscaras de red. Además, se puede analizar el tráfico que se genera entre los extremos. GNS3 nos permite la configuración del tráfico de una manera sencilla, a través de la definición de tasa de pérdida de paquetes o retrasos, y el programa Wireshark permite comprobar estas configuraciones.

3.6.2 PROTOCOLO DE ENCAMINAMIENTO RIP (v1 y v2)

En la segunda práctica relacionada con la asignatura, se estudia el protocolo de encaminamiento RIP en sus distintas versiones (RIPv1 y RIPv2). Es un protocolo de encaminamiento simple basado en vector distancia y diseñado para proporcionar una tabla de encaminamiento estable con apenas configuración.

En los distintos ejemplos realizados, vemos cómo se implementa el protocolo y cómo se construye la tabla de encaminamiento, así como los cambios que se producen según las circunstancias que se generan en la red.

Ejemplo 1

En la Figura 19, vemos el esquema de la red, que consta de 7 routers. El planteamiento es ver cómo trabaja el protocolo RIP y cuál es el camino elegido para la comunicación entre los extremos de la red (router R1 y el router R7). Como se puede ver en el esquema, existen dos rutas posibles: una a través del router R5 y la otra por los routers R2 y R3.

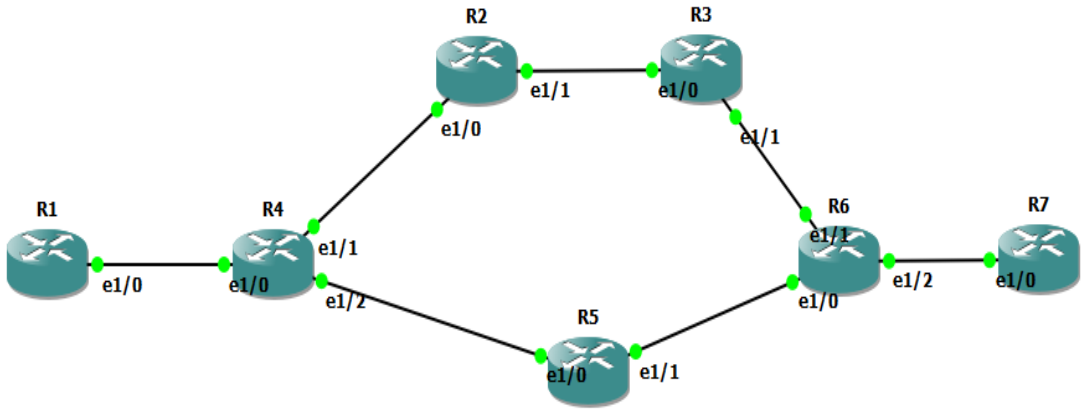

Figura 19.Ejemplo 1. RIP.

Empezamos con la configuración de cada uno de los dispositivos que intervienen en la red (la configuración de los router es la misma que se ha utilizado en la primera práctica de direccionamiento IP, utilizando los comandos `conf t`, `ip add`, `no shutdown` y `exit`).

En este diseño, además, se le añade la configuración de la interfaz de *loopback*, se pone para tener latente el protocolo en los extremos R1 yR7.

```
R1(config)#int loopback 0
R1(config-if)#
*Nov   8 10:22:52.579: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Loopback0, changed state to up
R1(config-if)#ip add 160.15.36.10  255.255.255.0
R1(config-if)#no sh
R1(config-if)#exit
```

Seguimos el mismo procedimiento para el otro extremo. Para comprobar que están bien configuradas las interfaces de los extremos ejecutamos `show ip int brief`; como resultado obtenemos:

```
R1#show ip int brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            unassigned      YES unset  administratively down down
Ethernet1/0                160.15.50.1     YES manual up                    up
Ethernet1/1                unassigned      YES unset  administratively down down
Ethernet1/2                unassigned      YES unset  administratively down down
Ethernet1/3                unassigned      YES unset  administratively down down
FastEthernet2/0            unassigned      YES unset  administratively down down
FastEthernet2/1            unassigned      YES unset  administratively down down
Loopback0                  160.15.36.10    YES manual up                    up
```

El resto de configuración que se tiene que realizar, la vemos con la información que se muestra en la Figura 20, y en la Tabla 10, donde se dan las direcciones que participan en el ejemplo 1.

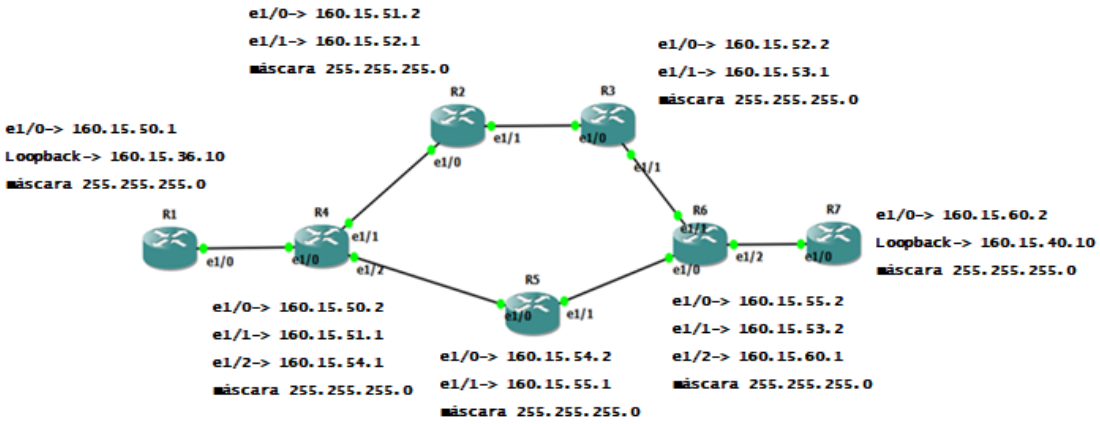

Figura 20. Ejemplo 1. RIP esquema de red.

COMPONENTE	DIRECCIÓN IP	MÁSCARA	INTERFAZ
R1 (router)	160.15.50.1	255.255.255.0	e1/0
	160.15.36.10	255.255.255.0	loopback
R2 (router)	160.15.51.2	255.255.255.0	e1/0
	160.15.51.1	255.255.255.0	e1/1
R3 (router)	160.15.52.2	255.255.255.0	e1/0
	160.15.53.1	255.255.255.0	e1/1
R4 (router)	160.15.50.2	255.255.255.0	e1/0
	160.15.51.1	255.255.255.0	e1/1
	160.15.54.1	255.255.255.0	e1/2
R5 (router)	160.15.54.2	255.255.255.0	e1/0
	160.15.55.1	255.255.255.0	e1/1
R6 (router)	160.15.53.2	255.255.255.0	e1/1
	160.15.55.2	255.255.255.0	e1/0
	160.15.60.1	255.255.255.0	e1/0
R7 (router)	160.15.60.2	255.255.255.0	e1/0
	160.15.40.10	255.255.255.0	loopback

Tabla 10. Ejemplo 1-RIP configuración.

El siguiente paso es la configuración del protocolo de encaminamiento RIP en todos los routers. El primero es el router R1:

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router rip
R1(config-router)#net 160.15.50.1
R1(config-router)#net 160.15.36.10
R1(config-router)#no sh
R1(config-router)#end
R1#wr
Building configuration...
```

Con el comando `router rip` definimos que tipo de protocolo de encaminamiento vamos a implementar en el router, en este caso RIP. Y a continuación le decimos las redes que están conectadas al dispositivo con `net xxx.xxx.xxx.xxx`, en el caso de utilización de routers de la familia Cisco, su configuración nos permite introducir la dirección de la interfaz conectada en vez de la dirección de la red para el protocolo RIP. Esta forma de configuración es la que se realizará durante todos los ejemplos del protocolo RIP en sus distintas versiones:

```
R1(config-router)#net 160.15.50.1
R1(config-router)#net 160.15.36.10
```

En este caso, el router tiene dos redes, una es *loopback* y otra la de salida a la red. Esta operación la realizamos de forma equivalente en los demás routers. No hay que olvidarse de guardar los datos de la configuración al terminar, con el comando `write`.

Configuración de todos los elementos de la red:

```
R2:
        R2(config)#router rip
        R2(config-router)#net 160.15.50.1
        R2(config-router)#net 160.15.50.2
R3:
        R3(config)#router rip
        R3(config-router)#net 160.15.52.2
        R3(config-router)#net 160.15.53.1
R4:
        R4(config)#router rip
        R4(config-router)#net 160.15.50.2
        R4(config-router)#net 160.15.51.1
        R4(config-router)#net 160.15.54.1
R5:
        R5(config)#router rip
        R5(config-router)#net 160.15.54.2
        R5(config-router)#net 160.15.55.1
R6:
        R6(config)#router rip
        R6(config-router)#net 160.15.53.2
        R6(config-router)#net 160.15.55.2
        R6(config-router)#net 160.15.60.1
R7:
        R7(config)#router rip
        R1(config-router)#net 160.15.60.2
        R1(config-router)#net 160.15.40.10
```

Esta es la configuración completa para la implementación del protocolo de encaminamiento RIP.

Lo siguiente es la realización de pruebas para comprobar que el funcionamiento es correcto. El primer paso es comprobar la conectividad entre los extremos (router R1 y router R7), mediante un `ping` desde R1:

```
R1#ping 160.15.40.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 160.15.40.10, timeout is 2 seconds:
.!!!.
Success rate is 60 percent (3/5), round-trip min/avg/max = 1572/1757/2000
ms
```

El resultado indica que el funcionamiento es correcto. El comando `ping` nos muestra el número de solicitudes enviadas y su tamaño y el *timeout* para la conexión, así como el porcentaje de éxito y el tiempo de ida y vuelta de cada una de las conexiones realizadas con éxito. Como lo que nos interesa es saber el camino que sigue, ya que RIP es un protocolo de encaminamiento, la siguiente prueba es realizar un `traceroute`:

```
R1#traceroute 160.15.40.10
Type escape sequence to abort.
Tracing the route to 160.15.40.10
1 160.15.50.2 512 msec 528 msec 488 msec
2 160.15.54.2 1332 msec 1100 msec 1120 msec
3 160.15.55.2 1704 msec 1648 msec 1912 msec
4 160.15.60.2 2564 msec 2292 msec 2092 msec
```

En el primer intento que realiza no es posible conseguir los datos, en el siguiente aparece la ruta completa que sigue para la conexión entre un extremo y el otro. Como se muestra en la Figura 21, donde está marcado el camino que sigue para la comunicación.

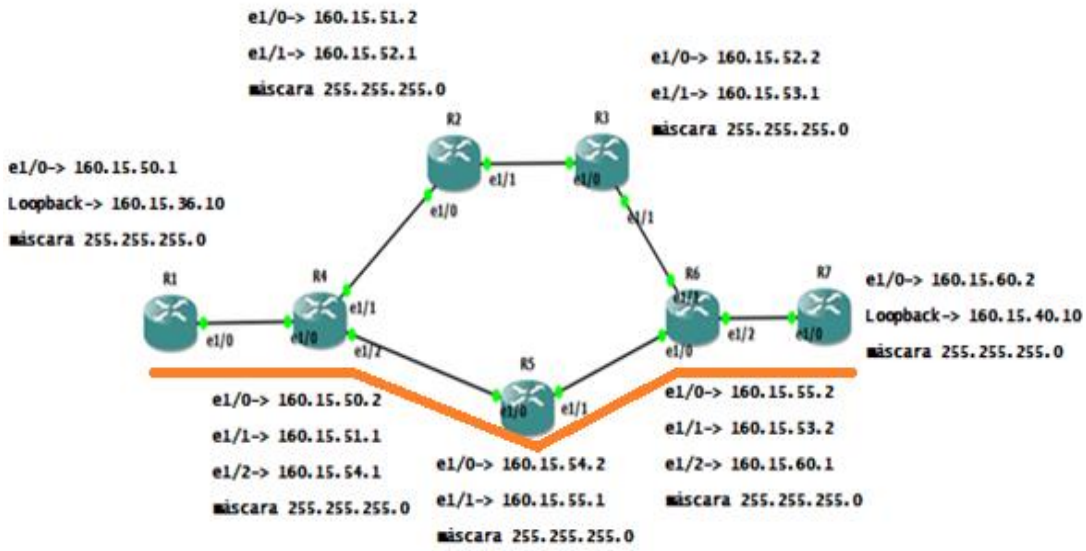

Figura 21. Ejemplo 1. RIP ruta.

Observamos con la línea naranja de la figura, que el propio protocolo elige la ruta a través del router R5, para la comunicación entre R1 y R7. Los saltos que va dando son

R1-R4-R5-R6-R7. RIP es un protocolo de camino más corto que emplea como métrica el número de saltos, por lo que el camino elegido es a través de R5, al ser más corto que el camino alternativo por R2 y R3.

Ahora hay que ver qué ocurre cuando desconectamos uno de los dispositivos del camino seleccionado e intentamos volver a comunicar los extremos. En este caso vamos a poner en *standby* el router R5. Aplicamos otra vez el comando `traceroute`:

```
R1#traceroute 160.15.40.10
Type escape sequence to abort.
Tracing the route to 160.15.40.10
  1 160.15.50.2 288 msec 444 msec 312 msec
  2  *  *  *
  3  *  *  *
  4  *  *  *
  5  *  *  *
  6  *  *  *
  7  *  *  *
  8  *  *  *
  9  *  *  *
 10  *  *  *
 11  *  *  *
 12  *  *  *
 13 160.15.60.2 2584 msec 1948 msec 2836 msec
```

Como se puede observar, la red necesita un tiempo para aprender la nueva ruta y conseguir la comunicación entre los extremos. Es por eso que no nos muestra el camino, aunque el resultado positivo en el salto número 13 nos indica que ya se ha producido la convergencia del protocolo de encaminamiento. Volvemos a realizar la operación, consiguiendo ya la información sobre el nuevo camino:

```
R1#traceroute 160.15.40.10
Type escape sequence to abort.
Tracing the route to 160.15.40.10

  1 160.15.50.2 488 msec 568 msec 448 msec
  2 160.15.51.2 924 msec 572 msec 1024 msec
  3 160.15.52.2 1328 msec 1300 msec 1168 msec
  4 160.15.53.2 1184 msec 1592 msec 2008 msec
  5 160.15.60.2 2224 msec 2520 msec 2348 msec
R1#
```

Se puede observar que ahora el número de saltos con respecto al caso anterior ha cambiado. En este caso tenemos un total de 5 saltos; son los que tienen que dar los paquetes para llegar al destino desde el origen (en la prueba anterior solo daba 4 saltos).

En la Figura 22 vemos que ahora para realizar la comunicación, los paquetes van desde el R1 al R7 pasando por R4-R2-R3-R6.

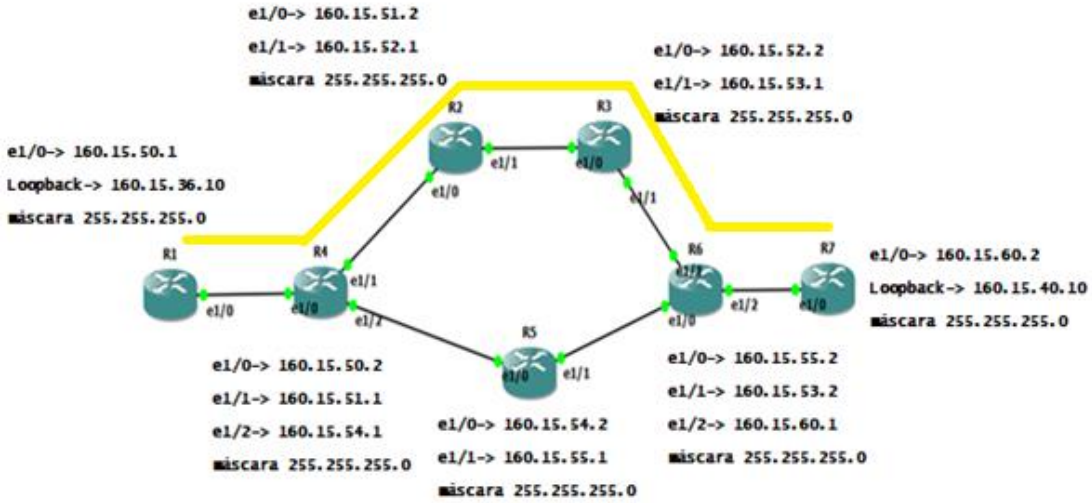

Figura 22. Ejemplo 1. RIP (R5 standby).

Si volvemos a la configuración inicial, activando de nuevo R5, la red tarda un pequeño tiempo en volver a aprender la nueva configuración y converger, pero el camino que elige cuando se realiza el `traceroute` es el camino más corto, ya que es una de las características el del protocolo de encaminamiento RIP, como se ha explicado antes.

A través de consola, podemos aprender más del protocolo, ya que tenemos un sinfín de comandos que nos muestran distintas informaciones sobre RIP y su funcionamiento. Uno de esos comandos es `show ip route xxx.xxx.xxx.xxx`, que nos muestra información sobre el encaminamiento para un destino concreto. En este caso, si lo ejecutamos desde el R1 para el destino R7:

```
R1#show ip route 160.15.40.10
Routing entry for 160.15.40.0/24
  Known via "rip", distance 120, metric 4
  Redistributing via rip
  Last update from 160.15.50.2 on Ethernet1/0, 00:00:05 ago
  Routing Descriptor Blocks:
  * 160.15.50.2, from 160.15.50.2, 00:00:05 ago, via Ethernet1/0
 Route metric is 4, traffic share count is 1
Camino largo
Routing entry for 160.15.40.0/24
```

```
  Known via "rip", distance 120, metric 5
  Redistributing via rip
  Last update from 160.15.50.2 on Ethernet1/0, 00:00:15 ago
  Routing Descriptor Blocks:
  * 160.15.50.2, from 160.15.50.2, 00:00:15 ago, via Ethernet1/0
  Route metric is 5, traffic share count is 1
R1#
```

Con `show ip route` la información que se obtiene es el protocolo de encaminamiento a través del cual se aprendió la ruta, nodo vecino a través del cual se aprendió y la métrica del camino. En este caso, además, nos indica la ruta óptima (se observa que es la ruta que tiene menor métrica, 4 saltos) y otro camino aprendido más largo, que tiene una métrica de 5.

En caso de querer depurar la red, podemos realizar un `debug` del protocolo, que nos va a mostrar toda la información relacionada con el dispositivo en el que estamos trabajando:

```
R1#
*Nov   8 11:59:26.503: RIP: sending v1 update to 255.255.255.255 via
Ethernet1/0 (160.15.50.1)
*Nov  8 11:59:26.503: RIP: build update entries
*Nov  8 11:59:26.503:   subnet 160.15.36.0 metric 1
*Nov   8 11:59:27.447: RIP: sending v1 update to 255.255.255.255 via
Loopback0 (160.15.36.10)
*Nov  8 11:59:27.447: RIP: build update entries
*Nov  8 11:59:27.447:   subnet 160.15.40.0 metric 6
*Nov  8 11:59:27.447:   subnet 160.15.50.0 metric 1
*Nov  8 11:59:27.451:   subnet 160.15.51.0 metric 2
*Nov  8 11:59:27.451:   subnet 160.15.52.0 metric 3
*Nov  8 11:59:27.451:   subnet 160.15.53.0 metric 4
*Nov  8 11:59:27.451:   subnet 160.15.54.0 metric 2
R1#
*Nov  8 11:59:27.455:   subnet 160.15.55.0 metric 5
*Nov  8 11:59:27.455:   subnet 160.15.60.0 metric 5
```

Se observa que se está trabajando con la versión 1 del protocolo RIP, una versión obsoleta del protocolo, cuyo uso está desaconsejado, ya que es *classful* y no permite trabajar con máscaras de longitud variable. Por tanto, aunque en este ejemplo, al trabajar con una única máscara de subred, el uso de la versión 1 no ha dado problemas,

en un caso general el uso de esta versión obsoleta podría producir problemas de encaminamiento.

Ejemplo 2

En este segundo ejemplo del protocolo RIP implementamos la versión 2 del protocolo, la única que debería usarse actualmente, ya que es un protocolo sin clases (*classless*) y admite el uso de máscaras de subred de longitud variable. En la Figura 23 vemos el diseño de la red diseñada para este ejemplo 2 de RIP, que está compuesta por 7 routers y se busca la comunicación entre los extremos, en este caso desde el router R1 al router R7. El direccionamiento IP de todos los equipos se muestra en la Tabla 11.

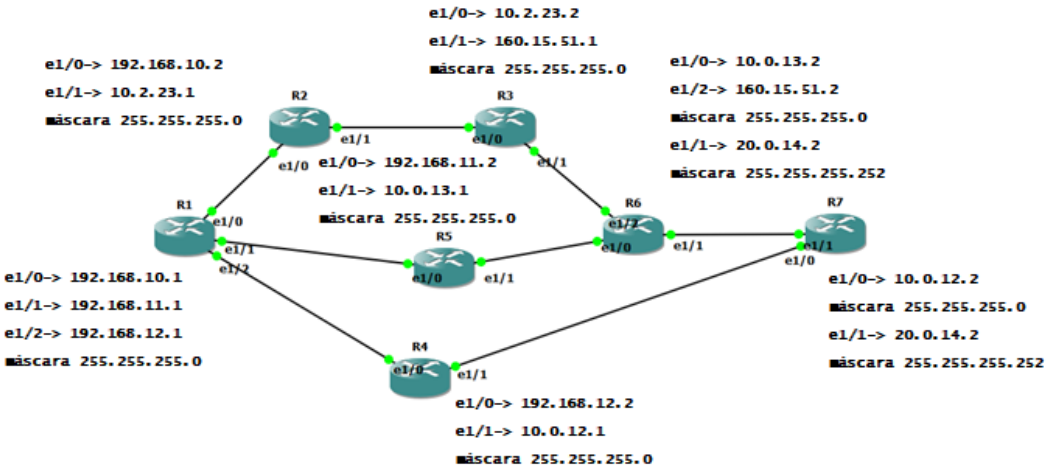

Figura 23. Ejemplo2 RIP V2.

COMPONENTE	DIRECCIÓN IP	MÁSCARA	INTERFAZ
R1 (router)	192.168.10.1	255.255.255.0	e1/0
	192.168.11.1	225.255.255.0	e1/1
	192.168.12.1	255.255.255.0	e1/2
R2 (router)	192.168.10.2	255.255.255.0	e1/0
	10.2.23.1	255.255.255.0	e1/1
R3 (router)	10.2.23.2	255.255.255.0	e1/0
	160.15.51.1	255.255.255.0	e1/1
R4 (router)	192.168.12.2	255.255.255.0	e1/0
	10.0.12.1	255.255.255.0	e1/1
R5 (router)	192.168.11.2	255.255.255.0	e1/0
	10.0.13.1	255.255.255.0	e1/1
R6 (router)	10.0.13.2	255.255.255.0	e1/0
	20.0.14.1	255.255.255.252	e1/2
	160.15.51.2	255.255.255.0	e1/1
R7 (router)	10.0.12.2	255.255.255.0	e1/0
	20.0.14.2	255.255.255.252	e1/1

Tabla 11. Ejemplo 2- RIP V2 configuración.

Como se puede observar, después de indicar el tipo de protocolo con el comando `router rip`, el siguiente paso es indicar qué versión del protocolo vamos a usar; en esta configuración sería RIPv2, para lo que introducimos la sentencia `version 2`, y a continuación especificamos las redes a las que el router R2 está conectado directamente: `network 192.168.10.0` y `network 10.2.23.0`.El comando `no auto-summary` evita que haga un resumen automático de la red.

El siguiente paso es la realización de pruebas con las que veamos que el protocolo RIP V2 funciona según se ha explicado en la parte teórica de la asignatura. Vamos a comprobar qué camino siguen los paquetes desde los extremos más alejados del diseño de la red, en este caso desde el router R1 al router R7. Hacemos un `traceroute` a la dirección de *loopback* del R7 (160.15.40.10):

```
R1#traceroute 160.15.40.10
Type escape sequence to abort.
Tracing the route to 160.15.40.10
1 192.168.12.2 664 msec 696 msec 524 msec
2 10.0.12.2 716 msec 1144 msec 1092 msec
```

Comprobamos con el comando que la ruta que sigue para comunicarse los extremos va desde el propio R1 pasando por el R4 hasta llegar al destino R7. La ruta seguida es la más corta, el propio protocolo busca y elige el camino óptimo hasta el destino. Si

desconectamos el router R4 para ver la adaptación del protocolo al cambio, despareciendo la opción mejor para llegar al R7.

```
R1#traceroute 160.15.40.10
Type escape sequence to abort.
Tracing the route to 160.15.40.10
1 192.168.11.2 320 msec 320 msec 304 msec
2 10.0.13.2 952 msec 728 msec 928 msec
3 20.0.14.2 1236 msec 1884 msec 1496 msec
```

El protocolo de encaminamiento ahora elige el nuevo camino más corto, en el que va desde R1 pasando por R5, R6 y llegando al R7. Para finalizar las pruebas, desconectamos también el R5, teniendo ahora R4 y R5 en un estado de Stand-By, y volvemos a ver como RIPv2 es capaz de aprender una nueva ruta para conseguir comunicarse entre sus extremos.

```
R1#traceroute 160.15.40.10
Type escape sequence to abort.
Tracing the route to 160.15.40.10
1  *  *  *
2  *  *  *
3  *  *  *
4  *  *  *
5  *  *  *
6  *  *  *
7  *  *  *
8  *  *  *
9  *  *  *
10  *  *  *
11  *  *  *
12  *  *  *
13  *  *  *
14  *  *  *
15  *  *  *
16  *  *  *
17  *  *  *
18  *  *  *
19  *  *  *
```

```
20  *
```

En este último caso, la actualización de la ruta tarda más que en los supuestos anteriores, llegando a superar el tiempo máximo. Se repite la operación y el resultado conseguido es el siguiente:

```
R1#traceroute 160.15.40.10
Type escape sequence to abort.
Tracing the route to 160.15.40.10
1 192.168.10.2 512 msec 388 msec 364 msec
2 10.2.23.2 716 msec 672 msec 684 msec
3 160.15.51.2 1188 msec 1188 msec 1284 msec
4 20.0.14.2 1464 msec 1480 msec 1684 msec
```

De todas las rutas que se han ido viendo a lo largo de los casos en el ejemplo, esta es la más larga; va desde R1, pasando por R2, R3, R6 y finaliza en R7. Hemos comprobado como el protocolo se va actualizando y se adapta a los cambios que se realizan en la red, en un tiempo relativamente corto.

Entre los comandos que nos ofrecen los routers CISCO sobre el protocolo RIP, podemos además comprobar la base de datos de información con la que trabaja en cada uno de los instantes. Introduciendo en la línea de comando la instrucción `show ip rip database`, veremos toda la información relativa a rutas RIP de la que dispone el router:

```
R1#show ip rip database
10.0.0.0/8    auto-summary
10.0.12.0/24
    [4] via 192.168.10.2, 00:00:26, Ethernet1/0
10.0.13.0/24
    [3] via 192.168.10.2, 00:00:26, Ethernet1/0
10.2.23.0/24
    [1] via 192.168.10.2, 00:00:26, Ethernet1/0
20.0.0.0/8    auto-summary
20.0.14.0/30
    [3] via 192.168.10.2, 00:00:26, Ethernet1/0
160.15.0.0/16    auto-summary
160.15.35.0/24    directly connected, Loopback0
160.15.40.0/24
    [4] via 192.168.10.2, 00:00:26, Ethernet1/0
```

```
160.15.51.0/24
    [2] via 192.168.10.2, 00:00:26, Ethernet1/0
192.168.10.0/24    auto-summary
192.168.10.0/24    directly connected, Ethernet1/0
192.168.11.0/24    auto-summary
192.168.11.0/24    directly connected, Ethernet1/1
192.168.12.0/24    auto-summary
192.168.12.0/24    directly connected, Ethernet1/2
R1#
```

En los dos ejemplos se ve cómo trabaja el protocolo RIP en las dos versiones v1 y v2, siendo la versión v1 una versión obsoleta del protocolo que no debería emplearse en las redes actuales. Aquí solo se exponen una mínima cantidad de comandos que se pueden usar, pero en cualquier de los manuales existente podemos ver toda la galería de comandos y la funciones que realizan.

La siguiente práctica que se realiza en la asignatura está relacionada con el protocolo de encaminamiento OSPF. En los ejemplos siguientes veremos sus características y cómo es su funcionamiento.

3.6.3 PROTOCOLO DE ENCAMINAMIENTO OSPF

En esta práctica relacionada con el encaminamiento, el objetivo es que el alumno descubra cómo es el funcionamiento del protocolo OSPF. La metodología que se sigue es la misma que en la práctica de RIP; el diseño de la topología de la red, la configuración de los elementos que intervienen en la red y la realización de las pruebas necesarias para comprobar los conocimientos impartidos en las clases de teoría.

Ejemplo 1

La topología de este primer ejemplo de OSPF es básica, como se puede ver en la Figura 24, un PC conectado a un switch y dos routers conectados entre sí, a través de un enlace serial.

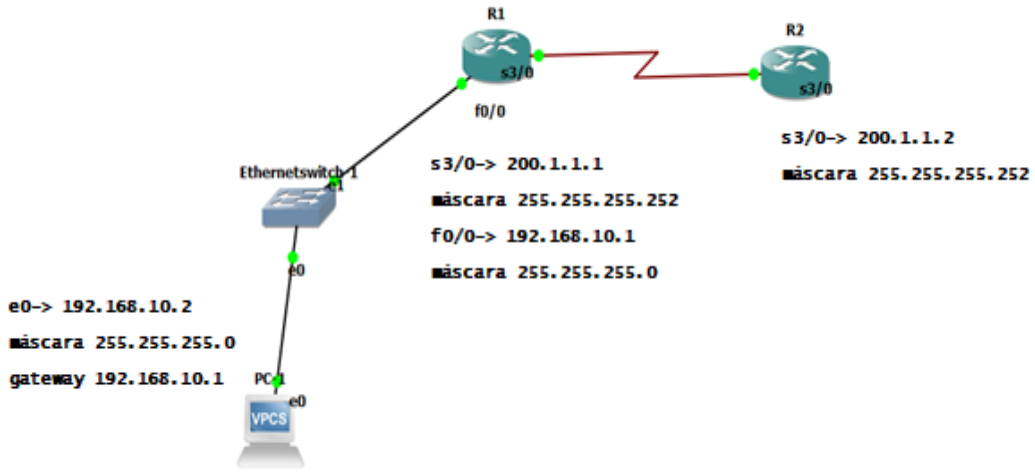

Figura 24. Ejemplo 1. OSPF.

La configuración de los routers definida en la tabla 12, es la misma que se ha seguido en los ejemplos anteriores de las prácticas.

COMPONENTE	DIRECCIÓN IP	MÁSCARA	INTERFAZ
R1 (router)	200.1.1.1	255.255.255.252	S3/0
	192.168.10.1	225.255.255.0	f0/0
R2 (router)	200.1.1.2	255.255.255.252	S3/0

Tabla 12. Ejemplo 1- OSPF configuración.

Una vez efectuada la configuración básica de los elementos principales de la topología, hacemos la configuración específica del protocolo OSPF:

```
R2#
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#router ospf 1
R2(config-router)#net 200.1.1.0 0.0.0.3 area 0
R2(config-router)#net 192.168.10.0 0.0.0.255 area 0
R2(config-router)#exit
R2(config)#exit
R2#wr
Building configuration...
```

Esta es la configuración del protocolo básica. Con el comando `router ospf 1` le indicamos que el protocolo que se va a usar es OSPF. A continuación, se le indican la

dirección de cada red y la inversa de la máscara que se utiliza y el área al que pertenece dicha red. Como siempre, es necesario guardar la configuración con `wr`.

Seguimos con la configuración del router R1:

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#net 200.1.1.0 0.0.0.3 area 0
R1(config-router)#net
*Dec 14 19:59:35.495: %OSPF-5-ADJCHG: Process 1, Nbr 200.1.1.2 on Serial3/0 from LOADING to FULL, Loading Done
R1(config-router)#net 192.168.10.0 0.0.0.255 area 0
R1(config-router)#exit
R1(config)#exit
R1#wr
Building configuration...
[OK]
```

Para ver que funciona correctamente, nos queda configurar el PC:

```
PC1: 192.168.10.2 255.255.255.0 gateway 192.168.10.1
```

Con un ping demostramos que la configuración es correcta, y vemos cómo existe comunicación entre los extremos.

```
PC1> ping 200.1.1.2
84 bytes from 200.1.1.2 icmp_seq=1 ttl=254 time=18.001 ms
84 bytes from 200.1.1.2 icmp_seq=2 ttl=254 time=53.003 ms
84 bytes from 200.1.1.2 icmp_seq=3 ttl=254 time=29.001 ms
84 bytes from 200.1.1.2 icmp_seq=4 ttl=254 time=55.004 ms
84 bytes from 200.1.1.2 icmp_seq=5 ttl=254 time=26.001 ms
```

Este es un ejemplo básico de cómo se realiza la configuración del protocolo OSPF.

Ejemplo 2

En este segundo escenario de OSPF vamos a complicar un poco más la configuración y vamos a aumentar el número de áreas, haciendo un número mayor de pruebas del protocolo.

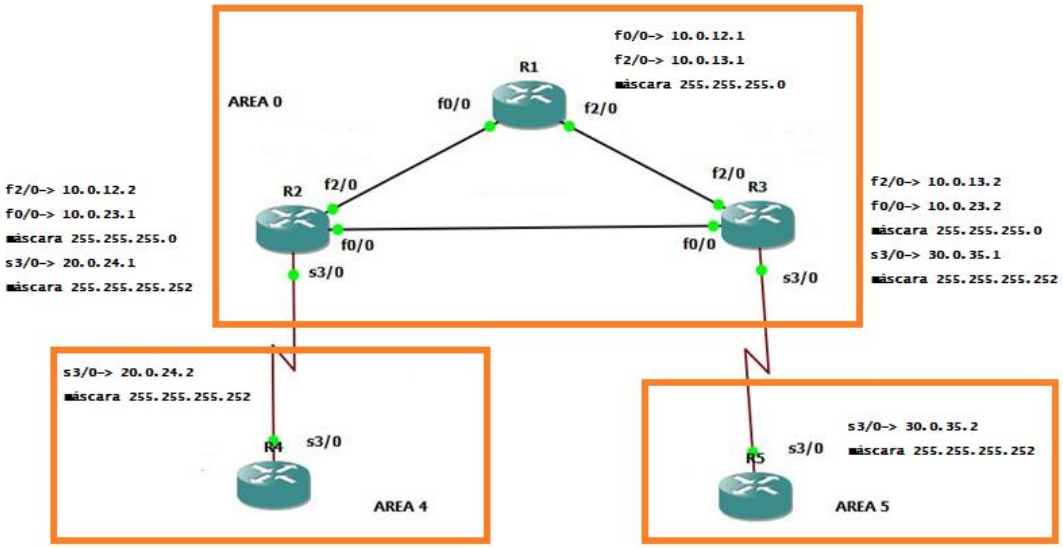

Figura 25. Ejemplo 2. OSPF.

En la topología de OSPF del ejemplo 2, que se ve en la Figura 25, podemos observar que está compuesta por cinco routers, en la que se puede ver tres áreas definidas (área 0, área 4, y área 5). En la tabla 13 vemos la dirección y máscara asignada a cada interfaz.

COMPONENTE	DIRECCIÓN IP	MÁSCARA	INTERFAZ
R1 (router)	10.0.12.1	255.255.255.0	f0/0
	10.0.13.1	225.255.255.0	f2/0
R2 (router)	10.0.12.2	255.255.255.0	f2/0
	10.0.23.1	255.255.255.0	f0/0
	20.0.24.1	255.255.255.252	s3/0
R3 (router)	10.0.13.2	255.255.255.0	f2/0
	10.0.23.2	255.255.255.0	f0/0
	30.0.35.1	255.255.255.252	s3/0
R4 (router)	20.0.24.2	255.255.255.252	s3/0
R5 (router)	30.0.35,2	255.255.255.252	s3/0

Tabla 13. Ejemplo 2- OSPF configuración.

La configuración de la topología es la misma que se ha realizado en el ejemplo 1 de OSPF. Lo primero es configurar el router con las direcciones y, una vez realizado esto, continuamos con el protocolo OSPF. La diferencia de este ejemplo es que tenemos interfaces FastEthernet y Serial:

```
R1(config)#INT F1/0 // en el caso de ser Fast definir el interfaz a usar
R1(config)#INT S1/0 // en el caso de ser Serial definir el interfaz a usar
```

OSPF tiene unos parámetros sencillos de definir. Tenemos que incluir las áreas en la que están las redes, así como, para cada red a la que se conecta el router, la dirección de la red y de máscara inversa a la que estemos usando en la configuración. A continuación, se muestra la configuración de un router del área 0:

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#net 10.0.12.0 0.0.0.255 area 0
R1(config-router)#net 10.0.13.0 0.0.0.255 area 0
R1(config-router)#exit
R1(config)#exit
R1#wr
Building configuration...
[OK]
```

Lo mismo para otro router que esté en otra área:

```
R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#router ospf 1
R4(config-router)#net 20.0.24.2 0.0.0.3 area 4
R4(config-router)#exit
R4(config)#exit
R4#wr
Building configuration...
[OK]
```

Para los routers de borde de área (R2 y R3) la configuración sería similar, teniendo cuidado de indicar el área correcta a la que pertenecen las distintas redes a las que se conectan.

Empezamos con las pruebas, y lo primero es testear si existe conectividad entre los extremos R4 y R5, y comprobar cuál es el camino que sigue para conseguirlo.

```
R4#TRACEROUTE 30.0.35.2
```

```
Type escape sequence to abort.
Tracing the route to 30.0.35.2
  1 20.0.24.1 36 msec 40 msec 48 msec
  2 10.0.23.2 60 msec 72 msec 40 msec
  3 30.0.35.2 68 msec 92 msec 88 msec
R4#
```

Vemos que llegar de un punto a otro, nos cuesta 3 saltos. Va del origen al destino pasando por los routers R2 y R3. Una vez que sabemos que es correcto, observamos la tabla de encaminamiento y buscamos la información correspondiente a OSPF:

```
R4#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
     20.0.0.0/30 is subnetted, 1 subnets
C       20.0.24.0 is directly connected, Serial3/0
     10.0.0.0/24 is subnetted, 3 subnets
O IA    10.0.12.0 [110/65] via 20.0.24.1, 00:11:07, Serial3/0
O IA    10.0.13.0 [110/66] via 20.0.24.1, 00:10:57, Serial3/0
O IA    10.0.23.0 [110/65] via 20.0.24.1, 00:11:07, Serial3/0
     30.0.0.0/30 is subnetted, 1 subnets
O IA    30.0.35.0 [110/129] via 20.0.24.1, 00:11:12, Serial3/0
R4#
```

Con el comando `show ip route` obtenemos la información de la tabla de encaminamiento del router, comprobando las rutas y cómo se han obtenido.

En la parte de la izquierda, podemos ver unas letras, al inicio del comando nos indica que es lo que significa cada una de ellas.

- C- indica que está conectado directamente a esa IP
- O- protocolo que se ejecuta, en este caso es OSPF

- IA- también relacionado con el protocolo, pero indica que OSPF Inter-Area, donde las rutas de un área a otra son un router de Borde de Area.

Si ejecutamos el comando en el resto de los routers, conseguimos información similar, por ejemplo, el R3:

```
R3#SHOW IP ROUTE
Gateway of last resort is not set
     20.0.0.0/30 is subnetted, 1 subnets
O IA    20.0.24.0 [110/65] via 10.0.23.1, 00:12:29, FastEthernet0/0
     10.0.0.0/24 is subnetted, 3 subnets
O       10.0.12.0 [110/2] via 10.0.23.1, 00:12:29, FastEthernet0/0
                  [110/2] via 10.0.13.1, 00:12:29, FastEthernet2/0
C       10.0.13.0 is directly connected, FastEthernet2/0
C       10.0.23.0 is directly connected, FastEthernet0/0
     30.0.0.0/30 is subnetted, 1 subnets
C       30.0.35.0 is directly connected, Serial3/0
R3#
```

Siguiendo con el comando show ip route xxx.xxx.xxx.xxx, podemos ver, para el destino especificado, la ruta aprendida, mediante qué protocolo y si es Inter-Área, la distancia administrativa del protocolo (prioridad de este frente a otros) y la métrica del camino:

```
R1#show ip route 30.0.35.2
Routing entry for 30.0.35.0/30
  Known via "ospf 1", distance 110, metric 65, type inter area
  Last update from 10.0.13.2 on FastEthernet2/0, 00:20:49 ago
  Routing Descriptor Blocks:
  * 10.0.13.2, from 30.0.35.1, 00:20:49 ago, via FastEthernet2/0
      Route metric is 65, traffic share count is 1
```

La salida anterior nos muestra información sobre la ruta de R1 a R5: estamos usando un protocolo de encaminamiento tipo OSPF, que la *distancia administrativa* o prioridad del protocolo es de 110 (valor por defecto si no se define nada en la configuración), que el camino tiene una *métrica* de 65 y, por último, que es de tipo *inter-área*.

Si queremos saber los vecinos OSPF que tienen cada uno de ellos, y si se pueden intercambiar mensajes usaremos:

```
R1#SHOW IP OSPF NEIGHBOR
Neighbor ID     Pri   State           Dead Time   Address         Interface
30.0.35.1         1   FULL/DR         00:00:33    10.0.13.2
FastEthernet2/0
20.0.24.1         1   FULL/DR         00:00:32    10.0.12.2
FastEthernet0/0
R1#
R2#SHOW IP OSPF NEIGHBOR
Neighbor ID     Pri   State           Dead Time   Address         Interface
30.0.35.1         1   FULL/DR         00:00:36    10.0.23.2
FastEthernet0/0
10.0.13.1         1   FULL/BDR        00:00:34    10.0.12.1
FastEthernet2/0
20.0.24.2         0   FULL/  -        00:00:30    20.0.24.2       Serial3/0
R2#
R3#SHOW IP OSPF NEIGHBOR
Neighbor ID     Pri   State           Dead Time   Address         Interface
20.0.24.1         1   FULL/BDR        00:00:37    10.0.23.1
FastEthernet0/0
10.0.13.1         1   FULL/BDR        00:00:37    10.0.13.1
FastEthernet2/0
30.0.35.2         0   FULL/  -        00:00:36    30.0.35.2       Serial3/0
R3#
R4#SHOW IP OSPF NEIGHBOR
Neighbor ID     Pri   State           Dead Time   Address         Interface
20.0.24.1         0   FULL/  -        00:00:36    20.0.24.1       Serial3/0
R4#
R5#SHOW IP OSPF NEIGHBOR
Neighbor ID     Pri   State           Dead Time   Address         Interface
30.0.35.1         0   FULL/  -        00:00:31    30.0.35.1       Serial3/0
R5#
```

Comprobamos las diferentes adyacencias que tienen cada uno de los elementos de la topología. De normal tienen que tener el estado (state) con el valor de FULL/ - es el estado general del router en OSPF. Significa que tiene plena adyacencia con el vecino y pueden intercambiar paquetes de saludo, actualizaciones, consultas…. Además, para interfaces multiacceso de difusión, se indica si el router actúa como Designated Router (DR) o Backup DR (BDR). En enlaces serial (punto a punto) no hay concepto de DR o BDR.

Siguiendo con comandos OSPF, el siguiente nos muestra información sobre las rutas existentes a los Routers de Borde de Área:

```
R1#SHOW IP OSPF BORDER-ROUTERS
OSPF Process 1 internal Routing Table
Codes: i - Intra-area route, I - Inter-area route
i 20.0.24.1 [1] via 10.0.12.2, FastEthernet0/0, ABR, Area 0, SPF 3
i 30.0.35.1 [1] via 10.0.13.2, FastEthernet2/0, ABR, Area 0, SPF 3
R1#
R2#SHOW IP OSPF BORDER-ROUTERS
OSPF Process 1 internal Routing Table
i 30.0.35.1 [1] via 10.0.23.2, FastEthernet0/0, ABR, Area 0, SPF 7
R2#
R3#SHOW IP OSPF BORDER-ROUTERS
OSPF Process 1 internal Routing Table
i 20.0.24.1 [1] via 10.0.23.1, FastEthernet0/0, ABR, Area 0, SPF 4
R3#
R4#SHOW IP OSPF BORDER-ROUTERS
OSPF Process 1 internal Routing Table
i 20.0.24.1 [64] via 20.0.24.1, Serial3/0, ABR, Area 4, SPF 2
R4#
R5#SHOW IP OSPF BORDER-ROUTERS
i 30.0.35.1 [64] via 30.0.35.1, Serial3/0, ABR, Area 5, SPF 3
R5#
```

Para cada router, tenemos la información de cómo llegar al router o routers del borde del área o áreas a las que pertenece.

Un ejercicio relacionado con el protocolo es ver cómo se adapta la red a las distintas circunstancias que puedan pasar. Suspendemos temporalmente el enlace entre los routers R2 y R3, como se ve en la figura 26.

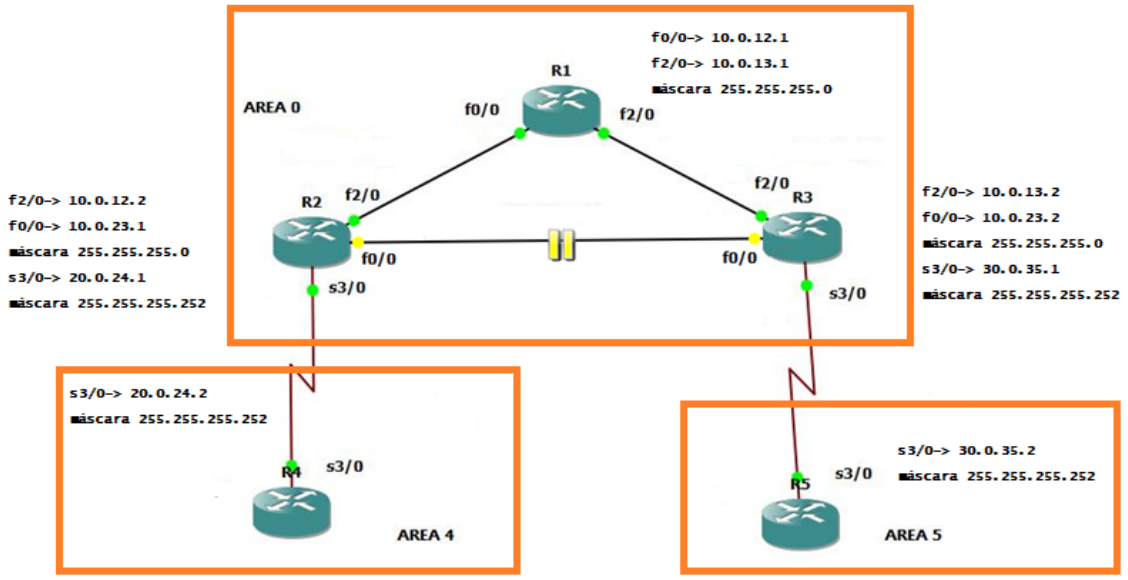

Figura 26. Ejemplo 2. OSPF stand-by.

Y vemos cuál es el camino que hay desde el R4 al R5 como hicimos al principio de las pruebas:

```
R4#TRACEROUTE 30.0.35.2
Type escape sequence to abort.
Tracing the route to 30.0.35.2
  1 20.0.24.1 48 msec 68 msec 32 msec
  2 10.0.12.1 40 msec 40 msec 48 msec
  3 10.0.13.2 28 msec 92 msec 80 msec
  4 30.0.35.2 72 msec 100 msec 44 msec
```

Ahora el camino está establecido a través del R1. El camino es más largo que en el caso anterior; pero vamos a ver qué ocurre si activamos el enlace que teníamos suspendido temporalmente y aumentamos el coste de este. Para ello, tenemos que volver a configurar el router e indicar el nuevo coste del enlace. OSPF recalculará los costes de los caminos teniendo en cuenta los nuevos costes de los enlaces, lo cual puede suponer que el nuevo camino más corto no sea el de menor número de saltos.

```
R2(config)#int f0/0
R2(config-if)#ip ospf cost 500
R2(config-if)#exit
R2(config)#exit
R2#wr
```

Con este nuevo dato repetimos la operación de traceroute y comprobamos que el aumento del coste del enlace hace que ruta elegida para llegar al extremo R5 sea por la ruta más larga en cuanto a número de saltos, que ahora resulta la óptima.

```
R4#TRACEROUTE 30.0.35.2
Type escape sequence to abort.
Tracing the route to 30.0.35.2
  1 20.0.24.1 36 msec 32 msec 60 msec
  2 10.0.12.1 60 msec 44 msec 44 msec
  3 10.0.13.2 80 msec 100 msec 72 msec
  4 30.0.35.2 128 msec 52 msec 112 msec
```

OSPF permite también tener en cuenta para calcular los mejores caminos, parámetros como el ancho de banda del enlace, de forma que podemos conseguir mejores resultados en nuestra configuración de la red. Si queremos también cambiar el valor del ancho de banda del enlace usamos el comando `bandwidth`, estableciendo el valor que queramos:

```
R4#CONF T
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#INTERFACE S3/0
R4(config-if)#bandwidth 28
R4(config-if)#exit
R4(config)#exit
R4#wr
Building configuration...
R4#SHOW INTERFACE SERIAL 3/0 | INCLUDE BW
  MTU 1500 bytes, BW 28 Kbit, DLY 20000 usec,
R4#
```

En la primera parte está la configuración del cambio de ancho de banda y con el comando `show` comprobamos el resultado que hemos conseguido.

Para completar la práctica relacionada con el protocolo OSPF, se puede ampliar los datos con el comando:

```
R2#SHOW IP OSPF INTERFACE
FastEthernet0/0 is up, line protocol is up
  Internet Address 10.0.23.1/24, Area 0
```

```
  Process ID 1, Router ID 20.0.24.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 30.0.35.1, Interface address 10.0.23.2
  Backup Designated router (ID) 20.0.24.1, Interface address 10.0.23.1
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:05
  Supports Link-local Signaling (LLS)
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 30.0.35.1  (Designated Router)
  Suppress hello for 0 neighbor(s)
FastEthernet2/0 is up, line protocol is up
  Internet Address 10.0.12.2/24, Area 0
  Process ID 1, Router ID 20.0.24.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 20.0.24.1, Interface address 10.0.12.2
  Backup Designated router (ID) 10.0.13.1, Interface address 10.0.12.1
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:06
  Supports Link-local Signaling (LLS)
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.0.13.1  (Backup Designated Router)
  Suppress hello for 0 neighbor(s)
Serial3/0 is up, line protocol is up
  Internet Address 20.0.24.1/30, Area 4
  Process ID 1, Router ID 20.0.24.1, Network Type POINT_TO_POINT, Cost:
64
  Transmit Delay is 1 sec, State POINT_TO_POINT
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:01
```

```
  Supports Link-local Signaling (LLS)
  Index 1/3, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 3
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 20.0.24.2
  Suppress hello for 0 neighbor(s)
```

Podemos ver toda la información de las interfaces del router R2 en relación con el protocolo OSPF, nos indica los parámetros de los temporizadores *Hello* y *Router Dead Interval* para el intercambio de mensajes *hello* entre router vecinos; también indica las *áreas* a las que pertenece cada interfaz y el *coste* de cada enlace. También se incluye información general del enlace y las direcciones IP de red que tiene configurado. Es un comando que nos da la información general y del que se pueden obtener muchos datos.

Otro comando que nos ofrece también información fundamental sobre OSPF es el relacionado con la base de datos topológica:

```
R4#SHOW IP OSPF DATABASE
            OSPF Router with ID (20.0.24.2) (Process ID 1)
                Router Link States (Area 4)
Link ID          ADV Router       Age          Seq#        Checksum Link
count
20.0.24.1       20.0.24.1       525         0x80000005 0x008C34 2
20.0.24.2       20.0.24.2       518         0x80000003 0x008140 2
                Summary Net Link States (Area 4)
Link ID         ADV Router      Age         Seq#       Checksum
10.0.12.0       20.0.24.1       309         0x80000009 0x0019D7
10.0.13.0       20.0.24.1       314         0x80000006 0x001ED3
10.0.23.0       20.0.24.1       260         0x80000010 0x00A537
30.0.35.0       20.0.24.1       314         0x80000007 0x0094F4
R4#
R1#SHOW IP OSPF DATABASE
            OSPF Router with ID (10.0.13.1) (Process ID 1)
                Router Link States (Area 0)
Link ID          ADV Router       Age          Seq#        Checksum Link
count
10.0.13.1       10.0.13.1       370         0x80000006 0x008805 2
20.0.24.1       20.0.24.1       199         0x8000000E 0x00CB16 2
30.0.35.1       30.0.35.1       319         0x80000007 0x0075A9 2
```

```
                Net Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum
10.0.12.2       20.0.24.1       371         0x80000001 0x00B6F8
10.0.13.2       30.0.35.1       581         0x80000002 0x00ABD7
10.0.23.2       30.0.35.1       319         0x80000001 0x003D28
                Summary Net Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum
20.0.24.0       20.0.24.1       576         0x80000002 0x00861F
30.0.35.0       30.0.35.1       581         0x80000002 0x00E298
R1#
```

Ofrece información sobre los anuncios de estado de enlace recibidos que contienen la información sobre los enlaces y permiten reconstruir la topología OSPF de la red.

Con estos dos ejemplos se ve el funcionamiento del protocolo de encaminamiento OSPF. Se puede ver cómo se configura de forma básica, cómo se definen las áreas, así como comprobar qué elementos están en el borde del área o dentro del área. También se observa qué sucede si cambiamos el coste de un enlace, comprobando cómo OSPF se adapta al cambio eligiendo a la ruta óptima en función de diversos parámetros, en vez de limitarse a la longitud de los caminos en número de saltos.

Los 4 ejemplos realizados anteriormente, sirven para afianzar la parte de teoría de la asignatura relacionada con el encaminamiento y en especial con los protocolos RIP en sus 2 versiones y el protocolo OSPF.

3.6.4 VLAN

Esta última practica está relacionada con analizar con diseño de red teniendo en cuenta varios escenarios.

- **Ejemplo 1**

Partimos del diseño de la Figura 27, donde los elementos que participan son dos switches definidos como ESW1 y ESW2 a los que se le conectamos 2 elementos finales cada uno de ellos pertenecen a una VLAN distinta (VLAN 10 y VLAN 20).

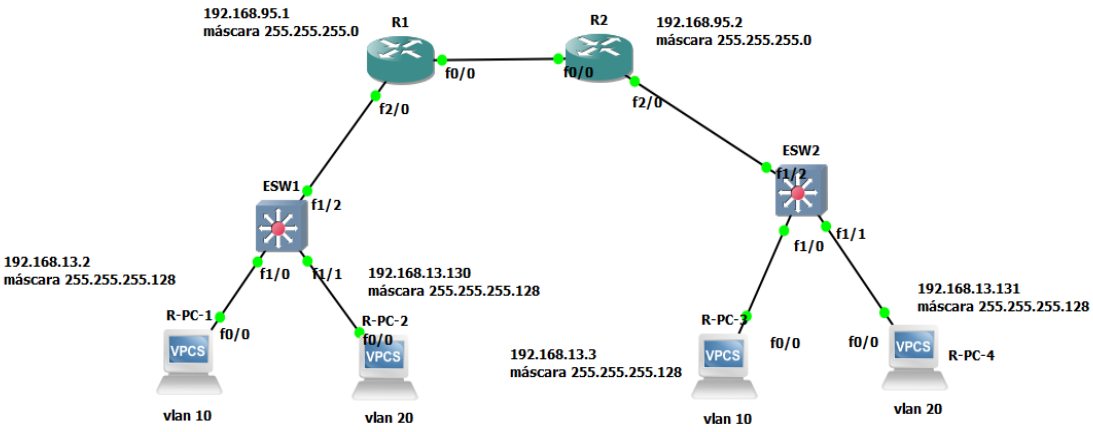

Figura 27. Ejemplo 1. VLAN.

La Tabla 14 muestra la información necesaria para la configuración de los distintos dispositivos que intervienen en el esquema.

COMPONENTE	DIRECCIÓN IP	MÁSCARA	INTERFAZ
R1 (router)	192.168.95.1	255.255.255.0	F0/0
	192.168.13.1	255.255.255.128	F2/0
R2 (router)	192.168.95.2	255.255.255.0	F0/0
	192.168.13.129	255.255.255.128	F2/0
R-PC-1	192.168.13.2	255.255.255.128	F0/0
R-PC-2	192.168.13.130	255.255.255.128	F0/0
R-PC-3	192.168.13.3	255.255.255.128	F0/0
R-PC-4	192.168.13.131	255.255.255.128	F0/0

Tabla 14. Ejemplo 1- VLAN configuración.

Los switches usados son dos routers configurados como dichos dispositivos, como se ve en el Apartado 3.5 correspondiente a Topologías Básicas (Primeros Pasos con GNS3). Empezamos preparando el switch para la VLAN.

```
ESW1#enable
ESW1#vlan database
```

El comando `vlan database` nos permite indicar que vamos a diseñar un esquema en el que van a participar VLANs, disponiendo los recursos necesarios; toda la información se almacena en una memoria flash. Continuamos con la configuración:

```
ESW1(vlan)#vlan 10 name sala1
VLAN 10 modified:
    Name: sala1
ESW1(vlan)#vlan 20 name despacho
VLAN 20 modified:
    Name: despacho
ESW1(vlan)#vtp server
Device mode already VTP SERVER.
ESW1(vlan)#vtp domain proyecto
Changing VTP domain name from redes1 to proyecto
ESW1(vlan)#proyecto
ESW1(vlan)#vtp pass proyecto
Setting device VLAN database password to proyecto.
ESW1(vlan)#exit
APPLY completed.
```

Definimos dos VLANs, a las que damos los nombre de *sala1* y *despacho,* que son las que van a pasar por el switch *ESW1.* También declaramos que `vtp server` para indicar que trabajamos en el nivel 2 (protocolo de mensaje), que va a ayudar a la configuración y administración de redes, donde todas ellas comparten el mismo dominio, en este caso declarado con el nombre *proyecto1.* Como configuración adicional le podemos dar una contraseña con el comando `vtp pass`. Comprobamos que está todo correcto:

```
ESW1#sh vtp s
VTP Version                     : 2
Configuration Revision          : 1
Maximum VLANs supported locally : 36
Number of existing VLANs        : 7
VTP Operating Mode              : Server
VTP Domain Name                 : proyecto1
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x56 0xDB 0x1D 0x8D 0x74 0xB3 0x4D 0x92
Configuration last modified by 0.0.0.0 at 3-1-02 00:03:06
Local updater ID is 0.0.0.0 (no valid interface found)
ESW1#
```

Nos muestra toda la información relacionada con la configuración de `vtp server` indicada en la configuración. Lo siguiente es la configuración del switch con los elementos que tiene conectados.

```
ESW1#
ESW1#CONF T
Enter configuration commands, one per line.
ESW1(config)#INT F1/2
ESW1(config-if)#SW MO TR
ESW1(config-if)#speed 100
ESW1(config-if)#duplex full
ESW1(config-if)#no sh
ESW1(config-if)#exit
```

Declaramos la interfaz conectada al router R1, este enlace está definido como modo troncal o **trunk**; este tipo de enlace es un enlace punto-a-punto entre dos dispositivos de red que transporta más de una VLAN, por lo que el tráfico debe ir etiquetado con el identificador de la VLAN correspondiente. En nuestro ejemplo queremos que pase por el enlace troncal, tráfico de las VLAN 10 y VLAN 20. Este modo de configuración de puerto se puede hacer entre switch y switch o entre switch y router. Con el comando `show` vemos como queda esta configuración:

```
ESW1#SH INT T
Port      Mode         Encapsulation  Status        Native vlan
Fa1/2     on           802.1q         trunking      1
Port      Vlans allowed on trunk
Fa1/2     1-4094
Port      Vlans allowed and active in management domain
Fa1/2     1,10,20
Port      Vlans in spanning tree forwarding state and not pruned
Fa1/2     1,10,20
```

Y las conexiones con los elementos finales del diseño.

```
ESW1#CONF T
Enter configuration commands, one per line.  End with CNTL/Z.
ESW1(config)#int f1/3
ESW1(config-if)#sw mo acc
```

```
ESW1(config-if)#sw acc vlan 10
ESW1(config-if)#speed 100
ESW1(config-if)#duplex full
ESW1(config-if)#no sh
ESW1(config-if)#exit
ESW1(config)#exit
ESW1#
ESW1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ESW1(config)#int f1/1
ESW1(config-if)#sw mo acc
ESW1(config-if)#sw acc vlan 20
ESW1(config-if)#speed 100
ESW1(config-if)#duplex full
ESW1(config-if)#no sh
ESW1(config-if)#exit
ESW1(config)#exit
ESW1#
```

Los puertos conectados con los elementos finales de la red, esta declarados como **mode access**. Este modo *Access* sirve para llevar tráfico de una única VLAN, por lo que va sin etiquetar. Se suele configurar cuando se conecta con elementos finales de la red, que pertenecen a una única VLAN. Comprobamos que lo hemos configurado correctamente:

```
ESW1#
VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
---
1    default                           active    Fa1/4, Fa1/5, Fa1/6, Fa1/7
                                                 Fa1/8, Fa1/9, Fa1/10, Fa1/11
                                                 Fa1/12,    Fa1/13,    Fa1/14,
Fa1/15
10   sala1                             active    Fa1/0, Fa1/3
20   despacho                          active    Fa1/1
1002 fddi-default                      act/unsup
1003 token-ring-default                act/unsup
1004 fddinet-default                   act/unsup
1005 trnet-default                     act/unsup
```

Repetimos las mismas operaciones para el ESW2, el switch 2, declaramos la base de datos de las VLANs, las definimos y configuramos para cada una de sus interfaces según se ha realizado anteriormente con switch 1.

El siguiente paso es la configuración de los routers, R1 y R2, donde en este caso tenemos que configurar subinterfaces.

```
R1#conf t
Enter configuration commands, one per line.  End with CNT
R1(config)#int f2/0.10
R1(config-subif)#des sala1
R1(config-subif)#enc dot1q 10
R1(config-subif)#ip add 192.168.13.1 255.255.255.128
R1(config-subif)#speed 100
R1(config-subif)#exit
R1(config)#conf f0//0
R1(config)#exit
R1#
R1#conf t
Enter configuration commands, one per line.  End with CNT        L/Z.
R1(config)#int f0/0
R1(config-if)#ip add 192.168.95.1 255.255.255.0
R1(config-if)#no sh
R1(config-if)#speed 100
R1(config-if)#duplex full
R1(config-if)#exit
R1#
```

Declaramos la interfaz, y el tráfico que va a circular por ella, el de la VLAN 10, describimos la VLAN y definimos el protocolo de encapsulamiento que vamos a utilizar, en este caso `enc dot1q 10`. 802.1Q es un protocolo de etiquetado que modifica el paquete de información que estamos enviando, pudiendo combinarse con distintas interfaces sin que la VLAN nativa se vea afectada. Vemos que las interfaces están activas:

```
R1#sh ip int b
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            192.168.95.1    YES NVRAM  up                    up
FastEthernet0/0.10         unassigned      YES unset  up                    up
Ethernet1/0                unassigned      YES NVRAM  administratively down down
Ethernet1/1                unassigned      YES NVRAM  administratively down down
```

```
Ethernet1/2                unassigned      YES NVRAM  administratively down down
Ethernet1/3                unassigned      YES NVRAM  administratively down down
FastEthernet2/0            unassigned      YES NVRAM  administratively down down
FastEthernet2/0.10         192.168.13.1    YES NVRAM  administratively down down
FastEthernet2/1            unassigned      YES NVRAM  administratively down down
R1#
```

Como en el caso de switch 2, repetimos las operaciones de configuración para el router 2, con las especificaciones correspondientes en cada caso.

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#int f2/0
R2(config-if)#int f2/0.20
R2(config-subif)#des despacho
R2(config-subif)#enc dot1q 20
R2(config-subif)#ip add 192.168.13.129 255.255.255.128
R2(config-subif)#no sh
R2(config-subif)#exit
R2(config)#exit
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#int f0/0
R2(config-if)#ip add 192.168.95.2 255.255.255.0
R2(config-if)#no sh
R2(config-if)#exit
R2(config)#exit
R2#
```

Para finalizar la configuración de los router, tenemos que indicar la ruta y la pasarela por dónde van los paquetes.

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip route 192.168.13.128 255.255.255.128 192.168.95.2
R1(config)#exit
```

Y su comprobación:

```
R1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route
     192.168.13.0/25 is subnetted, 1 subnets
S       192.168.13.128 [1/0] via 192.168.95.2
C    192.168.95.0/24 is directly connected, FastEthernet0/0
R1#
```

Y mismo procedimiento en R2:

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#ip route 192.168.13.0 255.255.255.0 192.168.95.1
```

El último paso que falta es la configuración de los cuatro dispositivos finales de la red, la misma que se ha utilizado en ejemplo anteriores. A continuación se indica la configuración de uno de los PCs, siendo igual para los otros 3 restantes.

```
R-PC-1> ip 192.168.13.2 255.255.255.128 192.168.13.1
Checking for duplicate address...
R-PC1: 192.168.13.2 255.255.255.128 gateway 192.168.13.1
```

Capítulo 4 – Conclusiones y Líneas Futuras

En este trabajo fin de grado, se demuestra que tenemos a nuestra disposición en el mercado una gran variedad de simuladores de redes, que se adaptan a las necesidades de los usuarios. Muchos son de licencia libre, lo cual garantiza su disponibilidad y posibilidades de adaptación, aunque los más potentes del mercado siguen siendo de licencia propietaria, como Riverbed Modeler (simulador actualmente utilizado en la asignatura objeto de este trabajo).

Una vez definidos los objetivos principales del trabajo fin de grado, estudiamos una serie de simuladores de redes, llegando a la conclusión de que el simulador GNS3 era el que cumplía mejor los requisitos marcados al inicio del planteamiento del proyecto.

GNS3 es el simulador de código libre que más se acercaba a las características que se necesitaban para la realización de las prácticas de la asignatura. Por tanto, una vez seleccionado, se estudiaron más a fondo sus principales ventajas e inconvenientes y sus compatibilidades.

El simulador nos ofrece la ventaja que es un software libre y de código abierto, donde la comunidad que tiene GNS3 interviene en las mejoras constantes que se originan en el programa. Siguiendo con las ventajas que tiene, permite realizar múltiples opciones de configuración relacionado con la conmutación y el encaminamiento.

La potente configuración que nos ofrece esta herramienta nos ha permitido efectuar simulaciones similares a las que se hacen actualmente en la asignatura, ajustando las configuraciones a la forma más adecuada al simulador.

Por el lado de las desventajas, hemos comprobado que algunas de las imágenes de los dispositivos que el usuario usa para sus simulaciones tienen que ser suministradas por el mismo usuario, aunque es posible encontrarlas en distintas páginas que están en la

red. Otra de sus desventajas, está a la hora de instalación; ya que a veces surgen problemas, debido a los antivirus y el firewall instalados en el ordenador, teniendo que modificar los permisos de instalación.

Al finalizar el estudio y analizar los resultados, llegamos a la conclusión de que GNS3 es un simulador muy potente, que nos da un rendimiento excelente para nuestro caso. Permite al usuario configurar los dispositivos de red incorporados en las simulaciones como si fueran dispositivos reales, teniendo que realizar las mismas secuencias de comandos que se harían en un entorno real, lo cual supone una importante ventaja.

Además, aunque en las pruebas realizadas solo hemos utilizado componentes de la gama Cisco, el simulador nos permite la utilización de componentes de otras compañías. El *Market* del que se dispone desde la propia web nos permite la descarga de imágenes para otros componentes de distintas compañías y nos ofrece la capacidad de realizar diferentes escenarios combinando elementos de distintos fabricantes. También nos ofrece el uso de máquinas virtuales, mejorando así el rendimiento de nuestra máquina. Si trabajamos de manera local, podemos configurar la cantidad de recursos de memoria RAM que podemos asignarle a cada dispositivo, consiguiendo con las dos opciones, tanto el modo local como el virtual, configurarlo para la realización de más operaciones. El simulador nos permite la creación dinámica de las simulaciones de configuraciones reales sin tener que movernos de nuestro lugar de trabajo.

Las actualizaciones que ofrece el simulador son constantes, mejorando la configuración del programa y permitiendo añadir nuevas imágenes para la realización de las simulaciones.

GNS3 es un simulador que nos permite realizar laboratorios personalizados, ya sea dentro de los que nos ofrece desde de su propio web o los que podemos realizar de otras empresas que nos sirven para conseguir certificaciones para realizar a posteriori instalaciones.

Como ya se ha comentado, una desventaja que tiene GNS3 es que la representación gráfica de los resultados es escasa, por lo que tenemos que apoyarnos en otros programas compatibles como Wireshark para la obtención de información relacionada con el tráfico. Así, una posible línea futura sería estudiar y, en su caso, adaptar a GNS3, los programas capaces de ofrecernos resultados de manera gráfica y estadísticas de las simulaciones para comprobar el rendimiento de éstas.

Otra línea futura consiste en comprobar si el simulador GNS3 se puede utilizar de manera práctica en otras asignaturas de los títulos de Ingeniería de Telecomunicación que se imparten en la ETSIT-UVa, previamente realizando un estudio de compatibilidad del programa con algunas prácticas más avanzadas que las que se han adaptado en este trabajo.

BIBLIOGRAFÍA

Blogspot.com. (12 de Julio de 2012). *Simuladores de Redes*. Recuperado el 8 de Marzo de 2019, de http://simuladoresredes.blogspot.com/

Bombal, D., & Duponchelle, J. (13 de enero de 2019). *Getting Started with GNS3.* Recuperado el 16 de enero de 2019, de https://docs.gns3.com/1PvtRW5eAb8RJZ11maEYD9_aLY8kkdhgaMB0wPCz8a38/index.html

Braden, R., Zhang, L., Berson, S., Herzog, S., & Jamin, S. (Septiembre de 1997). *Resource ReSerVation Protocol (RSVP)*. Recuperado el 9 de Octubre de 2019, de IETF Tools: https://tools.ietf.org/html/rfc2205

Chen, W.-p., Ye, G., Guanghui , H., Chunyu , H., & Hwangnam, K. (28 de Enero de 2005). *Oficial J-Sim*. Recuperado el 1 de Marzo de 2019, de https://sites.google.com/site/jsimofficial/

Cisco. (s.f.). *Cisco Packet Tracer*. Recuperado el 14 de enero de 2019, de Cisco Networking Academy: https://www.netacad.com/es/courses/packet-tracer

Coleman, A., & Duponchelle, J. (26 de Abril de 2019). *Downloading the GNS3 VM. Which virtualization software VirtualBox or VMware?* Recuperado el 30 de Mayo de 2019, de The official guide and reference for GNS: https://docs.gns3.com/1Bn-s1Izkjp13HxcPF4b8QSGfkWJYG_dpMt9U1DQjvZ4/

Danysoft. (16 de Abril de 2016). *Protocolo de Reservación de Recursos: RSVP.* Recuperado el 20 de Febrero de 2019, de danysoft.com/free/reservarecursos.pdf

Gallardo, C., Sebastián Melzi, M., & de Dios, N. (s.f.). *El simulador KivaNS*. Recuperado el 14 de enero de 2019, de studylib.es: https://studylib.es/doc/814977/el-simulador-kivans

Gonzalez, M. (17 de 12 de 2012). *Direccionamiento IPv4.* Recuperado el 9 de Octubre de 2019, de Redes Telemáticas: http://redestelematicas.com/direccionamiento-ipv4

Hedrick, C. (Junio de 1988). *Routing Information Protocol, RFC 1058.* Recuperado el 6 de Octubre de 2019, de IETF Tools: https://tools.ieft.org/rfc1058

Inc, Techopedia. (7 de Mayo de 2019). *TECHOPEDIA*. Obtenido de http://www.techopedia.com/definition/24846/routing-information.protocol-rip

IONOS España. (2019 de Agosto de 16). *El Subnetting para sacar el máximo partido a tu red.* Recuperado el 16 de Septiembre de 2019, de Digital Guide IONOS: https://www.ionos.es/digitalguide/servidores/know-how/subnetting-como-funcionan-las-subredes/

Loddo, J. V., & Saiu, L. (2007-2019). *Marionnet. A virtual network laboratory*. Recuperado el 11 de Marzo de 2019, de https://marionnet.org/site/index.php/en/documentation

Maiti, K. (9 de Febrero de 2018). *A Quick Look at Cloonix, the Network Simulator.* Recuperado el 11 de Marzo de 2019, de Open Source For You: https://opensourceforu.com/2018/02/a-quick-look-at-cloonix-the-network-simulator/

Malkin, G. (Noviembre de 1998). *RIP Version 2, RFC 2453.* Obtenido de IETF Tools: https://tools.ietf.org/html/rfc2453

Mininet Team. (2018). *Mininet Overview.* Recuperado el 14 de enero de 2019, de http://mininet.org/overview/

Moy, J. (Abril de 1998). *OSPF Version 2, RFC 2328.* Recuperado el 6 de Octubre de 2019, de IETF Tools: https://ietf.org/html/rfc2328

Muniz, D. (28 de Abril de 2016). *What are the Differences Between RIP Versions? .* Recuperado el 16 de Marzo de 2019, de 360training: https://www.360training.com/blog/differences-rip-versions/

Nichols, K., Blake, S., Baker, F., & Black, D. (Diciembre de 1998). *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. Recuperado el 9 de Octubre de 2019, de IETF Tools: https://tools.ietf.org/html/rfc2474

Nieto Leon, J. (28 de Agosto de 2013). *OSPF: ¿Qué es?, ¿Por qué OSPF?, Mensajes OSPF.* Recuperado el 24 de Marzo de 2019, de https://netjnl.wordpress.com/2013/08/28/ospf-que-es-por-que-ospf-mensajes-ospf/

oscarmanuuel. (14 de Marzo de 2014). *Simuladores de red.* Recuperado el 7 de Enero de 2020, de es.slideshare.net: https://es.slideshare.net/oscarmanuuel/simuladores-de-red

Regueras, L. (2018). *Guia Docente de Laboratorio de Diseño y Configuracion de Redes.* Recuperado el 6 de Abril de 2019

Regueras, L. (13 de Agosto de 2019). *Guia Docente de Laboratorio de Diseño y Configuracion de Redes.* Recuperado el 6 de Abril de 2019, de https://www.tel.uva.es/docencia/guiasdocentes/curso1819.htm

Rejón, J. (16 de Marzo de 2016). *Calidad de Servicio QoS (Quality of Service)*. Recuperado el 6 de Abril de 2019, de mundotelematico.com: mundotelematico.com/calidad-de-servicio-qos-quality-of-service/

Riverbed Technology. (2019). *Riverbed Modeler: A Suite of Protocols and Technologies with a Sophisticated Development Environment.* Recuperado el 12 de enero de 2019, de https://www.riverbed.com/gb/products/steelcentral/steelcentral-riverbed-modeler.html

Rosen, E., Viswanathan, A., & Callon, R. (Enero de 2001). *Multiprotocol Label Switching Architecture, RFC 3031*. Recuperado el 9 de Octubre de 2019, de IETF Tools: https://tools.ietf.org/html/rfc3031

Shen, N. (Octubre de 2004). *tools.ieft.org*. Recuperado el 9 de Octubre de 2019, de https://tools.ietf.org/html/rfc3906

Socolofsky, T., & Kale, C. (Enero de 1991). *A TCP/IP Tutorial, RFC 1180.* Recuperado el 9 de Octubre de 2019, de IETF Tools: https://tools.ieft.org/html/rfc1180

telectronika.com. (s.f.). Obtenido de https://telectronika.com/tutoriales/gns3-tutorial.instalacion-configuracion/

TETCOS. (s.f.). *NetSim Emulator*. Recuperado el 1 de Marzo de 2019, de https://tetcos.com/emulator.html

U.S. Naval Research Laboratory. (2010). Comparacion de las plataformas de emulacion de red CORE. *MILCOM CONFERENCIA DE COMUNICACIONES MILITARES,2010*, (págs. 864-869). San Jose, California, EE.UU. Recuperado el 26 de Mayo de 2019

Varga, A., & OpenSim Ltd. (2016). *OMNeT++ Simulation Manual.* Recuperado el 20 de Febrero de 2019, de https://doc.omnetpp.org/omnetpp/manual/

Wienberg, N., & Johnson, J. T. (21 de MARZO de 2018). *Cómo funciona MPLS.* Recuperado el 12 de Abril de 2019, de NETWORK WORLD: https://www.networkworld.es/telecomunicaciones/como-funciona-mpls

TABLAS

FIGURAS