



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG del Grado en Ingeniería Informática

Simulación de redes con GNS3



Presentado por Javier García González
en Universidad de Burgos -- 21 de enero de 2019
Tutores: Alejandro Merino Gómez y Daniel Sarabia Ortiz



D. Alejandro Merino Gómez, profesor del departamento de Ingeniería Electromecánica, área de Ingeniería de Sistemas y Automática y D. Daniel Sarabia Ortiz, profesor del departamento de Ingeniería Electromecánica, área de Ingeniería de Sistemas y Automática.

Exponen:

Que el alumno D. Javier García González, con DNI 71294819V, ha realizado el Trabajo final del Gº Ing. Informática titulado: Simulación de redes con GNS3.

y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual, Se autoriza su presentación y defensa.

En Burgos a de <junio de 2016>

Alejandro Merino Gómez

Daniel Sarabia Ortiz

Resumen

Hoy en día, las redes de computadores se encuentran integradas prácticamente en todos los lugares del mundo, desde las redes domésticas más sencillas hasta las redes más complejas en grandes centros de procesamiento de datos. Para ello se han desarrollado varias aplicaciones que nos permiten simular dichas redes.

Para este trabajo, vamos a probar uno de los programas para tal fin. Se trata del programa de simulación de redes GNS3 (Graphical Network Simulator) que permite simular redes complejas y utilizar elementos del fabricante CISCO con licencia gratuita.

Aparte se han estudiado diferentes opciones de implementación y se ha desarrollado un pequeño programa que nos permite configurar remotamente ciertos elementos de la red.

Descriptores

GNS3, Redes, Servidor DHCP, Servidor HTTP, Docker, máquina virtual, Ubuntu, PCs Virtuales, VirtualBox, VMWare, Python, CISCO.

Abstract

In today's world, computer networks are integrated almost everywhere in the world, from the simplest home networks to the most complex networks in big data processing centers. Because several applications have been developed that do not allow you to simulate those networks.

In this project, we are going to test one of the programs for this purpose. It is the network simulation program GNS3 (Graphical Network Simulator) that allows you to simulate complex networks and use elements of the CISCO manufacturer with a free license.

The different implementation options have been studied and a small program has been developed will allow us to remotely configure certain elements of the network.

Keywords

GNS3, Networks, DHCP Server, HTTP Server, Docker, Virtual Machine, Ubuntu, VirtualPCs, VirtualBox, VMWare, Python, CISCO.

Índice general

Índice general	XII
Índice de figuras	IX
Introducción	1
Objetivos del proyecto	4
2.1 Objetivos generales	4
2.2 Objetivos técnicos	4
2.3 Objetivos personales	5
Conceptos teóricos	6
3.1 Dockers	6
3.2 Servidor Web	7
3.3 Servidor DHCP	7
3.4 Router	7
3.5 Switch	8
3.6 Red LAN	8
3.7 Script	8
Técnicas y herramientas	9
4.1 GNS3	9
4.2 Máquinas virtuales	9
4.3 VPCS	11
4.4 Dockers	11
4.5 Telnet – Python	12
4.6 Enrutamiento	13
4.7 Metodología del proyecto	14

Aspectos relevantes del desarrollo del proyecto	15
5.1 Motivación del proyecto	15
5.2 Aprendizaje	16
5.3 Implementación del proyecto	17
5.4 Publicación	18
Trabajos relacionados	19
6.1 Construcción de una red MPLS y validación de GNS3 para su simulación	19
6.2 Implementación de un laboratorio virtual para aprendizaje de SDN	19
Conclusiones y Líneas de trabajo futuras	20
Bibliografía	22

Índice de figuras

Ilustración 1: Red institucional con enrutamiento RIP.	13
---	----

Introducción

Las redes son algo básico en el día a día de todas las personas, comenzando por la más importante, Internet, del cual depende prácticamente todo. En cada casa tenemos pequeñas redes que nos permiten obtener acceso a Internet. Las empresas, colegios, oficinas, etc. también disponen de sus propias redes. Por ello hay una necesidad de crear redes eficientes y operativas, tanto para usuarios avanzados como para usuarios finales sin conocimientos sobre redes.

Las redes se pueden crear y probar en tiempo real, pero también tenemos la posibilidad e incluso en ocasiones la necesidad de poder probar cada componente antes de implementarlo en la realidad. Para ello hemos empleado el programa GNS3, el cuál nos permite simular redes complejas y comprobar el envío de tramas, viendo donde nos puede dar problemas la conexión y localizando fácilmente los problemas.

Encontramos en el mercado más herramientas de simulación de redes, como puede ser el ejemplo de PacketTracer, pero no son tan completas como GNS3, además disponemos de licencia de uso gratuita y nos permite integrar elementos de un fabricante de los más importantes como es CISCO. Gracias a este programa, podemos utilizar máquinas virtuales para simular elementos de la red complejos utilizando Ubuntu o cualquier sistema operativo del que dispongamos.

1.1 Estructura de la memoria

La memoria esta estructurada en los siguientes apartados:

- **Introducción:** Descripción del proyecto y de la solución software estudiada.
- **Objetivos del proyecto:** Se explica brevemente los objetivos fijados y tratados en el trabajo.
- **Conceptos teóricos:** Explicación de los conceptos técnicos vistos a lo largo del trabajo.
- **Técnicas y herramientas:** Se explican las técnicas empleadas a lo largo del proyecto y todas las herramientas empleadas, tanto de gestión del proyecto como de implementación.
- **Aspectos relevantes:** Se comentan los puntos más importantes que se han encontrado durante el desarrollo.
- **Trabajos relacionados:** Explicación y estudio de proyecto relacionados con el campo estudiado.
- **Conclusiones:** Desarrollo de los datos obtenidos y visión general de las conclusiones obtenidas.
- **Líneas de trabajo futuras:** Posibles mejoras o extensiones del proyecto estudiado.

Contenido de los anexos entregados:

1.2 Materiales visualizables:

Los recursos se encuentran accesibles en la página Web del repositorio del proyecto:

https://github.com/javi13gg/TFG_GII_O_MA_18.01-Simulacion-de-redes-con-GNS3



Donde encontraremos:

- Memoria.pdf
- Memoria.docx
- Anexos.pdf
- Anexos.docx
- Proyecto descargable en formato GNS3: PROYECTO.gns3
- Máquinas virtuales necesarias para el funcionamiento:
 - Ubuntu
 - Ubuntu
 - Ubuntu
 - Ubuntu
- Programa Python de configuración: configurador.py
- Presentación del proyecto: presentación.pptx
- Video demostración: demostración.mp4
- Proyectos individuales:
 - Dockers.gns3
 - EjemploInternet.gns3

Objetivos del proyecto

2.1 Objetivos generales

- Simular una red institucional compleja capaz de conectar cada elemento.
- Comparativa de diferentes técnicas de simulación de elementos de la red.
- Estudio teórico y comparación de diferentes formas de ejecutar la aplicación GNS3.
- Pruebas operativas de la red en simulación.

2.2 Objetivos técnicos

- Lectura, información e instalación de la herramienta GNS3. En este punto se empieza a conocer el programa, cómo funciona, que complementos son necesarios y como se instala. Podemos encontrar esta información en los puntos del trabajo: Conceptos Teóricos y Técnicas - Herramientas.
- Información sobre complementos a utilizar. Comparativa de uso entre Dockers y Máquinas Virtuales, como utilizar Dockers y uso de UBUNTU en máquinas virtuales.
- Conexión de GNS3 con el exterior. Investigar las formas de conectar nuestro proyecto simulado en GNS3 con las redes externas de forma que sea visible y tenga acceso a Internet.
- Lanzar GNS3 contra un equipo servidor. Ejecutar GNS3 contra otro equipo (en este caso mi equipo portátil contra el equipo de mesa, que hará de servidor).
- Simulación de una red institucional. Parte principal del proyecto. Se trata de simular una red institucional completa en la cual se van a ir añadiendo ciertos elementos:

- Servidor WEB
- Servidor DHCP
- Router acceso exterior
- Switches
- Red LAN
- VLAN
- Configuración remota de la red. Programar en lenguaje Python un script que conociendo la topología de la red nos permita configurar ciertos elementos de forma remota.

2.3 Objetivos personales

- Realizar estudios sobre un campo visto en menor detalle durante el grado y con herramientas diferentes, ampliando estos conocimientos adquiridos.
- Llevar a la práctica metodologías vistas en la carrera de forma teórica.
- Mejorar mis conocimientos sobre redes.
- Descubrir nuevas funciones de la programación en Python.



Conceptos teóricos

A lo largo del proyecto se van a ir utilizando conceptos relacionados con las redes o con la informática en general, algunos de ellos se detallan a continuación para permitir una mejor comprensión de todos los elementos y técnicas.

3.1 Dockers

También conocidos como contenedores [\[1\]](#). Se trata de emplear un software determinado que nos permite simular hardware virtual que a grandes rasgos veremos como un equipo totalmente independiente, este equipo podrá tener su propio sistema operativo en el que poder trabajar sin modificar el software anfitrión. La diferencia con las máquinas virtuales más adelante explicadas es que no se carga una imagen completa del sistema operativo, sino que solo se incluyen las partes necesarias.



3.2 Servidor Web

Se trata de un programa capaz de ejecutar una aplicación en el lado del servidor, es capaz de realizar conexiones en ambos sentidos o en un solo y también de realizar conexiones tanto en modo síncrono como asíncrono con el lado del cliente [2]. Para procesar esta información el lado cliente necesita una aplicación conocida como navegador Web. Suele utilizarse un protocolo de comunicaciones conocido como HTTP.

3.3 Servidor DHCP

El servicio DHCP (Dynamic Host Configuration Protocol) es el protocolo de configuración dinámica de host, un estándar TCP/IP diseñado para simplificar la administración de la configuración IP de los equipos de nuestra red [3]. El estándar DHCP permite el uso de servidores DHCP para administrar la asignación dinámica a los clientes DHCP de la red, de direcciones IP y de otros detalles de configuración relacionados con el direccionamiento IP, tales como la puerta de enlace o los servidores DNS, por ejemplo, siempre que los clientes estén configurados para utilizar un servidor DHCP, en lugar de estar configurados manualmente con una dirección IP estática [2].

3.4 Router

Es un dispositivo que proporciona conectividad a nivel de red (o lo que es lo mismo, el nivel tres en el modelo OSI) [4]. Su función principal consiste en enviar o encaminar paquetes de datos de una red a otra, es decir, interconectar subredes, entendiendo por subred un conjunto de máquinas IP que se pueden comunicar sin la intervención de un encaminador (mediante puentes de red o un switch), y que por tanto tienen prefijos de red distintos [5].

3.5 Switch

Su función es interconectar dos o más host de manera similar a los puentes de red, pasando datos de un segmento a otro de acuerdo con la dirección MAC de destino de las tramas en la red y eliminando la conexión una vez finalizada ésta [\[5\]](#). Proporciona conectividad a nivel de la capa de enlace (nivel 2 del modelo OSI).

3.6 Red LAN

Una red LAN (Local Area Network) consiste en varios ordenadores conectados entre sí, normalmente en un espacio pequeño (sala, edificio, etc.) que permite que dichos equipos compartan recursos [\[4\]](#).

3.7 Script

Documento con código de programación que incluye instrucciones que permiten realizar las funciones para las cuales está diseñado [\[6\]](#).

Técnicas y herramientas

4.1 GNS3

La herramienta principal utilizada en el proyecto ha sido el programa GNS3, un simulador gráfico de redes que nos permite el diseño de complejas topologías y poder hacer simulaciones de su comportamiento.

Se trata de una herramienta muy potente que nos permite seleccionar muchas opciones de configuración. En nuestro caso en lugar de lanzar la simulación sobre el propio equipo, se va a lanzar sobre una máquina virtual GNS3 sobre VMWare, como podremos contemplar en los anexos, donde veremos el proceso de instalación y configuración [8].



4.2 Máquinas virtuales

En los casos de máquinas virtuales, se utiliza un software de virtualización como por ejemplo puede ser VirtualBox o VMWare (las dos utilizadas en este proyecto y gratuitas). En estos casos se instala una imagen de un sistema operativo completo sin dependencia del anfitrión ni con otros sistemas. Tan solo se comparte el hardware real.

- VirtualBox
VirtualBox es un Software que nos permite virtualizar arquitecturas x86-64, esta especificación indica que se soporta una cantidad muy superior de memoria virtual y física que sus

predecesores. La aplicación VirtualBox ha sido desarrollada por Oracle dentro de su gama de virtualización. Nos permite realizar instalaciones de sistemas operativos adicionales, "invitados" que se ejecutan sobre otro sistema conocido como "anfitrión". Cada sistema tendrá su propio entorno virtual independiente.

Disponemos de funcionalidades importantes de pago, pero estas no las vamos a tener en cuenta para nuestro proyecto, ya que no lo vamos a necesitar.

En cuanto a los dispositivos de almacenamiento, se crean dentro del disco duro del equipo anfitrión unas unidades llamadas *Virtual Disk Image* propias de esta aplicación.

Incluye funciones para poder cargar imágenes ISO que serán visualizadas como unidades ópticas de CD/DVD.

La configuración es bastante abierta, permitiendo controlar aspectos como la aceleración 3D, pantalla completa, conexiones de Ethernet, integración de ratón y teclado y la posibilidad de escoger la cantidad de memoria y de potencia de procesamiento a utilizar del equipo anfitrión.



- VMWare

Se trata de otro software de virtualización, perteneciente en este caso a la empresa EMC Corporation (DELL Inc). Está disponible para las arquitecturas X86. Dentro de la propuesta se incluyen VMware Workstation, VMware Server y VMware Player. En nuestro caso solo utilizamos la primera de ellas. Este software está disponible tanto para Windows como para Linux. Para MacOS está disponible bajo otro nombre, VMware Fusion.

El funcionamiento es similar al del mencionado anteriormente con VirtualBox, la aplicación se ejecuta sobre el sistema operativo del equipo anfitrión, a su vez en la aplicación se cargan

otros sistemas operativos virtualizados, los cuales generan su propio hardware virtual aprovechando parte de los recursos del anfitrión.



4.3 VPCS

Se trata de emuladores de PC que vienen ya configurados en el programa GNS3 con funciones muy básicas para poder probar la red. Gracias a ellos podemos simular equipos que no nos requieran instalaciones ni servicios especiales [\[9\]](#).

Estos equipos por ejemplo no son útiles para realizar un servidor, pero si son muy útiles para poder simular un equipo terminal sin tener que emplear todos los recursos que nos consume una máquina virtual con un sistema operativo completo.

4.4 Dockers

Se trata de virtualización como en los casos anteriores, pero en este tenemos la diferencia de que se configura solo parte del sistema operativo, en cada caso la parte que vayamos a necesitar. De esta forma conseguimos el mismo resultado que con un sistema completo, pero consiguiendo un considerable ahorro de recursos [\[10\]](#).

El uso de Dockers se ha estudiado apreciando una disminución de los recursos empleados por el equipo anfitrión que ejecuta el proyecto, pero por otra parte dificulta la configuración remota de la red, así como el proceso de exportar el proyecto para un uso posterior, de forma que se ha dejado como un estudio dentro del propio proyecto, pero finalmente se ha descartado su uso y se ha decidido utilizar máquinas virtuales y VPCS [\[11\]](#).

Es interesante el estudio ya que para proyectos que requieran mucha carga de equipos, o bien no necesitemos funciones como la configuración remota sí que es más que interesante su utilización.

La disminución de la carga de trabajo que producen es difícilmente medible, ya que depende de muchos factores, como el servicio final que dan o el número de equipos a los que acaban dando servicio. Pero si se puede asegurar que es apreciable dicha disminución.



4.5 Telnet – Python

Para la realización de un programa que nos permita la configuración remota de la red se ha escogido Python [\[7\]](#), que nos permite utilizar Telnet para poder conectar a los dispositivos y realizar configuraciones básicas como son cambios de IP, máscara de red y puerta de enlace [\[12\]](#).



Se ha escogido Python por encima de otros lenguajes dado que es lenguaje sobre el que más documentación tenemos relativa al uso con redes, además de ser uno de los lenguajes más potentes en cuanto a funciones realizadas.

4.6 Enrutamiento

El enrutamiento [5] consiste en buscar el camino dentro de un abanico de posibilidades dentro de una red cuya topología tiene una gran diversidad de rutas posibles.

Para el caso de este proyecto hemos utilizado un enrutamiento de tipo dinámico, el cual calcula las posibles rutas para tomar una de ellas. El algoritmo escogido ha sido el RIP (Routing Information Protocol), este calcula la ruta más corta posible de entre todas las disponibles basándose en el número de saltos (es decir, equipos que se encuentran en el camino) encontrados entre origen y destino. Este algoritmo tiene un límite de 15 saltos, si son más va a considerar ese destino como un punto al que no puede llegar.

Se trata de un protocolo libre, puede ser usado en routers de cualquier fabricante, frente a otros protocolos de uso propio como puede ser el ejemplo del protocolo EIGRP de CISCO Systems. A continuación, vemos la red creada en el proyecto, en la ilustración 1, donde se ha utilizado en sus routers el protocolo de enrutamiento RIP.

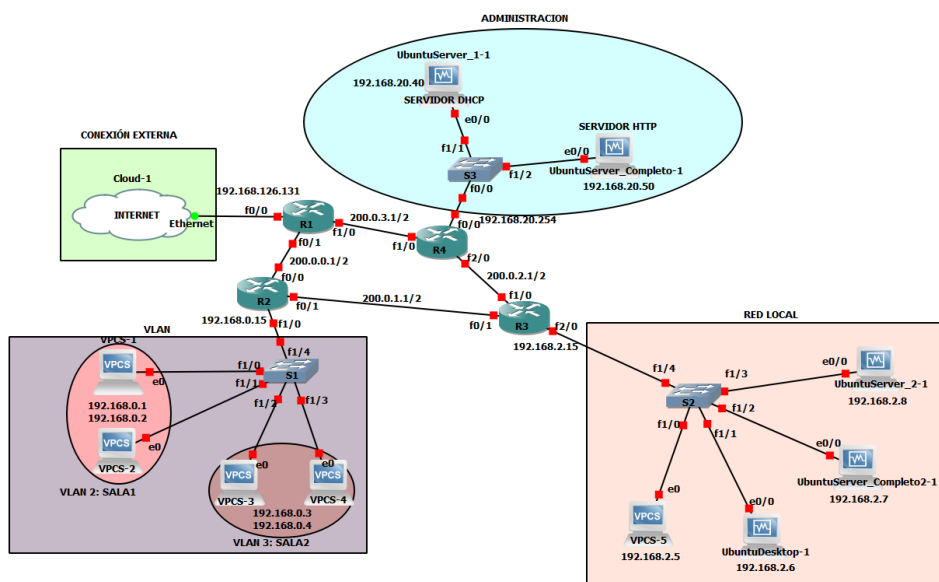


Ilustración 1: Red institucional con enrutamiento RIP.

4.7 Metodología del proyecto

Para el desarrollo del proyecto se ha utilizado la herramienta GitHub, donde se han indicado los pasos seguidos en el proyecto y cada cambio, para ello se ha utilizado la opción de Issues (tareas), de commits (cambios) y se ha utilizado el módulo de proyecto donde se puede ver en cada momento que tareas están hechas, en tratamiento, pendientes, etc.

La metodología ha sido ágil, permitiendo adaptarse al proyecto realizado y cambiando las tareas en función de lo que la herramienta GNS3 nos permitía en cada caso, de esta forma muchos de los objetivos han sido modificados sobre la marcha para adaptarnos.

Dentro de las metodologías Agiles se ha empleado la metodología SCRUM, también conocida como metodología del "caos". Se caracteriza por desarrollo incremental, dividiendo el proyecto en mucho pequeños proyectos, divididos cada uno de ellos en etapas. Estas etapas incluyen diferentes pasos: análisis, desarrollo y test. Cada una de las partes realizadas en el proceso de desarrollo es conocida como una interacción.

Gracias a esta metodología, hemos podido ser flexibles con el proyecto, adaptando cada etapa al tiempo disponible y modificando ciertos objetivos marcados inicialmente por otros más acordes al proyecto.

Cada cierto tiempo se ha fijado una reunión de control del proyecto, las cuales se han realizado mediante videoconferencia utilizando *Skype for Business* de la UBU, donde se ha revisado el trabajo realizado hasta el momento. También se han fijado reuniones de retrospectiva, para ver el trabajo a realizar a partir de los resultados ya obtenidos. Por otra parte, se han tenido reuniones de revisión en las que se ha comprobado el trabajo realizado hasta la fecha de dicha reunión [13].



Aspectos relevantes del desarrollo del proyecto

En este punto vamos a ver los aspectos tratados a destacar a lo largo de la ejecución del proyecto. Los problemas encontrados y las decisiones de continuidad tomadas.

5.1 Motivación del proyecto

Para la elección de este proyecto, me basé en una de las ramas de la informática que más ha llamado mi atención a lo largo de la carrera, de la cuál siempre he querido profundizar más en mi aprendizaje.

Por este motivo fue sencillo escoger este proyecto, el cuál me ha permitido conocer una herramienta nueva y muchos detalles sobre la configuración y simulación de redes que hasta hace poco desconocía.

Este proyecto me ha permitido además adentrarme un poco más en los conceptos que utilizo en el día a día en mi trabajo como técnico de soporte, en el cuál en innumerables ocasiones he tenido que configurar o reparar redes institucionales.



5.2 Aprendizaje

Debido a la complejidad del mundo de las redes de computadores, el tiempo de estudio de cada objetivo ha sido bastante largo, muchas horas de investigación para comprender si era posible cada punto de los que nos habíamos propuesto, y si lo era ver el método de llevarlo a la práctica.

La documentación más leída y consultada a lo largo de todo el proyecto ha sido la de la página de la aplicación, donde se encuentra toda la documentación sobre todos los módulos y gran parte de las opciones del programa, también incluida en la bibliografía en varios de los apartados estudiados:

- Documentación de GNS3 [[8](#)]

Además de esta Web he leído y consultado los siguientes libros, los cuales están indicados igualmente en el apartado de bibliografía:

- Computer Networking: A Top Down Approach, 6th edition, Jim Kurose, Keith Ross, Addison-Wesley, 2012 [[2](#)]
- Comunicaciones y redes de computadores, Stallings, Pearson, 2004 [[5](#)]
- Redes de Computadoras. Un enfoque descendente, 5º edición, Jim F. Kurose & Keith W. Ross, Pearson, 2010 [[15](#)]



5.3 Implementación del proyecto

A lo largo del proyecto se han visto algunos apartados que han llevado mucho tiempo de estudio y se han implementado en un tiempo reducido, como es el caso de la [\[14\]](#) conexión de la red generada en GNS3 al exterior (Internet), del cual se ha encontrado mucha información y videos de uso, algunos ya descatalogados y otros más actuales. Finalmente se ha escogido el método más actual y se ha dejado la conexión dentro del proyecto.

En cambio, tenemos otro caso como es el del uso de [\[1\]](#) Dockers. La recopilación de información y el estudio de estas herramientas llevó mucho tiempo, su puesta en funcionamiento fue relativamente rápida en comparación con otros elementos, pero se tuvo que descartar su uso por chocar con otros de los objetivos. Esto no se puede considerar algo negativo, ya que se ha demostrado su utilidad, aunque en este proceso no haya sido la mejor opción y se haya descartado su uso. Podemos ver varios ejemplos y el proceso de instalación en los anexos: Apéndice C, apartado C.4, sección uso de Dockers.

Se ha realizado un estudio en el cual se ha comprobado que la cantidad de recursos que el programa GNS3 requiere del equipo anfitrión se puede disminuir si utilizamos un equipo como servidor que lance el programa GNS3 y sobre el cuál se consumen dichos recursos pero trabajamos desde un equipo cliente que es desde el cuál lanzamos el proyecto, este equipo consume ciertos recursos, pero la carga más fuerte se hará sobre el servidor, de esta forma se reparte el consumo de dichos recursos. Podemos ver el ejemplo en los anexos: Apéndice C, apartado C.4, sección Conectar GNS3 contra un servidor remoto [\[16\]](#).

Por otro lado, tenemos otros componentes cuyo estudio y documentación requiere menos tiempo, debido a que son elementos previamente utilizados a lo largo del grado y de los cuales entendemos su funcionamiento, en cambio la puesta en funcionamiento de estos elementos ha sido más larga y compleja. El servidor Apache fue complejo de implementar ya que las máquinas virtuales utilizadas inicialmente eran las del laboratorio de redes de la politécnica. Estas no nos permitían la instalación de servicios adicionales por temas de

permisos, por lo que se descargó e instaló desde cero una máquina virtual con todos los permisos y con la conexión a Internet activa inicialmente, aunque luego para el proyecto se tuviese que volver a dejar sin conexión. Una vez conseguido este paso se realizó la configuración como servidor Apache con todas sus funciones y se estableció una página que nos permite obtener los datos de la red del servidor para poder comprobar su funcionamiento.

En cuanto a las redes virtuales VLAN, no se encontraron grandes problemas, pero al ser un punto con varios aspectos de configuración y necesitar de un Switch que inicialmente daba otros servicios, se decidió aislar la red con su propio switch, de esta forma su funcionamiento es más independiente y cumple mejor su labor.

5.4 Publicación

Al no tratarse de una aplicación puramente ejecutable, los ficheros necesarios y el código fuente de los scripts generados ha sido colgada en el repositorio GitHub. Para la ejecución del programa será necesario seguir los pasos de instalación y ejecución descritos en los anexos, en los cuales se explica el proceso completo de instalación y configuración para la realización de las pruebas necesarias, así como ejemplos de pruebas ya realizadas. También se subirá el video de presentación del proyecto y un video con una demostración de funcionamiento.

Trabajos relacionados

Los trabajos relacionados encontrados se describen a continuación:

6.1 Construcción de una red MPLS y validación de GNS3 para su simulación

Autor: Héctor Delgado Patiño. Organización: PFC Universidad de Cartagena.

Se trata de un Trabajo de Fin de Grado realizado para la Universidad de Cartagena en que se aborda una red de comunicaciones basada en el protocolo MPLS, para la que se han empleado los elementos CISCO y que ha sido probada y desarrollada en el programa GNS3, al igual que este proyecto [[17](#)].

6.2 Implementación de un laboratorio virtual para aprendizaje de SDN

Autor: Rubén Isa Hidalgo. Organización: TFG Universidad de Cantabria.

El contenido de este proyecto esta basado en la creación de un entorno de trabajo que simula, mediante el uso de elementos de GNS3, la simulación y práctica con redes definidas por Software. Se va a emplear el módulo Mininet del entorno GNS3 que nos permite crear redes de análisis de forma intuitiva [[18](#)].

Conclusiones y Líneas de trabajo futuras

Conclusiones

Las conclusiones obtenidas tras la realización del proyecto son:

- Se han desarrollado todos los objetivos propuestos, demostrando que GNS3 es una herramienta potente capaz de simular prácticamente cualquier función que podamos llevar a la práctica en la vida real.
- Gracias al uso de un software libre tan conocido, se han podido encontrar soluciones a los problemas que han ido surgiendo a lo largo de la realización, pese a la complejidad de alguno de los proyectos llevados a cabo, siempre ha sido posible encontrar documentación al respecto, que, si bien no solucionaba el problema, si nos ofrece las pistas necesarias para poder hacerlo por nuestra cuenta.
- Se han desarrollado los conocimientos adquiridos durante el grado, tanto a nivel de redes, como de sistemas operativos, programación y de gestión de proyectos. Abarcando un amplio espectro de todos ellos.
- Gracias a este proyecto se han aplicado técnicas y herramientas desconocidas hasta ahora, pero también otras muchas ya empleadas durante la carrera que han permitido el desarrollo de elementos del proyecto que sin dichos conocimientos hubiese sido casi imposible.
- El nivel de estudio y desarrollo me ha permitido conocer publicaciones de artículos, libros y foros muy interesantes sobre todos los temas tratados en el proyecto, que a buen seguro me resultarán útiles tanto en mi vida profesional como en mis estudios.
- Gracias al uso de la metodología ágil se ha podido llevar a cabo un proyecto de tal magnitud en un espacio reducido de tiempo, pudiendo adaptar cada objetivo a su duración final, más allá de la duración estimada que se calculó en un principio.

Líneas de trabajo futuras

Ante un proyecto tan amplio como este, solo podemos pensar en mejoras, ampliaciones y aplicaciones:

- Sería posible ampliar la red, incluyendo nuevos servidores, conectando mediante routers y switches como fuese necesario.
- Se puede llevar a la práctica la simulación del proyecto, de forma que se puede hacer un estudio comparativo entre las conclusiones obtenidas en la simulación y los resultados reales.
- Se puede incluir mayor complejidad algorítmica a la configuración remota de la red mejorando el script generado mediante Python.
- Automatizar la generación del fichero que se le pasa al script encargado de la configuración remota de la red.
- Estudio de la transmisión de datos empleando el programa WhireShark.
- Posibilidad de realizar el mismo proyecto en otros entornos de simulación redes como puede ser PacketTracer y comparar el rendimiento y las posibilidades de cada uno.

Bibliografía

- [1] Uso de Dockers en GNS3:
<https://docs.gns3.com/1KGkv1Vm5EgeDusk1qS1svacpuQ1ZUQSVK3XqJ01WKGc/index.html>

- [2] Libro Redes de Computadoras. Un enfoque descendente, 5^o edición, Jim F. Kurose & Keith W. Ross, Pearson, 2010

- [3] Servidor DHCP.
<https://2003server.webcindario.com/dhcp/definici.htm>

- [4] Router. (2018, 7 de noviembre). Wikipedia, La enciclopedia libre. Fecha de consulta: 11:38, enero 14, 2019 desde <https://es.wikipedia.org/w/index.php?title=Router&oldid=111831803>.

- [5] Libro Comunicaciones y redes de computadores, Stallings, Pearson, 2004.

- [6] Script:
<https://es.wikipedia.org/wiki/Script>

- [7] Python:
<https://www.python.org/>

- [8] Página Web oficial de GNS3:
<https://gns3.com/>

- [9] Documentación de GNS3 sobre VPCS:
<https://docs.gns3.com/1ALk8HxK9qPCjhNL2OzFtW64Z7b2kBwEnEe5xbPG2gJ0/index.html>

- [10] Documentación de GNS3 sobre Dockers:
<https://docs.gns3.com/1KGkv1Vm5EgeDusk1qS1svacpuQ1ZUQSVK3XqJ01WKGc/index.html>

- [11] Enlace al proyecto de Dockers del repositorio en GitHub:
https://github.com/javi13gg/TFG_GII_O_MA_18.01-Simulacion-de-redes-con-GNS3/blob/master/Dockers.gns3

- [12] Documentación sobre Python. Telnet:
<https://docs.python.org/2/library/telnetlib.html>

- [13] Apuntes de la asignatura Gestión de Proyectos.

- [14] Conexión de GNS3 con Internet:
<https://docs.gns3.com/1vFs-KENh2uUFfb47Q2oeSersmEK4WahzWX-HrMIMd00/index.html>

- [15] Libro Computer Networking: A Top Down Approach, 6th edition, Jim Kurose, Keith Ross, Addison-Wesley, 2012 URL:
<http://www-net.cs.umass.edu/kurose-ross-ppt-6e/>

[16] Proyecto en Github:

https://github.com/javi13gg/TFG_GII_O_MA_18.01-Simulacion-de-redes-con-GNS3

[17] "Construcción de una red MPLS y validación de GNS3 para su simulación" URL:

<http://repositorio.upct.es/bitstream/handle/10317/5251/tfg682.pdf?sequence=1>

[18] "Implementación de un laboratorio virtual para aprendizaje de SDN" URL:

<https://repositorio.unican.es/xmlui/bitstream/handle/10902/9377/387880.pdf>