Universidad Zaragoza

Trabajo Fin de Grado

Gestión y análisis de un escenario virtualizado de control para una red inalámbrica basado en tecnología Wi-Fi

Management and analysis of a virtual control scenario for a WiFi-based wireless network

Autor

Diego García Roca

Director

Julián Fernández Navajas

ESCUELA DE INGENIERÍA Y ARQUITECTURA

2021

Gestión y análisis de un escenario virtualizado de control para una red inalámbrica basado en tecnología Wi-Fi

RESUMEN

En los últimos años, los dispositivos móviles han aumentado tanto en complejidad como en número. Este desarrollo ha provocado que las redes inalámbricas se dividan entre las dos tecnologías predominantes: las redes de arquitectura celular para exteriores y las redes Wi-Fi en interiores. Estas últimas son la base de este proyecto.

Las redes Wi-Fi están en constante evolución para hacer frente a los nuevos desafíos que surgen. Actualmente, el estándar mas extendido es Wi-Fi 5, aunque el desarrollo de Wi-Fi 6 ha avanzado enormemente y ya comienzan a comercializarse dispositivos que lo implementan. Wi-Fi debe dar solución al enorme crecimiento del número de dispositivos con capacidad de conexión debido, por ejemplo, a la expansión de *Internet of Things* (IoT) o a la demanda de mayores tasas de transmisión por parte de los servicios multimedia en tiempo real.

Estas circunstancias han empujado a los operadores hacia el desarrollo de redes con un mayor número de puntos de acceso especializados en la prestación de servicios determinados. Esta solución pasa por la implementación de LVAP (*Light Virtual Access Points*), que puedan ser orquestados, encendidos y apagados, en función de la demanda. Además, mediante la incorporación de una estructura de red controlada por software (SDN), se permite distribuir y balancear la carga de tráfico en redes densamente pobladas, a la vez que se facilita el proceso de gestión y monitorización de la red.

A lo largo de este documento se describen los procedimientos, mecanismos y pruebas realizadas para el despligue de una infraestrutura de red que permita el uso de las tecnologías de control SDN para Wi-Fi (SDWN) y disponga de un sistema de gestión que haga posible la réplica de escenarios reales en entornos controlados de laboratorio para un correcto análisis de prestaciones.

Índice de Contenidos

Índice de Figuras

Capítulo 1

Introducción

Este primer bloque de la memoria se dedicará a dar una breve explicación y contexto de los conceptos más prominentes del proyecto, así como su estado actual, ventajas y puntos débiles, además de su rol y problemática en el desarrollo del proyecto. Estos conceptos principales son: la virtualización, las redes definidas por software y la tecnlogía Wi-Fi.

Virtualización

Uno de los conceptos clave de este documento es la virtualización. Esta tecnología permite crear una división lógica de los recursos hardware en diferentes equipos virtuales, llamados generalmente máquinas virtuales. Cada máquina puede tener su propio sistema operativo, ejecutar diferente *software* o estar dedicada a servicios independientes mientras se encuentran utilizando recursos de porciones diferentes del *hardware* subyacente. La virtualización permite emular desde aplicaciones y servicios sencillos hasta redes completas [7].

Anteriormente, los servicios se solían ejecutar en equipos dedicados únicamente a una tarea concreta, provocando el encarecimiento de proporcionar dichos servicios en términos de equipamiento y mantenimiento. Gracias a la virtualización, estos servicios pudieron abstraerse a nivel lógico, logrando así una reducción de costes.

La virtualización hoy en día presenta numerosas ventajas, siendo las más importantes [22, 27]:

- **Mayor fiabilidad**: Cuando un equipo se ve afectado por cualquier tipo de fallo, la calidad del servicio puede verse reducida. En cambio, una máquina virtual es sencilla de desplegar y sustituir en caso de errores. El proceso de recuperación

resulta más simple y puede realizarse en minutos, mejorando enormemente la fiabilidad y proporcionando un servicio ininterrumpido.

- **Escalabilidad**: La virtualización proporciona mayores posibilidades de expansión de una red o un servicio al poderse replicar los equipos con mucha más facilidad y poder distribuir las prestaciones de los equipos reales de manera más eficiente y flexible entre diferentes aplicaciones.

- **Seguridad**: Un entorno virtual es fácilmente controlable y sencillo de monitorizar por un usuario. Cualquier error o software malicioso está limitado al equipo virtualizado evitando su traslado a otros equipos físicos o máquinas virtuales que compartan el *hardware*. Por este motivo, los entornos virtualizados son muy útiles para testear diversas aplicaciones sin poner en riesgo el sistema que lo aloja.

- **Facilidad de gestión**: La creación de equipos y entornos virtuales es rápida y es posible monitorizarlos de forma sencilla desde herramientas dedicadas.

- **Reducción de costes**: En los entornos no virtualizados, mientras la aplicación o servicio correspondiente no consume recursos del servidor, se están malgastando recursos. Con la virtualización de este entorno se consigue un enfoque con una eficiencia mayor, reduciendo costes en recursos físicos.

Las máquinas virtuales utilizadas en este proyecto se han diseñado con el objetivo de formar parte de una red virtual inalámbrica. Esto significa que los recursos *hardware* que normalmente se utilizan en la creación de una máquina virtual no son suficientes. Además de la necesidad de asignar recursos como el disco, la memoria o el procesador, los equipos virtuales necesitarán acceso a las interfaces de red, tanto cableadas como inalámbricas. La máquina virtual asociada a dicha interfaz deberá capturar y obtener el control de la interfaz inalámbrica real, para proporcionar soporte a las aplicaciones que se utilizan en la red virtual que se ha diseñado. De esta manera, se posibilita crear uno o varios puntos de acceso reales en una red compuesta por equipos y enlaces virtuales.

Tecnología Wi-Fi

Por otro lado, otra tecnología básica para este proyecto son las redes inalámbricas y ls estándares Wi-Fi. Wi-Fi es el nombre con el que se refiere a los estándares IEEE 802.11, publicados en 1997. Estas directrices definen las bases de las redes inalámbricas de área local. Estas redes presentan ventajas como movilidad, flexibilidad

y tasas de transmisión elevadas. Dichas cualidades han provocado que sea una de las tecnologías más extendidas en la actualidad, con grandes previsiones de crecimiento, lo que ha provocado un abaratamiento en los componentes Wi-Fi, propiciando aún más su desarrollo.

En este proyecto, se utiliza el modo infraestructura de las redes Wi-Fi. La arquitectura empleada es la mostrada en la siguiente figura:

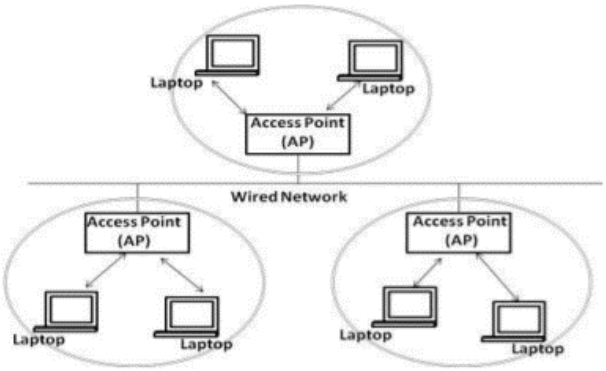

Figura 1.1: Arquitectura de red WLAN (*Wireless Local Area Network*).

En este formato, la red cuenta con uno o varios puntos de acceso a los que los dispositivos inalámbricos (STAs) deben asociarse para conseguir acceso a la red. Los puntos de acceso se encuentran conectados, generalmente de forma cableada, con el *backbone* de la red. Todos los dispositivos que se encuentren en rango de cobertura de un punto de acceso podrán conectarse a través de los AP (*Access Point*). Esto significa que los puntos de acceso son uno de los elementos más importantes y críticos de la red inalámbrica. En este modo de operación, existe también la posibilidad de producirse un *handoff*: esto ocurre cuando una STA se aleja del rango de cobertura óptimo de un AP y comienza un proceso de asociación con un nuevo punto de acceso. Las redes virtuales de prueba se diseñarán de acuerdo con este tipo de arquitectura.

Sin embargo, las redes WLAN cuentan con un ancho de banda limitado, que ha de ser repartido entre los usuarios conectados simultáneamente. Debido a ello, los mayores desafíos a los que se enfrentan las redes inalámbricas son algunos como las nuevas aplicaciones con altos requerimientos de QoS (*Quality of Service*), crecimiento del número de usuarios y *handoff* ligeros y transparentes [24].

En este contexto de desarrollo, y teniendo en cuenta el enorme crecimiento del número de dispositivos conectados a Internet, las soluciones previamente desarrolladas no fueron pensadas para un crecimiento tan repentino y acelerado. Estas proyecciones

implican que las redes móviles e inalámbricas deben evolucionar para convertirse en más eficientes, seguras, inteligentes y, más importante aún, completamente escalables para hacer frente al enorme número de nuevos usuarios.

SDN

Las redes definidas por software y las redes virtuales son actualmente las que proyectan mayores posibilidades de lograr paliar dichos problemas en un futuro cercano [14, 12]. Aunque los principios básicos de las SDN y las redes virtualizadas son similares, son tecnologías diferentes que pueden usarse combinadas. En este caso, una SDN suele actuar como centro de control de las diferentes funciones de las redes virtualizadas que a su vez son gestionadas mediante una función que se denomina orquestación.

Los principios de las SDN son: separar los planos de control y datos de una red, incluir un controlador para la red de datos y virtualizar los componentes de la red [17]. Gracias a la centralización del control en un equipo, la infraestructura de la red subyacente queda abstraída de las aplicaciones [26]. Una SDN presenta tres planos diferenciados: el plano de control, que se encuentra centralizado en un equipo controlador, el de datos, conformado por los equipos encargados de la transmisión de los flujos de información y el plano de gestión, que proporciona al administrador de la red un mecanismo de monitorización, análisis y despliegue de la red. Como resultado, estas divisiones consiguen gran programabilidad, automatización y control sobre la red, además de otorgarles una enorme flexibilidad, escalabilidad y capacidad de adaptación a las necesidades de los diferentes servicios.

En este proyecto, la tecnología SDN se aplicará al despliegue de una red inalámbrica. Las ventajas del uso de una SDWN (*Software Defined Wireless Network*) son variadas: permite el control de la red de acceso Wi-Fi, mejora el rendimiento en los procesos de asociación entre dispositivos y puntos de acceso, reduce el impacto negativo de los handoffs y gestiona el ancho de banda disponible de una manera más eficiente. Debido a estos beneficios, en la arquitectura desarrollada, el controlador es el encargado de la creación de LVAPs (*Light Virtual Access Points)* y su asignación a los dispositivos inalámbricos asociados con los puntos de acceso de la red.

El plano de gestión, de gran importancia en el desarrollo, estará implementado por una herramienta de análisis y gestión de red que permitirá al administrador observar constantemente el funcionamiento de la red, realizar modificaciones y controlar todos los aspectos de los escenarios creados.

1.1. Motivación

La creciente popularidad de los smartphones y dispositivos IoT ha provocado un gran crecimiento en el tráfico de datos móviles enviados a través de las redes inalámbricas. Las diferentes estadísticas y estudios de futuro demuestran que las redes móviles de arquitectura celular son las más utilizadas en redes *wide-area* y exteriores, mientras que la tecnología Wi-Fi permanecerá como la más utilizada en entornos de interior y redes WLAN [9].

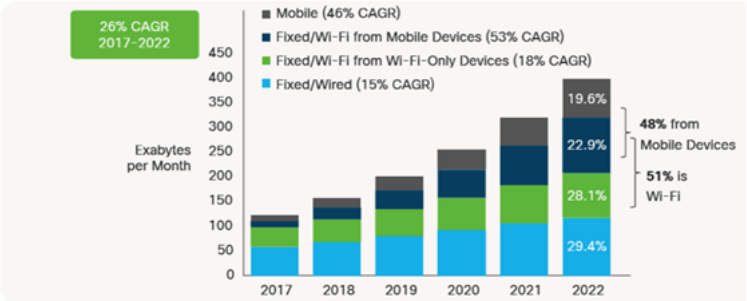

Figura 1.2: Previsiones de crecimiento Wi-Fi.

Siguiendo esta tendencia de crecimiento, las diversas aplicaciones también han evolucionado. El flujo de datos de aplicaciones con altas necesidades de calidad de servicio está en auge: aplicaciones de voz, contenidos multimedia, streaming y dispositivos IoT. Actualmente, el tráfico intercambiado en redes Wi-Fi desde dispositivos móviles supera ya el 59 % y se prevé que continúe en aumento. Para hacer frente a estos desafíos, se requieren redes inalámbricas muy robustas [9].

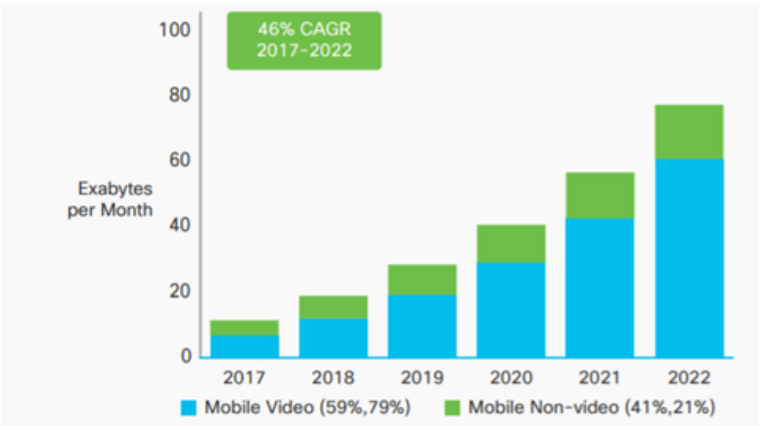

Figura 1.3: Desglose por tipos de tráfico.

Como solución, muchos proveedores Wi-Fi apuestan por diseñar redes más ligeras

y seguras. Esto permitirá incrementar en gran medida la cantidad de puntos de acceso Wi-Fi para escalar las redes inalámbricas de forma proporcional al aumento del tráfico. Estudios realizados por grandes compañías, como Cisco, estiman que para el año 2022 se habrán desplegado 549 millones de puntos de acceso públicos, frente a los 124 millones que existían en el año 2017.

Una de las líneas de desarrollo más apoyadas entre los diseñadores y proveedores consiste en aplicar las tecnologías de la virtualización de equipos y redes y las redes definidas por *software* a las redes de acceso Wi-Fi para aligerarlas y facilitar su escalado. En este contexto se enmarca el desarrollo de este proyecto: es necesario para un correcto análisis de estas redes, contar con un entorno controlado de laboratorio que permita la replica de escenarios habituales en los que realizar los trabajos de desarrollo de nuevas prestaciones.

1.2. Objetivos

El objetivo principal de este trabajo consiste en el despliegue de una infraestrutura de red que permita el uso de las tecnologías de control SDN para Wi-Fi (SDWN) basado en la utilización de puntos de acceso virtuales ligeros (LVAP) que puedan ser controlados (configurando las diferentes funciones de red) y orquestados (encedido y apagado bajo demanda). Además, dicha infraestructura debe disponer de un sistema de gestión que haga posible la réplica de escenarios reales en entornos controlados de laboratorio para un correcto análisis de prestaciones por parte de los administradores de la red. Para facilitar la consecución del objetivo, éste ha sido dividido en hitos menores.

Por un lado, se necesita disponer de una herramienta de gestión y análisis de redes que permita tener al administrador una visión completa y en conjunto de los escenarios. Como se deben distribuir los equipos *hardware* que alojan las máquinas virtuales en ubicaciones físicas muy variadas, esta distribución física del escenario no debe suponer al administrador una complejidad añadida. La herramienta que sea seleccionada debe permitir la gestión de estos equipos remotos de forma eficiente y dar una visión clara de la red, independientemente de la localización de los servidores.

Posteriormente, se diseñarán dos máquinas virtuales, que permitirán alojar los LVAP y sus controladores, a partir de una imagen base Debian 10 Minimal. En dichas máquinas se incluirán las herramientas, servicios y aplicaciones necesarias para

dotarlas de la funcionalidad deseada, además de permitir el acceso a los recursos físicos externos que la tecnología Wi-Fi requiere. La primera de ellas será un equipo virtual correspondiente a un punto de acceso Wi-Fi, dependiente de una interfaz inalámbrica física que actuará tanto como punto de acceso real como de LVAP. La segunda máquina cumplirá la función de controlador, necesario para el correcto funcionamiento de la SDWN. Las imágenes diseñadas deberán ser ligeras y eficientes para asegurar la flexibilidad y la escalabilidad del despliegue.

A continuación, se realizará un estudio de las posibilidades del despliegue y su gestión. Se diseñarán diferentes escenarios de red virtuales con el objetivo de observar las interconexiones, flujos de datos e interacciones entre los diferentes equipos, así como identificar y planificar soluciones para los diferentes inconvenientes que pudiesen presentarse. Este despliegue usará como base los equipos y la red previamente desarrollada para anteriores proyectos de investigación sobre SDWN.

El objetivo último del proyecto será lograr la integración de todos los objetivos anteriores para permitir un despliegue rápido y ligero de los equipos configurados y facilitar su gestión, realizando para ello una prueba de concepto. Posibilitaremos, de esta forma, facilitar el escalado de la red y la adición de nuevos equipos sin necesidad de modificar la topología de la red de forma significativa.

1.3. Estructura de la Memoria

La organización de la memoria es la que sigue:

- En este primer capítulo de introducción, se presenta una breve explicación de las tecnologías clave del proyecto, su estado actual, su problemática y el enfoque del proyecto de acuerdo con esta.

- El segundo capítulo, Conceptos, se dedica a una explicación de mayor profundidad acerca de las herramientas específicas utilizadas durante el proyecto, en concreto, una visión más detallada de las SDN y el funcionamiento de GNS3 como plano de gestión.

- El tercer capítulo enumera y proporciona una breve explicación sobre las herramientas mas relevantes que se han utilizado. Se centra, sobre todo, en el simulador de redes virtuales GNS3, su funcionamiento y el uso dado a los

servidores remotos durante la realización del proyecto. En este mismo apartado, se explica la funcionalidad de los templates y su uso para crear imágenes virtuales.

- El cuarto capítulo detalla las pruebas realizadas durante el desarrollo del trabajo con el objetivo de estudiar el despliegue óptimo de la red virtual.
- El quinto y último capítulo expone las conclusiones finales del proyecto y proporciona algunas ideas acerca de posibles mejoras que implementar en el despliegue.

Capítulo 2

Conceptos

Como se ha comentado anteriormente, el objetivo de este proyecto es el despliegue y posterior análisis de redes virtualizadas basadas en la tecnología Wi-Fi. En este apartado se detallarán conceptos que resultan necesarios, así como el estado del arte.

2.1. Redes definidas por *software* (SDN)

Las redes estándar están basadas, generalmente, en el uso de dispositivos cuya función en la red es fija, como un *switch* o un *router*. Las utilidades de estos equipos de red son divididas de manera lógica en tres planos: el de control, el de datos y el de gestión.

Los elementos de una red tradicional cumplen con funcionalidades de uno o varios planos, implementando protocolos correspondientes a dichos planos. Por el contrario, las redes definidas por *software* crean separaciones bien definidas entre estos planos. La división de estas funcionalidades en planos diferentes facilita la labor de gestión del administrador de la red [18].

Las redes definidas por *software* son un conjunto de técnicas que componen una arquitectura de red caracterizada por ser dinámica, fácilmente gestionable y adaptable [20]. Como se ha mencionado anteriormente, la arquitectura SDN queda dividida en diferentes planos como se observa en la siguiente figura:

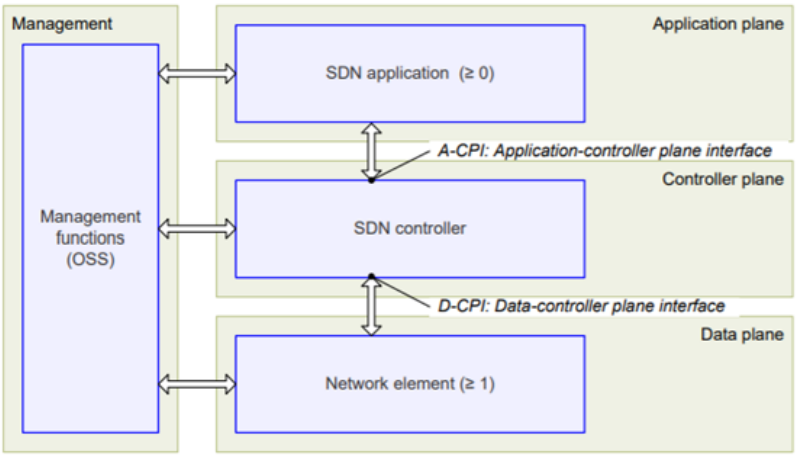

Figura 2.1: Planos en la arquitectura SDN [26]

El plano de datos engloba los recursos físicos y equipos de los que se compone la red, visibles y controlados por los correspondientes controladores. Cabe destacar que, a diferencia de las redes tradicionales donde los equipos que componen este plano tenían funcionalidades de datos y control, en las SDN sólo cumplirán labores de reenvío marcadas por los controladores, desligándolos en gran medida de la carga computacional que supone la toma de decisiones para el manejo de los flujos de datos, reduciendo así la latencia en estos nodos [13].

En segundo lugar, el plano de control se compone de uno o varios equipos controladores encargados de traducir las necesidades de la capa de aplicación y operar los correspondientes recursos del plano de datos [13, 18, 20]. El elemento más destacable de la red de control es el controlador, su introducción en la red limita la función de los *switch* a únicamente el reenvío de paquetes.

Las funciones de un controlador incluyen la monitorización constante del estado de los equipos, cálculo de los caminos óptimos para los flujos de datos y, basándose en la información obtenida en los procesos anteriores, actualizar las tablas de rutas de cada uno de los *switch* de la red. El protocolo de control (para comunicar los controladores con los equipos de red) más extendido en la actualidad es OpenFlow, puesto que permite la interoperabilidad entre equipos de diferentes fabricantes [6].

Finalmente, el plano de gestión engloba tareas que deben ser realizadas independientemente de los planos de control y datos. Como una función adicional y deseable, puede proporcionar una interfaz directa entre el gestor de la red y los diferentes equipos que componen el resto de los planos.

La función del plano de gestión monitoriza el correcto funcionamiento de la red y los equipos y permite al administrador de la red aplicar las configuraciones a las máquinas y proporcionar soporte al entorno. En el proyecto que se ha desarrollado, todas estas funciones del sistema de gestión de las redes de prueba desplegadas son de vital importancia.

2.2. Virtualización

En el capítulo de introducción se han mencionado las tecnologías básicas que fundamentan este proyecto: una de ellas es la virtualización. En esta sección, se detallarán las herramientas de virtualización utilizadas durante el desarrollo del trabajo, su funcionamiento y la problemática enfrentada.

La virtualización de máquinas no es posible sin la intervención de un hipervisor. Un hipervisor es un software que permite crear máquinas virtuales mediante el reparto lógico de los recursos del equipo anfitrión, como la memoria o la CPU(*Central Processing Unit*). Estos se subdividen en dos tipos diferentes según su funcionamiento.

Para comprender la virtualización, debe conocerse el funcionamiento básico de una CPU virtual y el papel principal de un hipervisor. En el hardware real, el sistema operativo (en inglés, *operative system*, OS) traduce los programas en instrucciones que son ejecutadas por la CPU física. En las máquinas virtuales sucede algo similar. Sin embargo, la diferencia principal entre ambas radica en que la vCPU (*Virtual Central Processing Unit*) está siendo simulada por el hipervisor y, por tanto, es este el que debe encargarse de traducir las instrucciones dirigidas a la vCPU y convertirlas en instrucciones para la CPU real.

Los hipervisores suelen clasificarse entre dos tipos: los de Tipo 1 se instalan directamente sobre el hardware y tienen comunicación directa con este, eliminando así la necesidad de un sistema operativo en el equipo Host. Los hipervisores de Tipo 2, al contrario que el 1, requieren que el equipo tenga instalado un sistema operativo ya que su funcionalidad consiste en dar la capacidad de virtualizar al sistema operativo base, ejemplos de hipervisores de Tipo 2 serían VirtualBox[1] o VMWare[2] [19].

[1]https://www.vmware.com/es.html
[2]https://www.virtualbox.org/

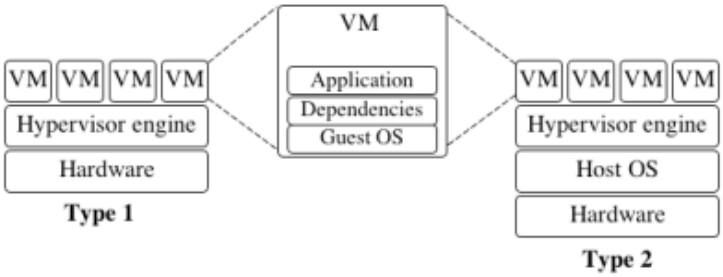

Figura 2.2: Tipos de hipervisores.

Los hipervisores utilizados en este proyecto son KVM y QEMU. KVM [3] es un hipervisor de código libre integrado en el kernel de Linux. Su tipo correspondiente no está del todo definido, podría considerarse de Tipo 1 ya que puede utilizar enteramente el kernel como un hipervisor, sin alterar en absoluto el sistema del host manteniéndose totalmente funcional, característica básica del Tipo 2. KVM permite emular por completo una CPU virtual, de forma que el hipervisor se encarga de recibir las instrucciones dirigidas a la vCPU y traducirlas para su ejecución por la CPU física.

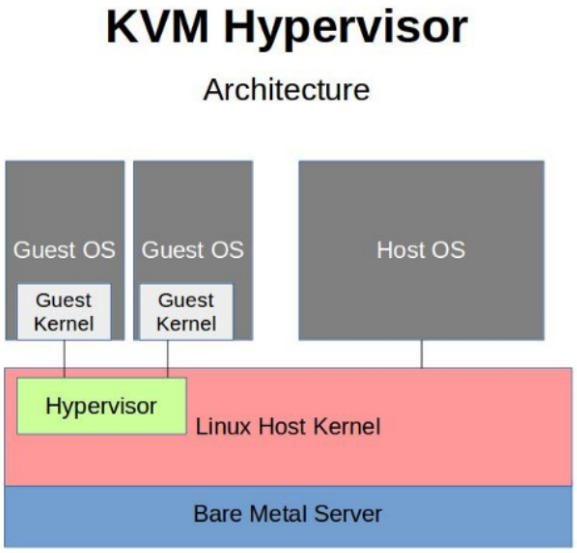

Figura 2.3: Arquitectura de KVM

Por otro lado, QEMU [4] es un hipervisor de Tipo 2 que realiza virtualización de hardware tales como: disco, VGA, PCI, etc. La funcionalidad principal de QEMU es ejecutar un sistema operativo sobre otro, por ejemplo, Windows sobre Linux [8]. Utiliza *dynamic binary translation* para permitir que un código diseñado para un procesador

[3]https://www.linux-kvm.org
[4]https://www.qemu.org/

específico pueda ser interpretado por uno de otra distribución [8].

En conclusión, QEMU puede usarse como hipervisor independiente, pero debido a que emula máquinas virtuales enteramente mediante software, puede llegar a ser lento e ineficiente. Por ello, suele utilizarse KVM en conjunto con QEMU, llamado comúnmente hipervisor QEMU/KVM.

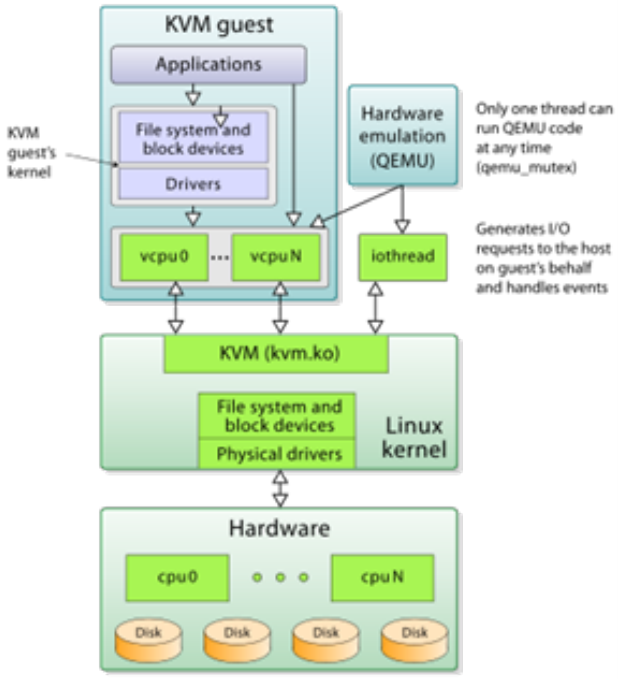

Figura 2.4: Arquitectura conjunta de KVM y Qemu

Este entorno presentado en la Figura 2.4 es el que se ha utilizado a lo largo del proyecto. Mediante el hipervisor KVM/QEMU se crean y lanzan máquinas virtuales con sistema Debian 10 Minimal. El hardware utilizado son tres equipos Intel NUC 5i3RYH cuyo sistema operativo es un Ubuntu Server versión 20.04 [5].

El papel de la virtualización en este trabajo no se limita a un equipo virtual cualquiera: se necesita crear un punto de acceso inalámbrico. Para ello, no basta con disponer de recursos como procesador, memoria o disco; también debe obtenerse el control de partes del hardware como interfaces de red o puertos del sistema. El equipo *Host* dispondrá de una o varias interfaces inalámbricas externas que actuarán como punto de acceso real. Al mismo tiempo, se creará una máquina virtual como punto de acceso virtual que, como se muestra en la figura 2.4, podrá capturar la interfaz para darle uso.

[5]https://ubuntu.com/server/docs

Capítulo 3

Herramientas

En este capítulo, se describen las herramientas principales que se han usado a lo largo del proyecto. Se dará una visión general de sus funcionalidades y el contexto de su uso en los escenarios que se han desarrollado.

3.1. Hostapd

Hostapd (*Host access point daemon*) es un *software* que permite la creación de un punto de acceso a partir de una interfaz de red inalámbrica. Implementa la gestión y configuración de puntos de acceso según el estándar IEEE 802.11 y IEEE 802.1X/WPA/WPA2/EAP Authenticators [3].

Está diseñado para como un programa que opera en segundo plano y actúa como gestor del punto de acceso y sus funcionalidades de seguridad y autenticación. Es una herramienta completa y sus desarrolladores siguen proporcionando soporte para las nuevas tecnologías.

El papel de esta herramienta en el trabajo consiste en una primera aproximación a la creación de puntos de acceso y el estudio de su funcionamiento. Sin embargo, no implementa funcionalidades típicas de una SDN, por lo que será sustituido por otro software de control de puntos de acceso en las etapas finales del proyecto.

3.2. DNSmasq

A la hora de crear puntos de acceso, la red debe contar con algún mecanismo de asignación de direcciones IP a las STAs que se asocien. Una practica común es disponer de un servidor DHCP (*Dynamic Host Configuration Protocol*) en la red privada para

asignar de manera dinámica estas direcciones. Para ello, se ha utilizado la herramienta dnsmasq [1].

Dnsmasq es un software que proporciona, entre otros, un servidor DHCP configurable para la red. Es una herramienta ligera que consume pocos recursos del sistema, haciéndola ideal para las máquinas virtuales que se implementan en este proyecto. Otro de los motivos por los que se selecciona esta aplicación es la posibilidad de configurar un servidor DNS en la red privada. En las redes en proceso de crecimiento, como es este caso de estudio, se suele instalar un servidor DNS básico para acelerar el proceso de consulta y dnsmasq daría soporte a esa posibilidad.

3.3. Odin Wi5

Wi5 es un tipo de arquitectura de programación de espectro radio para la gestión de recursos en las bandas sin licencia. El proyecto busca introducir un plano de control para el espectro que permite una configuración flexible de los recursos propios de una red inalámbrica [12].

Como parte de Wi5 se desarrolla el protocolo Odin, con el objetivo de gestionar los intercambios de información entre controladores y puntos de acceso, llamados OdinAP y Odin Controller. La funcionalidad de esta herramienta es la creación de Light Virtual Access Points (LVAP). Un LVAP es una abstracción que permite realizar handovers transparentes al usuario, dando así una solución al problema de la movilidad Wi-Fi. Su funcionamiento, a grandes rasgos, es el siguiente. Cuando un dispositivo se asocia por primera vez con un punto de acceso, el controlador de la red crea un LVAP para él. Este LVAP contiene parámetros de comunicación exclusivos para la STA asociada. Cada punto de acceso tendrá un LVAP por cada dispositivo asociado con él. Este LVAP puede ser traspasado entre distintos puntos de acceso físicos, por lo que la STA verá constantemente el mismo AP aunque se esté desplazando y cambie su asociación [14, 12].

En este trabajo se implementarán los equipos Odin Wi5 para actuar como puntos de acceso y controlador de la SDN y se estudiarán sus posibilidades de despliegue y gestión.

3.4. GNS3: Graphic Network Simulator

GNS3 (Graphic Network Simulator) es un simulador de redes que permite diseñar diferentes topologías de red y ejecutar simulaciones sobre ellas.

Es una herramienta open-source, su código gratuito es descargable desde Github y multiplataforma.

El simulador GNS3 permite emular dispositivos de gran cantidad de distribuciones, como switches virtuales de Cisco, Linux/GNU, instancias de Docker, ASAs Cisco, Brocade vRouters, entre otros.

GNS3 utiliza dos componentes para su funcionamiento, el cliente GNS3 que presenta la interfaz gráfica desde la que se diseñan y gestionan las diferentes topologías y configuran los modos de operación y uno o varios servidores, que se ocupan de alojar y correr las diferentes instancias virtuales para la simulación.

3.4.1. Servidores GNS3

Existen varios tipos de servidores que pueden ser utilizados, de forma simultánea si es necesario, para emular los diferentes equipos del escenario:

- **Servidor Local**: durante la instalación del cliente GNS3, adicionalmente, se instala en el equipo físico un servidor GNS3 que permitirá correr las imágenes compatibles con la distribución en la que se haya instalado. En este proyecto, la interfaz gráfica de GNS3 se encuentra instalada sobre un sistema Windows 10, por tanto, este servidor local se lanza como un proceso de Windows y puede ser utilizado para ejecutar instancias IOS y Cisco basadas en Dynamips.

- **GNS3 VM**: GNS3 proporciona un sustituto para el servidor local en forma de máquina virtual. Ésta incluye una máquina virtual Ubuntu, ejecutada sobre VMWare Workstation, en la que se ha instalado el servidor de GNS3. La ventaja sobre el servidor local reside en que la distribución Ubuntu incluye el hipervisor KVM, necesario para utilizar maquinas GNU/Linux emuladas mediante QEMU de forma óptima.

- **Servidor Remoto**: se utiliza un servidor GNS3 en un equipo remoto, instalado en una distribución Windows o GNU/Linux. Utilizar servidores remotos presenta la ventaja de reducir la carga de procesado en las máquinas reales que aloja el cliente de GNS3 y la máquina virtual de GNS3.

Durante la realización de este proyecto se han usado servidores de los tres tipos, con el fin de probar los diferentes rendimientos de las topologías dependiendo del formato de servidor sobre el que corran. Sin embargo, uno de los objetivos del proyecto consiste en permitir desplegar servidores GNS3 remotos en localizaciones diferentes, alejados del equipo de gestión, manteniendo la funcionalidad. Por este motivo el servidor remoto es el más recomendado. El proceso de configuración de los diferentes servidores se encuentra detallado en el Anexo IV.

3.4.2. Templates

Como se ha mencionado anteriormente, una de las ventajas de la virtualización es la facilidad que presenta para escalar y adaptar diferentes topologías de forma rápida y eficaz. Aunque la dependencia de los equipos hardware continúa presente, la dificultad en el despliegue de nuevas máquinas se ve reducida en gran medida.

GNS3 incluye una funcionalidad para facilitar el proceso de creación de nuevos equipos virtuales, los Templates. Estos permiten vincular un archivo de imagen virtual con determinados parámetros de arranque para ser cargados en un servidor concreto. La interfaz del cliente almacena los templates creados en su propia memoria, mientras que las imágenes virtuales de cada máquina se encuentran en los propios servidores GNS3. El Template proporciona diversas opciones de configuración, tanto básicas como el tipo de consola, número de interfaces, como opciones de modificación de las cadenas de arranque. En este proyecto se han simulado máquinas QEMU, en este emulador se determinan las configuraciones iniciales de la maquina a simular mediante un script de lanzamiento que deberá ser modificado en caso de desearse otro método de operación.

Los Templates de GNS3 presentan una opción que modifica dicho script y las dependencias necesarias de manera simplificada.

Las máquinas basadas en templates se ejecutan sobre los servidores de GNS3 de dos maneras diferentes: como Snapshots de la imagen original o sobreescribiendo la imagen fuente de manera permanente, modificándose el template para cualquier escenario de red creado posteriormente.

El primer modo de operación almacena las configuraciones que han sido añadidas al equipo virtual en un fichero de imagen diferente que se ejecutará tras cargar la imagen fuente cada vez que se arranque una máquina. Esta herramienta es útil para conservar el estado de una máquina durante un tiempo, aunque presenta limitaciones, como dificultades para ser trasladada a un servidor o incorporada a una nueva

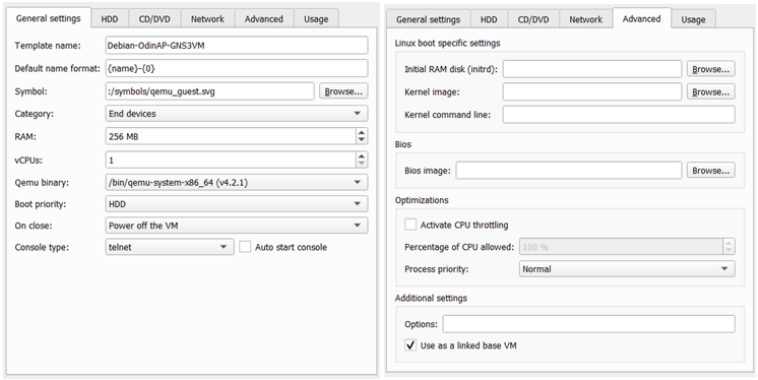

Figura 3.1: Opciones de configuración de un template

topología. Como solución, puede vincularse a la imagen fuente para conservar los nuevos parámetros del equipo a largo plazo, así como facilitar la migración de la máquina a un nuevo servidor a la vez que el template queda actualizado.

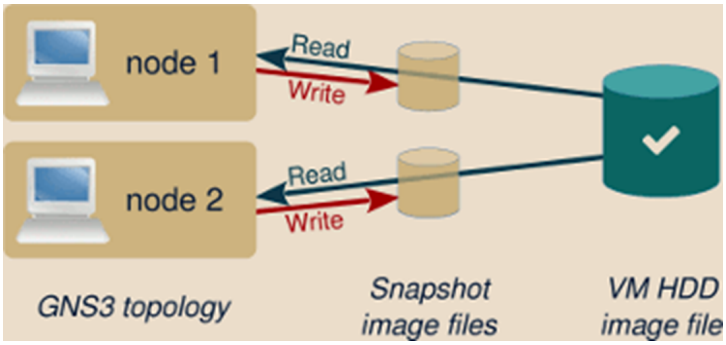

Figura 3.2: Uso de *snapshots*

Por otro lado, puede ejecutarse la imagen virtual asociada al template de manera que los cambios que se apliquen sobre el original sobrescriban la imagen. Cabe destacar que este modo de operación del enlace entre el template y la imagen solo permite que se simule una única maquina en todos los escenarios de red que gestione el agente desde el cliente GNS3 mientras permanezca activo.

A lo largo del proyecto, se crearon diversos templates para las máquinas Debian utilizando esta opción. Debido a que los equipos elegidos eran ligeros, se decidió configurar sobre ellos diversas herramientas para ser usadas durante las primeras fases de pruebas: Hostapd, que permite la configuración de un punto de acceso a una máquina virtual con acceso a una interfaz inalámbrica, OpenVSwitch para permitir la operación de la red como una SDN mediante el protocolo Openflow, así como actualizaciones del tamaño en memoria de la imagen, modificaciones en versiones de kernel y variables del sistema. Todas las configuraciones se encuentran detalladas en el Anexo II.

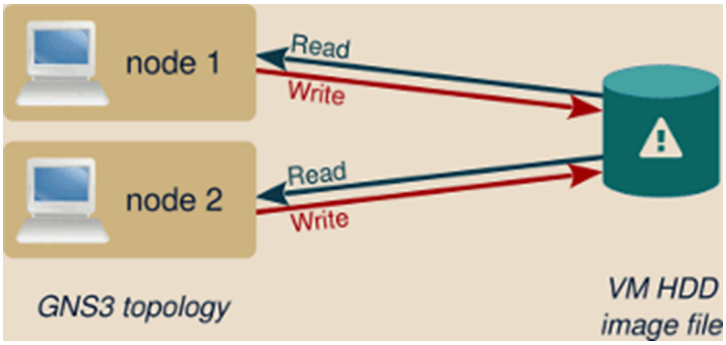

Figura 3.3: Escritura sobre la imagen base

Una vez se diseñó una máquina acorde con las necesidades del proyecto, se actualizó la imagen, y por tanto el template, permitiendo su uso simultáneo en las diferentes topologías.

3.4.3. GNS3 como herramienta de gestión

Generalmente, las funciones enmarcadas en el plano de gestión se llevan a cabo mediante comunicaciones entre el usuario y los dispositivos de gestión basadas en protocolos como SNMP, SSH, HTTP.

En este proyecto el plano de gestión engloba las funciones de comunicación del usuario gestor tanto con las máquinas virtuales desplegadas, como con los servidores GNS3 configurados en el hardware.

Por un lado, la configuración de los servidores desplegados debe realizarse mediante conexiones SSH o acceso físico al equipo. Una vez sean configurados, las credenciales definidas durante el proceso se utilizarán para dar acceso a cualquier gestor a dichos servidores y los equipos simulados sobre este. Ver Anexo IV.

Los equipos virtuales pertenecientes a la red son lanzados en remoto sobre un servidor previamente configurado. Mediante el cliente GNS3 un usuario es capaz de lanzar, gestionar y configurar todos los dispositivos simulados. De este modo, se consigue centralizar en una única interfaz remota la gestión y monitorización de diversos escenarios distribuidos entre diferentes equipos.

GNS3 permite establecer credenciales de acceso en los servidores remotos para garantizar la seguridad en las comunicaciones entre el administrador de la red y los equipos. Esta opción permite establecer el plano de gestión en cualquier ubicación, basta con conocer el usuario y la contraseña para obtener acceso a las máquinas y

servidores.

Dado que el objetivo de este proyecto es el estudio de las posibilidades de un despliegue de red y la capacidad de una herramienta de análisis de red para gestionarlo, GNS3 en este caso, resulta de interés profundizar ligeramente en sus métodos de funcionamiento.

Para la realización de las tareas de gestión se utiliza el protocolo HTTP [23]. Las comunicaciones entre el cliente de GNS3, gestor en este caso, y los servidores GNS3 remotos, agentes a gestionar, se realizan basándose en una HTTP REST API. La base de la interacción entre los servidores remotos y la interfaz de cliente de GNS3 se realiza a través de intercambios de tramas paquetes HTTP basados en JSON (JavaScript Object Notation) [25] de acuerdo con la programación de la API ya mencionada [7], las tramas contienen mensajes específicos en los que se referencian ubicaciones de memoria de los servidores, junto a instrucciones de operación.

Representational State Transfer (REST) es un tipo de arquitectura que proporciona guías y directrices para diseñar servicios web fácilmente escalables. Utiliza instrucciones de HTTP para realizar llamas entre las diferentes máquinas. Comúnmente, estas instrucciones señalizan acciones de lectura, creación, actualización y borrado. En una API REST basada en HTTP, estas instrucciones se corresponden, respectivamente, con GET, POST, PUT, DELETE.

En concreto, la arquitectura de la API de GNS3 es la mostrada en la figura siguiente.

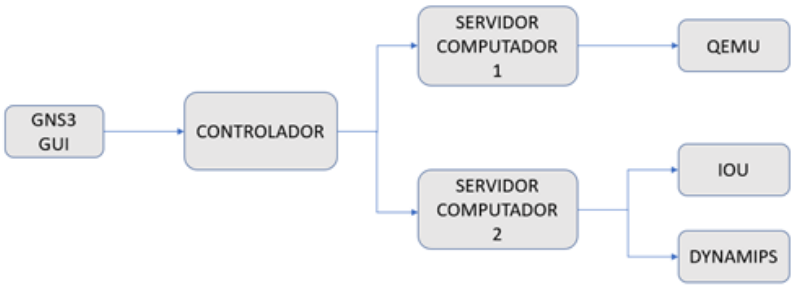

Figura 3.4: Diagrama de la arquitectura GNS3

Se encuentra dividida en cuatro partes [2]. En primer lugar, la interfaz de GNS3 (GNS3 GUI) se ocupa de mostrar la topología de un proyecto representada en un canvas, permitiendo al usuario realizar acciones y gestionar las máquinas del modelo a través del envío de peticiones de la API al controlador. Esta interfaz de usuario no tiene por qué encontrarse en el mismo equipo que los servidores.

Por otro lado, el controlador es el encargado de gestionar las peticiones recibidas desde el cliente GNS3, así como de controlar el estado de cada proyecto. Este controlador corre sobre los servidores GNS3. Por último, los computadores manejan los emuladores necesarios para mantener las máquinas virtuales, creando instancias de estos para cada nodo.

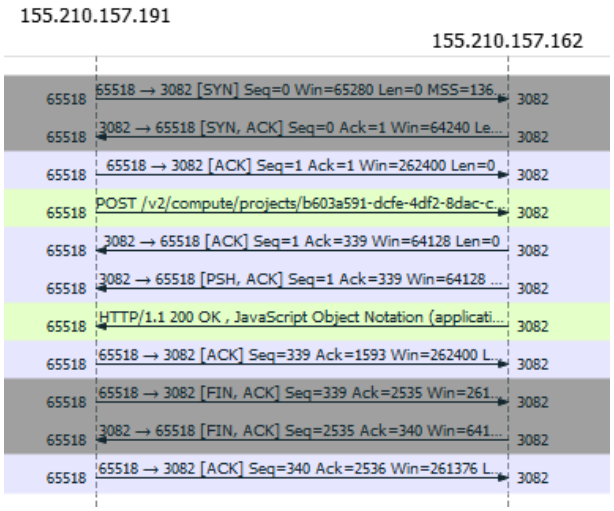

Figura 3.5: Intercambio de tramas en el arranque de una VM

La Figura 3 muestra un ejemplo del intercambio realizado entre el equipo gestor y el servidor 155.210.157.162 en el momento en que el usuario introduce la instrucción para el arranque de una máquina virtual.

El paquete HTTP enviado por el cliente GNS3 contiene la instrucción POST de la ruta en la que esta almacenado el archivo de imagen de una de las máquinas, indicando así al controlador que debe crearse una instancia de QEMU para lanzar la máquina. Como medida de seguridad, los intercambios HTTP deben acompañarse de las credenciales para permitir el acceso a los servidores.

```
▾ Hypertext Transfer Protocol
  › POST /v2/compute/projects/b603a591-dcfe-4df2-8dac-cfea23d147ab/qemu/nodes/bdead8d3-f29b-400c-90be-f2bf8391d623/start HTTP/1.1\r\n
    Host: 155.210.157.162:3082\r\n
    content-type: application/json\r\n
    Accept: */*\r\n
    Accept-Encoding: gzip, deflate\r\n
    User-Agent: Python/3.6 aiohttp/3.6.2\r\n
  ▾ Authorization: Basic cHJveWVjdG86cHJveWVjdG8=\r\n
      Credentials: proyecto:proyecto
  › Content-Length: 0\r\n
```

Figura 3.6: Desglose de la trama HTTP

Capítulo 4

Escenarios de Pruebas

En este apartado, se detallarán los diferentes escenarios de pruebas diseñados, su objetivo y motivación, además de las conclusiones obtenidas a raíz de cada uno de ellos.

Para lanzar y ejecutar los escenarios se han preparado hasta tres servidores remotos GNS3. Cada uno de ellos se encuentra corriendo en equipos NUC con sistema operativo Ubuntu Server 20.04. Estos servidores son los encargados de almacenar las respectivas imágenes virtuales y configuraciones de cada escenario.

4.1. Escenario 0: Funcionamiento de GNS3 y errores de captura

Como paso previo a desarrollar los escenarios definitivos, debe estudiarse el funcionamiento de la virtualización y la carga de tráfico generada por la interacción de los equipos virtuales con el escenario de red real (subred 155.210.157.0/24) y el equipo en el que se encuentre corriendo la interfaz gráfica de GNS3.

Adicionalmente, en el diseño de la topología final, se ha decidido que el cliente GNS3 y su interfaz gráfica implementen las funcionalidades del plano de gestión de la SDN. A través del cliente y las consolas serie configuradas podrán accederse, configurar y gestionar todos y cada uno de los equipos que componen la topología de una forma sencilla, rápida y directa.

A este fin se han diseñado varias topologías sencillas para medir correctamente el tráfico y estudiar los inconvenientes que pudiesen surgir, además de documentar como gestiona el intercambio y la comunicación entre equipos virtuales y virtuales-reales el sistema de GNS3.

4.1.1. Escenario 0.1: Acceso a Internet

Este escenario consiste únicamente de dos equipos alojados en los servidores 155.210.157.160 y 161 respectivamente, conectados a dos nodos *Cloud* para permitir la salida al exterior. Dotar a los equipos virtuales de conectividad a Internet es un paso básico para poder continuar el desarrollo de las pruebas.

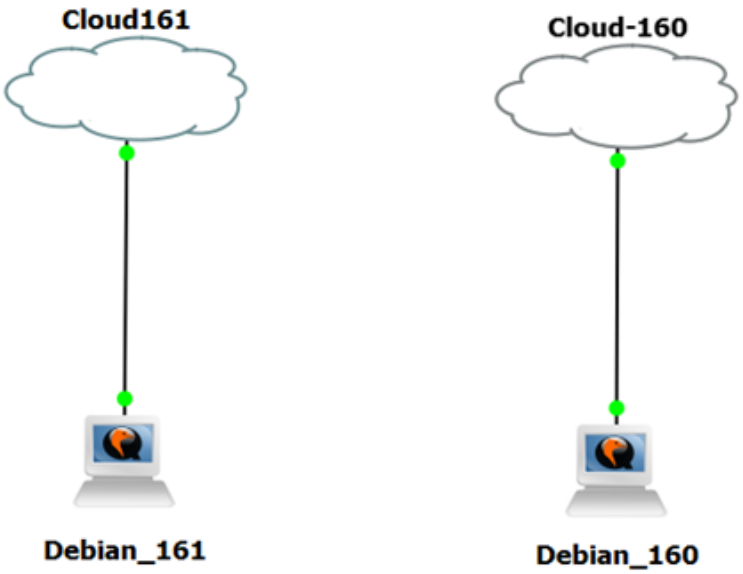

Figura 4.1: Escenario de red 0.1: Acceso a Internet

Se captura en el enlace virtual entre el equipo Debian y el *Cloud* 161, durante la realización de un *ICMP Echo Request* entre ambas máquinas Debian. Esta captura, utilizando Wireshark [5], se realiza de forma remota en el servidor 155.210.157.161, por lo que el cliente GNS3 realiza solicitudes HTTP para que le sean reenviadas dichas tramas capturadas en remoto.

15	0.085250	155.210.157.189	155.210.157.161	HTTP/J…	60	GET /v2/compute/projects/9714d08d-fb7e-4cb3-b240-dc99b37db8f7/cloud/nodes/a829bf20-257e-4e2d-ab4e-2c09a4cf09d8/
16	0.085355	155.210.157.161	155.210.157.189	TCP	54	3080 → 59212 [ACK] Seq=1 Ack=434 Win=64128 Len=0
17	0.088586	155.210.157.161	155.210.157.189	TCP	358	3080 → 59212 [PSH, ACK] Seq=1 Ack=434 Win=64128 Len=304 [TCP segment of a reassembled PDU]
18	0.088686	155.210.157.161	155.210.157.189	TCP	2145	3080 → 59212 [PSH, ACK] Seq=305 Ack=434 Win=64128 Len=2091 [TCP segment of a reassembled PDU]
19	0.088858	155.210.157.161	155.210.157.189	TCP	2596	3080 → 59212 [PSH, ACK] Seq=2396 Ack=434 Win=64128 Len=2542 [TCP segment of a reassembled PDU]
20	0.111141	155.210.157.189	155.210.157.161	TCP	60	59212 → 3080 [ACK] Seq=434 Ack=1665 Win=262400 Len=0
21	0.112192	155.210.157.189	155.210.157.161	TCP	60	59212 → 3080 [ACK] Seq=434 Ack=3756 Win=262400 Len=0
22	0.153295	155.210.157.189	155.210.157.161	TCP	60	59212 → 3080 [ACK] Seq=434 Ack=4938 Win=261120 Len=0

Figura 4.2: Inicio de una captura remota en GNS3

Sin embargo, estas tramas reenviadas al equipo de gestión también son capturadas y reenviadas, provocando la aparición de una gran cantidad de tráfico redundante, e incluso tormentas de tráfico, que dificulta la realización de capturas y, en un escenario más complejo, podría generar problemas de sobrecarga.

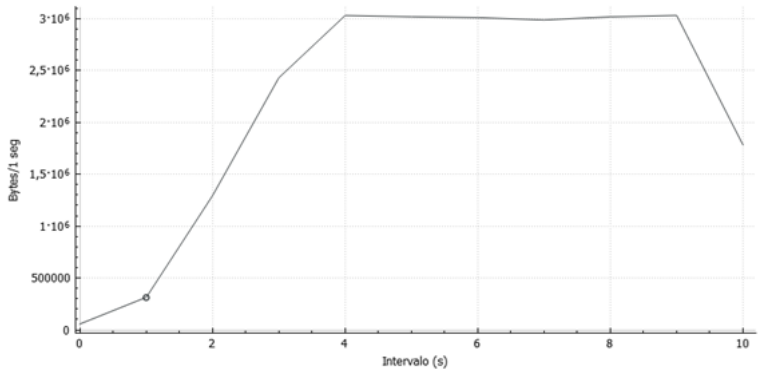

Figura 4.3: Tráfico en el enlace Debian y *Cloud*

Casi en su totalidad, se observan tramas TCP realimentadas:

Protocolo	Porcentaje de paquetes	Paquetes	Porcentaje de bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	16975	100.0	23961642	18M	0	0	0
Ethernet	100.0	16975	1.0	237650	179k	0	0	0
Logical-Link Control	0.0	5	0.0	190	143	0	0	0
Internet Protocol Version 4	99.9	16957	1.4	339140	256k	0	0	0
User Datagram Protocol	0.0	4	0.0	32	24	0	0	0
Transmission Control Protocol	99.8	16943	97.4	23332928	17M	16723	23169896	17M
Internet Control Message Protocol	0.1	10	0.0	640	483	10	640	483
Address Resolution Protocol	0.1	13	0.0	598	452	13	598	452

Figura 4.4: Desglose de las tramas capturadas

Debido a la naturaleza del objetivo de este proyecto, las capturas de tráfico son una herramienta absolutamente necesaria, por lo que este problema debe ser solucionado. Para ello se introduce una pequeña modificación en el escenario siguiente.

4.1.2. Escenario 0.2: Solución tráfico realimentado

Con el objetivo de filtrar el tráfico que ingresa en la red virtual, se ha introducido un equipo *switch* en la frontera de la subred para evitar que el tráfico redundante modifique las capturas en el interior del escenario virtual.

Este dispositivo *switch* podría ser sustituido por cualquier otro equipo que filtrase el tráfico no dirigido a las máquinas virtuales. Mediante esta modificación, se permite mantener el intercambio de paquetes en los enlaces interiores libre de la redundancia previamente comentada.

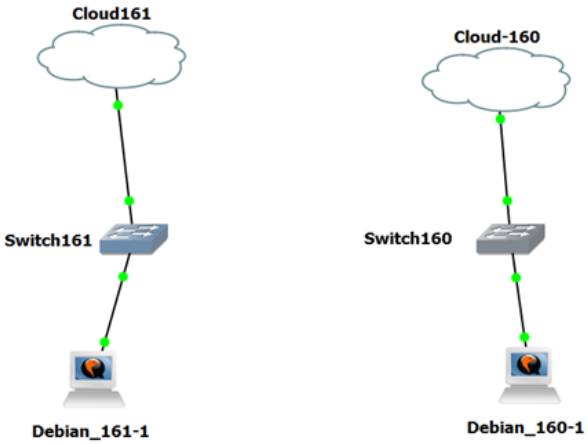

Figura 4.5: Escenario de red 0.2

Como muestra la siguiente gráfica, el tráfico realimentado continúa creciendo hasta el segundo 3, cuando se detuvo la captura en la interfaz entre el nodo *Cloud* y el *Switch*. Este tráfico continuaría incrementándose mientras durase la captura:

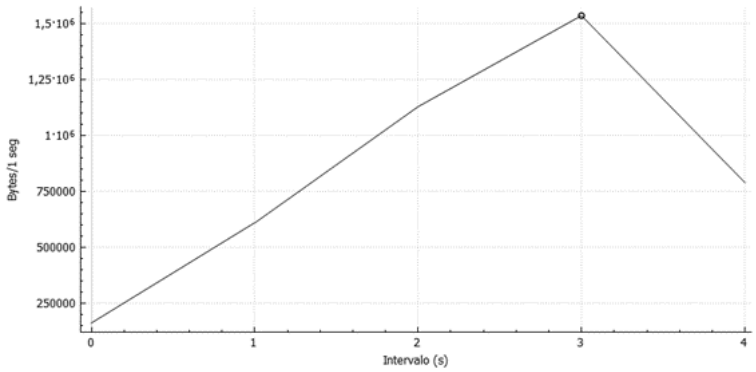

Figura 4.6: Tráfico en el enlace *Switch-Cloud*

Simultáneamente, se capturó en el enlace entre la máquina Debian161 y el *switch* para comprobar el correcto funcionamiento del filtrado de tráfico.

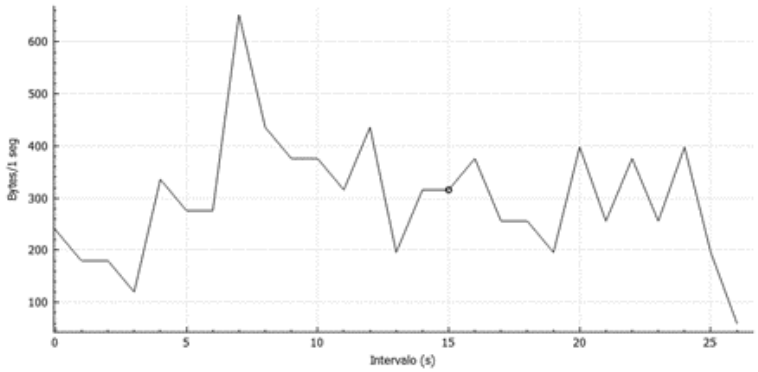

Figura 4.7: Tráfico en el enlace *switch*-Debian

Una vez solucionado el problema se podrán realizar capturas en los enlaces virtuales sin que se vean afectadas por el tráfico resonante. Sin embargo, para analizar el tráfico en los enlaces entre las interfaces virtuales y reales, nodo *Cloud* a *switch* en este caso, se deberá capturar la interfaz real en local desde el servidor correspondiente.

4.1.3. Comunicación entre cliente y servidores

El cliente GNS3 (155.210.157.190 en este caso), es decir, la interfaz gráfica, es la herramienta que se utiliza para coordinar el funcionamiento de los servidores remotos GNS3 y los equipos virtuales. Por ello, en cualquier captura realizada en las interfaces de red de los servidores se pueden observar paquetes TCP que son intercambiados constantemente entre el cliente GNS3 y los servidores.

```
2749 20.459908    155.210.157.164    155.210.157.161    TCP      60 55345 → 3080 [ACK] Seq=434 Ack=12332360 Win=1057024 Len=0
2750 20.459991    155.210.157.161    155.210.157.164    TCP   42394 3080 → 55345 [PSH, ACK] Seq=12535300 Ack=434 Win=64128 Len=42340
2751 20.460024    155.210.157.161    155.210.157.164    TCP   21954 3080 → 55345 [PSH, ACK] Seq=12577640 Ack=434 Win=64128 Len=21900
2752 20.460092    155.210.157.164    155.210.157.161    TCP      60 55345 → 3080 [ACK] Seq=434 Ack=12342580 Win=1057024 Len=0
2753 20.460092    155.210.157.164    155.210.157.161    TCP      60 55345 → 3080 [ACK] Seq=434 Ack=12352800 Win=1057024 Len=0
```

Figura 4.8: Tramas TCP de control GNS3

Los servidores, en este caso el 155.210.157.161, escuchan y responden las transmisiones del cliente en el puerto 3080. Este puerto habrá sido definido previamente en el fichero de configuración de cada servidor.

Estas comunicaciones de control son constantes, aunque ninguna máquina se encuentre corriendo en el servidor. Sin embargo, la cantidad de tráfico se incrementa cuando hay un mayor número de máquinas corriendo.

Protocolo	Porcentaje de paquetes	Paquetes	Porcentaje de bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	58	100.0	5126	4054	0	0	0
Ethernet	100.0	58	15.8	812	642	0	0	0
Logical-Link Control	8.6	5	3.7	190	150	0	0	0
Internet Protocol Version 4	53.4	31	12.1	620	490	0	0	0
User Datagram Protocol	15.5	9	1.4	72	56	0	0	0
Transmission Control Protocol	37.9	22	27.7	1422	1124	11	220	174
Address Resolution Protocol	37.9	22	19.7	1012	800	22	1012	800

Protocolo	Porcentaje de paquetes	Paquetes	Porcentaje de bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	208	100.0	13144	10k	0	0	0
Ethernet	100.0	208	22.2	2912	2284	0	0	0
Logical-Link Control	2.4	5	1.4	190	149	0	0	0
Internet Protocol Version 6	0.5	1	0.3	40	31	0	0	0
Internet Protocol Version 4	90.4	188	28.6	3760	2949	0	0	0
Transmission Control Protocol	90.4	188	35.7	4688	3677	177	3540	2777
Address Resolution Protocol	6.7	14	4.8	626	491	14	626	491

Figura 4.9: Tramas intercambiadas sin equipos activos (arriba) frente a dos equipos corriendo(abajo)

De forma análoga, los equipos virtuales también se comunican con el cliente, enviando tráfico de control:

```
449 10.404102   155.210.157.161   155.210.157.190   TCP   54 5900 → 58779 [ACK] Seq=1217 Ack=501 Win=502 Len=0
450 10.457959   155.210.157.190   155.210.157.161   TCP   64 58786 → 5902 [PSH, ACK] Seq=111 Ack=82503 Win=515 Len=10
451 10.458030   155.210.157.161   155.210.157.190   TCP   54 5902 → 58786 [ACK] Seq=82503 Ack=121 Win=502 Len=0
454 10.499889   155.210.157.161   155.210.157.190   TCP   54 5901 → 58782 [ACK] Seq=1217 Ack=501 Win=502 Len=0
```

Figura 4.10: Tráfico de control entre máquinas virtuales y cliente

La configuración por defecto de GNS3 asigna puertos TCP entre el número 5000 y el 10000 para los intercambios con el equipo del cliente.

El rango de puertos que utilizarán las máquinas virtuales puede modificarse desde el menú de preferencias de GNS3.

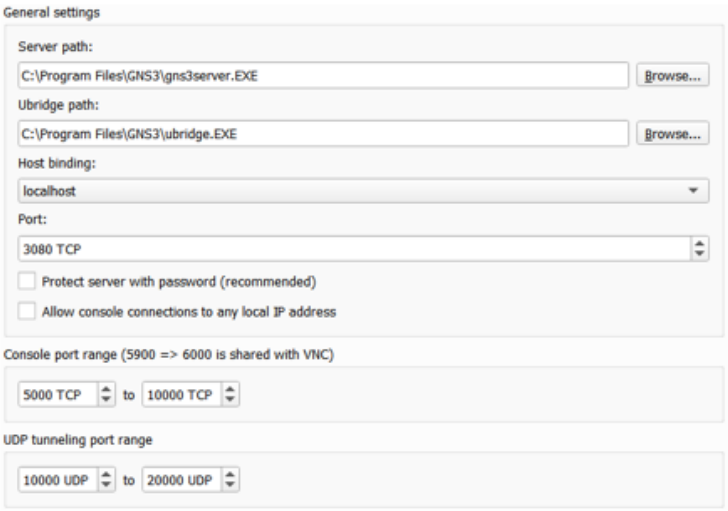

Figura 4.11: Menú de preferencias del servidor

4.1.4. Escenario 0.3: Túneles y tráfico UDP.

En los escenarios estudiados previamente, la comunicación entre los dispositivos virtuales de diferentes servidores se realizaba a través del nodo *Cloud*, es decir, mediante la red real. Sin embargo, puede ser de utilidad establecer enlaces directos entre varios nodos de la red virtual independientemente del servidor en que se encuentren corriendo.

Para permitir la comunicación entre ellos, GNS3 utiliza túneles UDP construidos entre los 2 equipos. Las tramas Ethernet son encapsuladas en datagramas UDP y enviadas a través del túnel, desencapsulándose en el equipo final.

A fin de estudiar estos intercambios, se ha modificado el escenario anterior para incluir conexiones directas entre equipos de diferentes servidores.

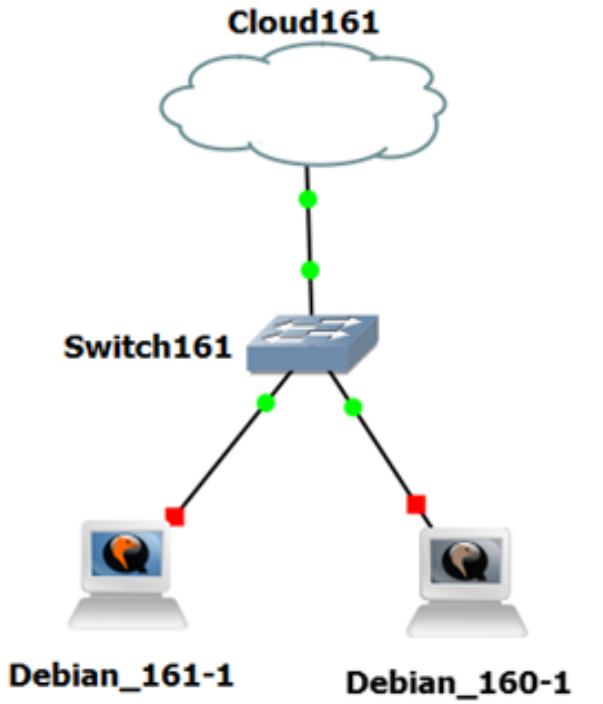

Figura 4.12: Escenario de red 0.3: Túneles UDP

Como prueba, se realiza un *ICMP Echo Request* desde el equipo Debian_160, cuya interfaz de red tiene la dirección 155.210.157.168, hacia el Debian_161, con dirección 155.210.157.170.

Capturando el tráfico en el enlace que comunica la máquina Debian_160 con el *switch*, se observa correctamente el intercambio de paquetes:

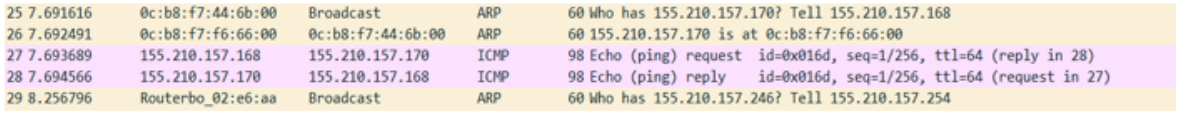

Figura 4.13: Intercambio de paquetes en la red virtual

Sin embargo, el tráfico visto en la interfaz real del servidor 160 corresponde únicamente a paquetes TCP, debidos al flujo de control explicado anteriormente, ARP (*Address Resolution Protocol*, que no son encapsulados y enviados por el túnel y finalmente datagramas UDP, en los que se transportan los paquetes ICMP en este caso.

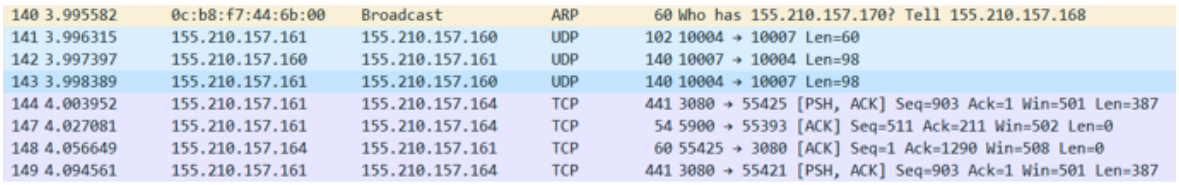

Figura 4.14: Paquetes capturados en la interfaz real

Los datagramas UDP recibidos en un servidor son desencapsulados antes de encaminarse a la red virtual, de esta forma el encapsulado y la transmisión en los túneles se realiza de manera transparente a la red virtual en GNS3.

4.2. Escenario 1: Implementación de un punto de acceso inalámbrico

Este escenario es el primero directamente orientado al despliegue de una red virtual inalámbrica. En él se ilustra el proceso de adquisición de una interfaz inalámbrica y la configuración de un punto de acceso virtual en una máquina Debian. Como ya se ha comentado, un punto de acceso vitual necesita disponer de un elemento *hardware* que funcione como adaptador inálambrico.

La topología diseñada es simple, consta únicamente de 3 equipos: una máquina Debian 10, un *switch* y un nodo *Cloud*.

En primer lugar, se ha delegado el control al equipo Debian de un puerto USB al que se ha conectado un adaptador inalámbrico. El punto clave de este escenario consiste en utilizar dicho adaptador para crear un punto de acceso Wi-Fi. Este paso requiere una configuración algo más compleja que los demás de este apartado. En primer lugar,

deben obtenerse los identificadores del adaptador Wi-Fi para seleccionar el *hardware* en concreto. Posteriormente deben modificarse las reglas de uso de PCI/USB del sistema para permitir el acceso al puerto en que se ha conectado el dispositivo. Estas reglas deben modificarse cuidadosamente puesto que pueden crear agujeros de seguridad en el sistema. Una vez se encuentra disponible la interfaz inalámbrica, se ha utilizado la herramienta Hostapd para configurar un punto de acceso. Todas las configuraciones necesarias se encuentran detalladas en el Anexo III.

Por simplicidad, los equipos virtuales de esta topología se encuentran corriendo en el mismo servidor GNS3, lo que permite observar de manera más sencilla el tráfico intercambiado, evitando los intercambios encapsulados de GNS3 en la red real.

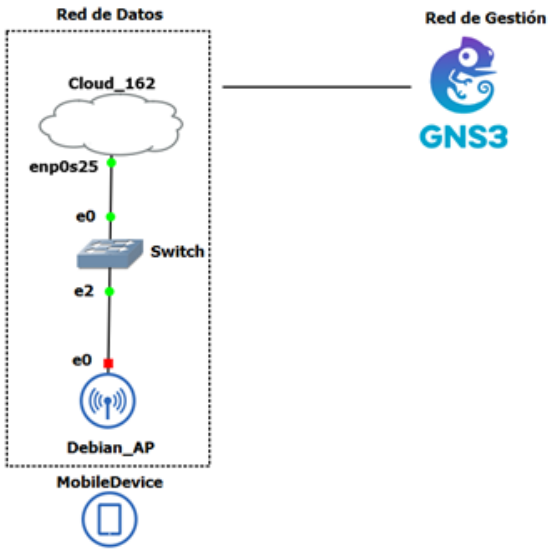

Figura 4.15: Escenario de red 1: Punto de Acceso

Es de interés resaltar la disposición de los planos de las redes definidas por software en los diseños de la red. En este primer escenario, el plano de gestión queda implementado por la interfaz de GNS3, desde la que se ha configurado cada uno de los equipos, así como monitorizado el tráfico que permite la depuración del escenario por parte del administrador.

Por otro lado, los equipos incluidos conforman la red de datos. El nodo *Cloud* actúa como interfaz de salida de la red virtual, proporcionando conexión a Internet al resto de equipos localizados en la red, además de a los equipos que se vinculen en un futuro al punto de acceso. Este tráfico será reenviado a través de la red de datos hacia el exterior por el nodo *Cloud*. El equipo *switch* se ha incluido por dos motivos,

el primero para actuar de frontera entre el tráfico de la red real y la virtual, evitando realimentaciones. Asimismo, debido a que en este momento el NUC que corre el servidor de GNS3 tiene disponibles únicamente dos interfaces, la inalámbrica utilizada por el punto de acceso y la física enp0s25, accesible desde el nodo *Cloud*, se requiere un equipo conmutador para permitir la conexión con varios equipos a la interfaz. Por último, la máquina Debian que actúa como punto de acceso a la red de datos, a través del que los dispositivos obtendrán acceso a la red de datos que se ha diseñado. Este formato de red permite añadir fácilmente nuevos puntos de acceso para escalar la red de datos a las dimensiones que se requieran. El plano de control aún no ha sido incluido en este escenario, se añadirá y discutirá su integración con el resto de la red en futuros escenarios.

Para terminar, los dispositivos móviles que se asocien con el AP deben recibir una dirección IP de la red para poder iniciar comunicaciones. En caso de que la red real sobre la que se construye la red virtual disponga de un servidor DHCP, este será el encargado de asignarlas. Si por el contrario no existe un DHCP, deberá configurarse uno en cualquier ubicación de la red. Dado que en este punto aún no se ha incluido un controlador, el servidor DCHP se configura en el punto de acceso utilizando la herramienta *dnsmasq* [1].

4.3. Escenario 2: Integración plano de control

4.3.1. Escenario 2.1

Una vez se ha decidido la disposición de los planos de control y gestión, debe introducirse la red de control. Esta red contendrá los equipos controladores que diseñarán las tablas de encaminamiento y regularán los flujos de tráfico en la red de datos.

Debido al amplio abanico de opciones, en este escenario se estudiarán las diversas posibilidades de implementación de la red de control. Como primera aproximación se ha incluido el controlador en la misma red que el plano de datos, pero en un servidor diferente. La red de control se vale de su propio nodo *Cloud*, es decir, la interfaz del servidor, para los intercambios del tráfico de control con la red de datos.

Los equipos virtuales de la red se encuentran alojados en el servidor 155.210.157.162, mientras que la red de control está en el 155.210.157.161. De esta forma, comparten tanto la red virtual como la física. Al no existir interconexiones directas entre instancias

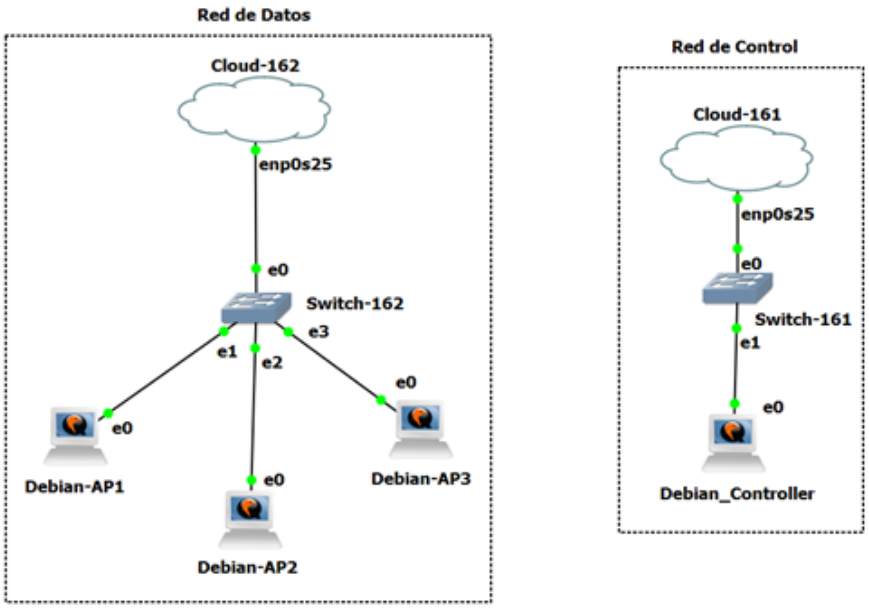

Figura 4.16: Escenario de red 2.1

virtuales de diferentes servidores GNS3, el tráfico intercambiado entre la red de datos y la de control viajará de forma natural en la red.

Dividir de esta manera los equipos físicos podría resultar ineficiente si se desea incrementar la cantidad de puntos de acceso. Asignar uno de los servidores como *host* de la red de datos limita la escalabilidad, puesto que las modificaciones realizadas a la imagen virtual de la máquina Debian provocan que requiera mayor espacio en disco y memoria RAM. Mientras que el servidor dedicado a la red de control se encuentra desaprovechado.

Por otro lado, la configuración de red de este escenario es sencilla. Cada equipo dispone únicamente de una interfaz de red virtual con una dirección asignada de la red IP 155.210.157.0/24. El flujo de paquetes de la red de datos sería visible tanto en la red virtual como en la real, además de recorrer la ruta una sola vez. Esto último supone una consideración importante, como se verá en futuros escenarios.

4.3.2. Escenario 2.2

Otra de las posibles opciones de integración del plano de control consiste en ubicarlo en la red de datos, por lo que compartiría el nodo nube con el plano de datos. De esta manera, la ubicación de los equipos virtuales en los servidores se vuelve más determinante. El escenario diseñado es el siguiente:

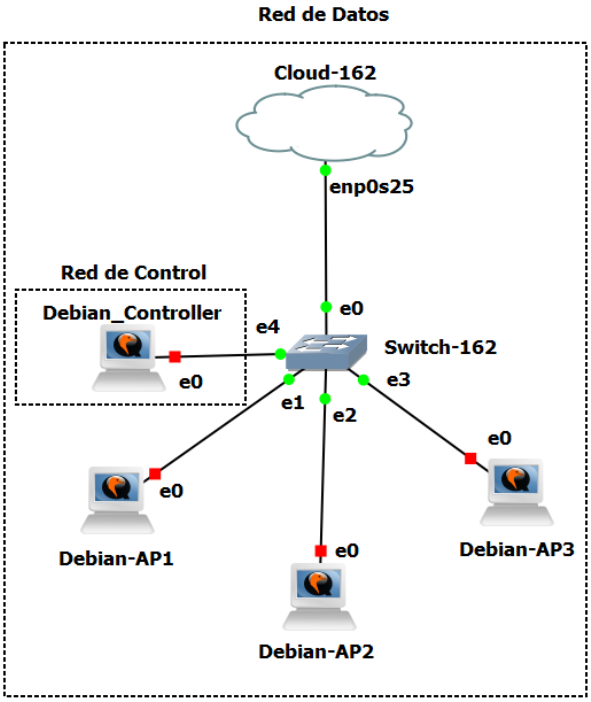

Figura 4.17: Escenario de red 2.2

Para comprobar las interacciones entre los tráficos generados por cada plano, se cambiarán las ubicaciones de los equipos entre los tres servidores GNS3 disponibles. La topología permanecerá inalterada, pero los flujos de tráfico se verán modificados.

En primer lugar, todas las máquinas del escenario se han alojado en el mismo servidor, cuya interfaz se ha vinculado al nodo *Cloud* como interfaz de salida. Manteniendo las direcciones de los equipos virtuales en la red 155.210.157.0/24, los intercambios se realizarán sin encapsulamientos, su encaminamiento será simple puesto que tanto el controlador como los puntos de acceso se encuentran en la misma subred y sus interfaces conectadas al mismo *switch*. El tráfico de datos tampoco se verá modificado, el flujo generado será enrutado a través de la red de datos y obtendrá acceso al exterior a través de la interfaz real del servidor.

Concentrar todos los puntos de acceso en un servidor es una opción viable en términos de Hardware, el equipo NUC dispone de hasta 4 puertos USB y dos interfaces para *Ethernet*, además, es posible aumentar el número de adaptadores Wi-Fi utilizando alargadores para cubrir las distancias deseadas. Sin embargo, este diseño presenta algunos inconvenientes. Utilizar un único servidor limita la capacidad de procesado, los recursos del equipo NUC son limitados y la cantidad de equipos que podrá emular son limitados. Por otro lado, todos los planos del sistema se encuentran compartiendo

la interfaz real del servidor como puerta de salida, esto puede resultar peligroso por muchos motivos: fallos en el enlace, no hay una interfaz secundaria que mantenga activa la red, o sobrecargas de tráfico. Debido a la naturaleza inalámbrica del sistema, estimar el tráfico que podría tener que soportar la red de datos resulta complicado., por este motivo utilizar un único servidor con una única interfaz podría conllevar problemas.

Como alternativa, puede dividirse la carga entre los tres servidores disponibles. Dicha modificación no requiere modificar las interconexiones de los equipos virtuales presentes en el escenario, pero si afectará al formato del tráfico intercambiado y al enrutamiento por lo que es necesario estudiar correctamente el impacto que tendrá sobre la red.

Por ejemplo, se reubican las máquinas virtuales de la siguiente manera: el servidor 155.210.157.161, servidor 161 en adelante, alojara el equipo controlador y el punto de acceso 1 y el equipo 155.210.157.162, servidor 162 en adelante, se mantiene como host de los puntos de acceso 2 y 3. Es recomendable mantener el nodo *Cloud* y el conmutador al que se conectan los demás equipos de red en un mismo servidor, en este caso ambos corren en el servidor 162.

Utilizando esta configuración el flujo de tráfico intercambiado se ve drásticamente alterado. En primer lugar, las tramas *Ethernet* de datos intercambiados son encapsuladas en datagramas UDP y enrutados por los enlaces reales entre los servidores involucrados. Por ejemplo, una petición *ICMP Echo Request* entre el controlador (Debian_Controller) y el punto de acceso 2 (Debian_AP2) provoca el siguiente tráfico UDP:

No.	Time	Source	Destination	Protocol	Length	Info
88	6.483718	155.210.157.162	155.210.157.161	UDP	140	10008 → 10004 Len=98
92	6.773929	155.210.157.162	155.210.157.161	UDP	102	10008 → 10004 Len=60
101	7.484071	155.210.157.161	155.210.157.162	UDP	140	10004 → 10008 Len=98
102	7.485086	155.210.157.162	155.210.157.161	UDP	140	10008 → 10004 Len=98
104	7.506918	155.210.157.162	155.210.157.161	UDP	102	10008 → 10004 Len=60
108	7.773754	155.210.157.162	155.210.157.161	UDP	102	10008 → 10004 Len=60
114	8.203634	155.210.157.162	155.210.157.161	UDP	102	10008 → 10004 Len=60

Figura 4.18: Tráfico UDP en el enlace

Sin embargo, una vez los datagramas ingresan en el servidor correspondientes son desencapsulados y procesados por el equipo virtual correspondiente, por ello esta modificación en la ruta y el encapsulamiento de las tramas de nivel 2 se realizan de forma a transparente a los equipos virtuales.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.671710	155.210.157.169	155.210.157.170	ICMP	98	Echo (ping) request id=0x0260, seq=1/256, ttl=63 (reply in 5)
5	0.672561	155.210.157.170	155.210.157.169	ICMP	98	Echo (ping) reply id=0x0260, seq=1/256, ttl=64 (request in 4)
11	3.727973	155.210.157.169	155.210.157.170	ICMP	98	Echo (ping) request id=0x0260, seq=4/1024, ttl=63 (reply in 12)
12	3.728487	155.210.157.170	155.210.157.169	ICMP	98	Echo (ping) reply id=0x0260, seq=4/1024, ttl=64 (request in 11)
14	4.728967	155.210.157.169	155.210.157.170	ICMP	98	Echo (ping) request id=0x0260, seq=5/1280, ttl=64 (reply in 15)
15	4.729819	155.210.157.170	155.210.157.169	ICMP	98	Echo (ping) reply id=0x0260, seq=5/1280, ttl=64 (request in 14)
21	5.730529	155.210.157.169	155.210.157.170	ICMP	98	Echo (ping) request id=0x0260, seq=6/1536, ttl=64 (reply in 22)
22	5.731350	155.210.157.170	155.210.157.169	ICMP	98	Echo (ping) reply id=0x0260, seq=6/1536, ttl=64 (request in 21)

Figura 4.19: Tráfico en el enlace virtual

Este *ping* realizado entre dos equipos que corren en diferente host provoca un mayor intercambio en la red real. Observando el diagrama del escenario, se deduce que el camino utilizado en la red virtual seria controlador, *switch* y punto de acceso 2, pero debido al funcionamiento de la virtualización se generará más cantidad de paquetes en la red física. El servidor 161, donde se encuentra el controlador, envía la trama de datos encapsulada al servidor 162, donde se encuentran tanto el *switch* como el punto de acceso 2, y este servidor reenviará los *Echo Reply* generados en consecuencia. Por tanto, este ping en la red real se traduce en un intercambio de paquetes UDP entre los servidores 161 y 162. Si se realiza un nuevo *ping*, esta vez dirigido al exterior desde el controlador, este encapsulamiento y redireccionamiento entre servidores provocara que cada paquete que sea enviado o recibido aparezca repetido en la red física.

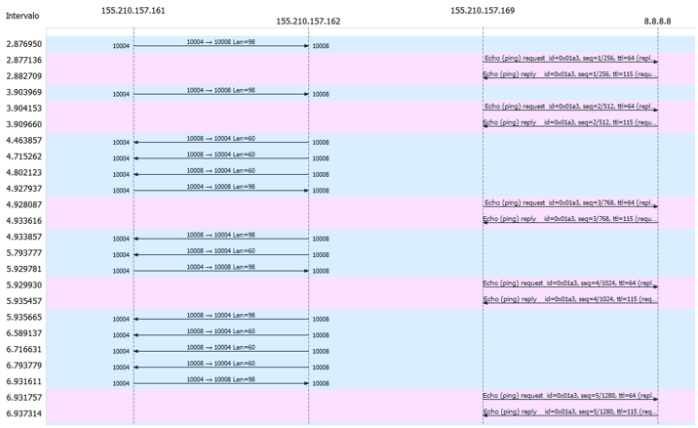

Figura 4.20: Tráfico en el interior(izquierda) frente a exterior(derecha)

Como muestra la gráfica de flujo del tráfico capturado, por cada *ICMP Echo Request* que el controlador genera en la red real se observará un datagrama UDP hacia el servidor 162 correspondiente al encapsulado del *ping* que se envía en el enlace controlador-switch, una vez en el servidor, se desencapsula y es reenviado hacia el exterior, recordemos que la interfaz del servidor 162 es usada por la red virtual como puerta de salida. Para los paquetes *ICMP Echo Reply* recibidos se realiza el mismo procedimiento, pero de manera inversa, aunque la cantidad de paquetes generados será

la misma.

Por tanto, si se aumentase el número de equipos pertenecientes a diferentes servidores en un mismo escenario, este tráfico repetido en la red real se incrementará de igual manera, dependiendo de los caminos definidos en la tabla de enrutamiento. Así pues, deberá alcanzarse un compromiso entre el número de host presentes en un escenario y sus interconexiones a la vez que se limita en la medida de los posible el tráfico redundante.

4.3.3. Escenario 3

En los anteriores escenarios, se ha estudiado la ubicación de los puntos de acceso y el controlador en los diferentes servidores habilitados, así como las ventajas y desventajas de su interconexión.

El canal de datos existente entre un controlador y los diferentes equipos puede configurarse utilizando una red común a los canales de datos o ser de uso exclusivo para comunicaciones de control. Por el contrario, en este diseño se propone asignar una nueva subred virtual al plano de control y separarlo así de la red de datos.

Para ello se modifica el template original de las maquinas Debian para añadirles más interfaces virtuales de red. Además, hasta ahora todos los escenarios presentados compartían una misma red IP, tanto los equipos reales como virtuales tenían asignadas direcciones de la red 155.210.157.0/24, por lo que el encaminamiento podía simplificarse en gran medida y el equipo encargado de esa tarea era el router de la red física (155.210.157.254). Sin embargo, al añadir una nueva red IP virtual, deben hacerse modificaciones para permitir el flujo de paquetes desde y hacia la nueva red.

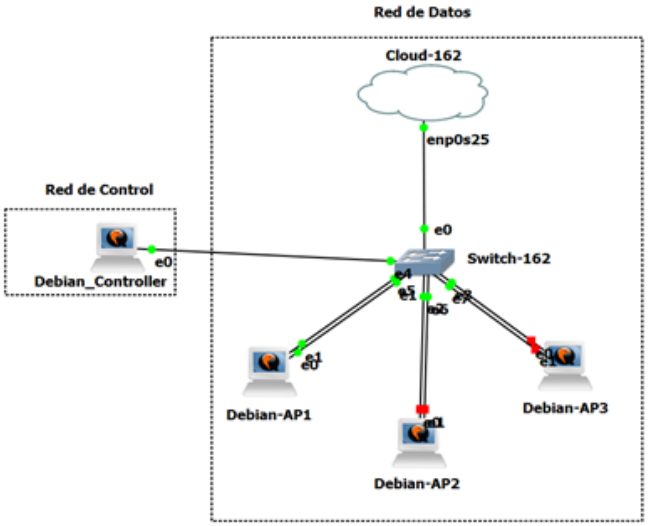

Figura 4.21: Escenario de red 3

El primer diseño planteado consiste en la conexión directa del controlador al conmutador de la red de datos. Los equipos de la red de datos poseen dos interfaces diferentes, una de ellas para la comuncación con la red de datos y la segunda disponible para el tráfico de control.

La tabla de enrutamiento de los equipos también es trivial en este punto. Sin embargo, el equipo controlador dispone únicamente de una interfaz de red, en la nueva red 10.0.0.0/24, por lo que su acceso al exterior está limitado por las configuraciones del router de la red física.

Los equipos de la red de datos obtendrán acceso al exterior a través de su interfaz en la red 155.210.157.0/24, mientras que podrán comunicarse con la red de control a través de la interfaz con dirección 10.0.0.0/24.

Figura 4.22: Intercambio en las redes de control (izquierda) y datos(derecha)

Siguiendo el mismo principio, puede asignarse el equipo de control a otro servidor en el que pueden estar alojados diversos equipos para distribuir la carga de la virtualización como se ha expuesto previamente. Este formato tiene un funcionamiento similar, las máquinas que formen parte de la red de datos contarán con dos interfaces direccionadas en redes IP diferentes para permitir la comunicación con Internet y el plano de control.

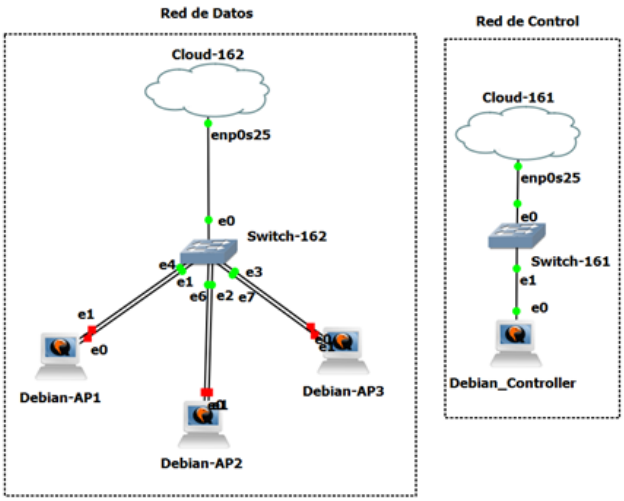

Figura 4.23: Escenario de red 3.2

Si se desease dotar al equipo controlador de conectividad a Internet, en ambos escenarios, debería modificarse la configuración del router físico de la red real para incluir las rutas correspondientes a la nueva red asignada al plano de control, así como las reglas de *firewall* correspondientes para garantizar la seguridad del equipo controlador. Es importante remarcar que, aunque la arquitectura de la red diseñada sea SDN, esta está vinculada a la red virtual, por lo que el controlador no tendrá influencia sobre los equipos enrutadores reales, sino en las rutas virtuales creadas entre los equipos simulados.

Alternativamente, una posibilidad que requiere de menos configuración consiste en asignar al equipo de control una segunda interfaz, como se ha hecho en los puntos de acceso, cuya dirección pertenezca a la red datos.

Por último, si se desea evitar la inclusión de nuevos conmutadores o interfaces de red, puede realizarse la interconexión mostrada en el escenario siguiente:

Este diseño se basa en la capacidad de *forwarding* de los puntos de acceso. El equipo controlador se encuentra enlazado con el punto de acceso en una interfaz perteneciente a la red de control. Sin embargo, utilizará la interfaz de la red de datos para obtener acceso al exterior.

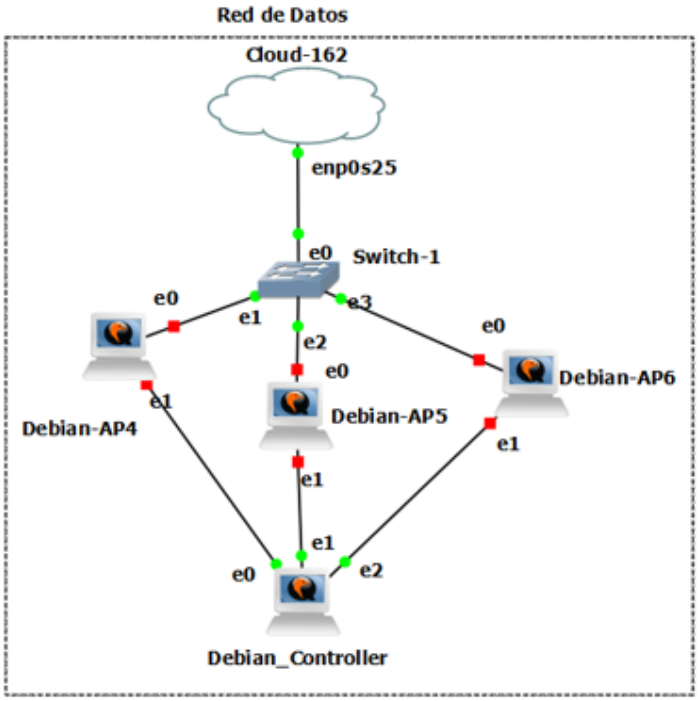

Figura 4.24: Escenario de red 3.3

Esta configuración también requiere modificar el enrutamiento en la red real, así como en la virtual. Además, debe habilitarse el *forwarding* de paquetes en las máquinas que se interconecten directamente con el equipo de control. De esta forma, puede mantenerse el controlador en una red IP alternativa mientras se limita la cantidad de nuevos enlaces que deberían crearse para dar soporte a la red de control, puesto que un único enlace entre punto de acceso y controlador bastaría.

Esta última opción es viable y funcional, pero puede llegar a presentar inconvenientes. Si se opta por usar un único enlace, la red de control contaría con un solo punto de conexión con el plano de datos, algo que es recomendable evitar, puesto que si dicho enlace fallase el plano de datos quedaría aislado de las instrucciones de control. Los equipos Debian admiten hasta un máximo de 10 interfaces virtuales, por lo que sólo se podrían interconectar nueve puntos de acceso con un controlador, si se desea aumentar el número de máquinas deberían introducirse conmutadores o nuevos equipos de control.

Diseñar un plano de control escalable para una SDN es un desafío considerable. En un inicio, se ha diseñado la red con una arquitectura centralizada, un único controlador es responsable de la gestión de todos los equipos que componen la red. Este montaje

en concreto lastra la escalabilidad de la red. Además, el uso de un solo controlador tiene un gran impacto en la eficiencia de la red, puesto que podría provocar latencia, retrasos y cuellos de botella en los enlaces cercanos al controlador. [6]

La solución más adecuada para los problemas relacionados con el escalado de la red es la implementación de una red control con varios equipos controladores. El flujo de datos se repartirá entre los diferentes dispositivos de control de acuerdo con la topología, distribuyendo de este modo la carga y evitando los riesgos de un punto único de fallo.

4.4. Resumen de resultados

Tras el montaje y análisis de los escenarios, en este apartado se recogen los resultados y conclusiones extraídas.

A raíz del Escenario 0, se ha comprendido mejor el funcionamiento de las comunicaciones entre servidores GNS3, basadas en túneles UDP, además de solucionar los problemas de realimentación de tráfico durante las capturas. Las pautas indicadas en este primer apartado deben ser respetadas en todos los futuros escenarios que se diseñen, para asegurar su correcto funcionamiento.

A continuación, se preparó un escenario sencillo con el objetivo de lograr la adquisición de una interfaz inalámbrica real por parte de una máquina virtual, el Escenario 1. El objetivo de este escenario era aprender a capturar diferentes interfaces Wi-Fi y modificar el equipo virtual en consecuencia para conseguir el montaje de un punto de acceso. Estas pruebas se realizaron utilizando la herramienta Hostapd, en lugar de los equipos Odin Wi5, por simplicidad, dado que una vez definido el proceso de obtención de la interfaz de red, este iba a ser idéntico independientemente de la máquina virtual que se utilizase.

Por último, los Escenarios 2 y 3, se diseñaron con el objetivo de estudiar las posibilidades de despliegue de las redes de control y datos. Las conclusiones y mejores prácticas para el despliegue son las siguientes:

Interfaces de salida

Debido al reenvío de tráfico a través de los túneles UDP para obtener acceso al exterior, puede aparecer demasiado tráfico circulando en los enlaces entre equipos. Para reducir, en la medida de lo posible, este reenvio de tráfico es recomendable utilizar todas las interfaces de salida (nodos *Cloud*) disponibles en los servidores, en lugar de

concentrar todo el tráfico con destino externo en una única interfaz. Además, disponer de varias salidas evita cuellos de botella o pérdidas de conexión en caso de fallo en uno de los enlaces.

Ubicación de las máquinas virtuales

En la medida de lo posible, debería balancearse la carga de virtualización entre los diferentes servidores. Los equipos alojados en diferentes servidores que se encuentren interconectados provocarán el reenvío de tráfico, esto no es un problema grave en los escenarios actuales, pero puede llegar a reducir la eficiencia. Por ello, se recomienda utilizar las interfaces de salida en conjunto con los enlaces virtuales. Por último, los *switches* que filtran el trafico de entrada a través del nodo *Cloud* deben estar alojados en el servidor al que pertenezca dicho nodo *Cloud*.

Redes IP

En los escenarios desarrollados se han planteado dos opciones: utilizar una red IP diferente para la red de datos y la de control o asignar a todos los equipos direcciones de una misma red. En términos de rendimiento, este no es un punto decisivo. Sin embargo, utilizar redes diferentes en el plano de control y en el de datos facilita la monitorización del tráfico, la toma de decisiones por parte del controlador y más importante aún, la seguridad de la red. El controlador es el centro lógico del despliegue y ubicarlo en una red diferente facilita la implementación de reglas de filtrado de tráfico para su protección.

Enlaces y equipos redundantes

A lo largo del estudio, se ha mencionado la necesidad de implementar equipos de *backup* en caso de fallos y para reducir la sobrecarga en enlaces y equipos. Las máquinas virtuales utilizadas son ligeras y pueden operar ocupando pocos recursos. Esta ventaja facilita la implementación de equipos y enlaces redundantes para facilitar la distribución del tráfico y la creación de túneles entre máquinas virtuales más eficientes.

Capítulo 5

Conclusiones y Líneas Futuras

5.1. Conclusiones

La finalidad de este trabajo ha consistido en estudiar los diferentes escenarios que pueden diseñarse en un despliegue de red, prestando especial atención a las posibles formas de gestionar y monitorizar la red.

En primer lugar, se ha seleccionado GNS3 como herramienta para la gestión y el análisis de las redes creadas. Se ha realizado un estudio de su funcionamiento y sus posibilidades, comprobando que es una aplicación adecuada para gestionar los escenarios. Una vez completado el estudio, se han desplegado los servidores remotos que dan soporte a los equipos virtuales, para comprobar su funcionamiento en diferentes ubicaciones.

Una vez se hubo seleccionado la herramienta de gestión, se configuraron las máquinas virtuales a partir de una imagen base Debian 10 Minimal. A esta máquina se han añadido las diferentes herramientas y *daemons* que permitían ejecutar las funcionalidades que se buscaban. El objetivo ha consistido en preparar las máquinas de forma ligera para permitir un despliegue rápido y flexible.

Tras configurar los equipos Debian virtuales se ha procedido a estudiar las posibles formas de crear un punto de acceso. Antes de todo, debía permitirse a la máquina virtual hacer uso de una interfaz de red física. Para ello, se ha tenido que actualizar las distintas versiones de drivers y modificar tanto el kernel de la máquina como las reglas de acceso a dispositivos hardware. Una vez aprendida y documentada la operación para una interfaz concreta, se ha comprobado que el proceso es válido para distribuciones de diferentes fabricantes. Finalmente, se ha comprobado el funcionamiento del punto de acceso mediante la herramienta Hostapd, que ha permitido crear un punto de acceso

Wi-Fi virtual haciendo uso de la tarjeta real Wi-Fi, cuyo control se ha cedido la máquina virtual.

Por último, una vez se habían preparado todos los componentes de la red, se han estudiado las diferentes posibilidades de despliegue. En primer lugar, el estudio se ha centrado en comprender la forma en que el tráfico enviado por las interfaces virtuales es encaminado entre los servidores reales. Posteriormente se ha utilizado lo aprendido en este primer estudio para decidir la distribución de los planos de red SDN en los equipos reales, comprobando el funcionamiento cuando, por ejemplo, se interconectan equipos virtuales alojados en diferentes servidores o se dedican servidores al completo a simular máquinas de un único plano SDN.

En base al trabajo y las pruebas realizadas, se ha concluido que el despliegue de una red de acceso Wi-Fi real con puntos de acceso ligeros virtuales es posible y funcional, con independencia de su ubicación. Además, GNS3 ha resultado ser una herramienta que ofrece al administrador la capacidad de gestionar la red de manera sencilla e intuitiva. Los servidores GNS3 son seguros y su forma de comunicación ayuda a diferenciar y monitorizar tráficos. A esto se suma la interfaz de usuario de GNS3, que ha permitido mantener una visión completa y actualizada de la red

5.2. Líneas Futuras

Las redes SDN son una tecnología relativamente nueva, constantemente en desarrollo, por lo que las mejoras que podrían añadirse son muy numerosas. A esto hay que sumarle la orientación de los nuevos dispositivos hacia las tecnologías inalámbricas. En este apartado podrían detallarse muchos añadidos diferentes para continuar desarrollando la red, aunque se centrará en las que se consideran más relevantes.

Sustitución de *switch*

A lo largo del trabajo se han incluido mayoritariamente equipos *switch* que implementaban openVSwitch para realizar las labores de *forwarding* de acuerdo con el protocolo OpenFlow. Teniendo en cuenta el objetivo de escalabilidad de la red, sería interesante sustituir estos equipos por *router* o *switch* de nivel 3 que permitiesen la interconexión de diferentes redes más complejas para simplificar el proceso de expansión en diferentes localizaciones.

Sistema de *backup*

Como se ha comentado a lo largo de este documento, los equipos virtuales son fácilmente reemplazables en caso de fallo. Sin embargo, si uno de los equipos clave falla puede hacer peligrar el funcionamiento de la red, por ejemplo, si el controlador sufre una caída. Por ello sería necesario incluir una serie de equipos de apoyo para reemplazarlos mientras dure el fallo.

Seguridad

Los puntos de acceso Wi-Fi utilizados en los escenarios de prueba pueden utilizar seguridad WPA2 o inferior. Este estándar es robusto ante ataques de fuerza bruta pero vulnerable a spoofing, por lo que actualizar las claves de acceso a WPA3 podría reducir la amenaza. De la misma forma, el formato de red que se ha planteado es vulnerable también a ataques DoS. Saturar con tráfico uno de los puntos de acceso podría no ser un problema tan grave. Sin embargo, esto afectaría al rendimiento del controlador que sería incapaz de diferenciar entre tráfico útil y tráfico dañino. Para solucionar esto se propone dotar a la SDN de herramientas *Intrusion Prevention Sysem* (IPS) e *Intrusion Detection System* (IDS) [11] . Además, utilizar el estándar sFlow en conjunto con el protocolo OpenFlow haría el plano de control más robusto en cuanto a la detección y rechazo del tráfico malicioso [4].

Actualización a Wi-Fi 6

Los equipos virtuales basados en Odin implementa el estándar de Wi-Fi 5 (Estándar 802.11ac), aunque en los últimos tiempos el desarrollo de su sucesor, Wi-Fi 6 (Estándar 802.11ax), ha aumentado enormemente. Esta sería la mejora más compleja de todas las planteadas. Las ventajas de Wi-Fi 6 frente a su predecesor son varias:

Permite operar tanto en las bandas de 2.4 y 5 GHz, mientras que Wi5 opera únicamente en la primera. Wi-Fi 6 utiliza OFDMA para la división de los canales, lo que reduce la latencia e incrementa la eficiencia en redes con gran cantidad de dispositivos Implementa MU-MIMO que permite la transmisión de datos simultánea a varios dispositivos, mientras que la versión anterior debía hacerlo uno a uno.

En resumen, Wi-Fi 6 proporciona mayor velocidad y ancho de banda y puede dar servicio a un mayor número de dispositivos a la vez [10].

Machine Learning

Puede dotarse al controlador de mayor "inteligencia" a través de algoritmos más complejos y modernos, así como técnicas de Machine Learning. A partir de estas modificaciones, se pueden plantear objetivos de planificación y ajuste dinámico adecuados para entornos mucho más poblados o la implementación de mecanismos de QoS [21, 16]

Bibliografía

[1] dnsmasq(8) - linux man page. URL: https://linux.die.net/man/8/dnsmasq.

[2] Gns3 2.1.22 developer documentation. URL: http://api.gns3.net/en/2.1/.

[3] Hostapd. URL: https://w1.fi/hostapd/.

[4] sflow protocol. URL: https://sflow.org/.

[5] Wireshark, Nov 2018. URL: https://www.wireshark.org/.

[6] A.Abuarqoub. A review of the control plane scalability approaches in software defined networking. 2020.

[7] R. Barbosa J. Bernardino B. Rodrigues, F. Cerveira. Virtualization: Past and present challenges.

[8] Fabrice Bellard. Qemu, a fast and portable dynamic translator. In *USENIX annual technical conference, FREENIX Track*, volume 41, page 46. Califor-nia, USA, 2005.

[9] Cisco. Cisco annual internet report (2018–2023) white paper. March 2020.

[10] ZTE Company. Wi-fi 6 technology and evolution white paper.

[11] M. Amanowicz D. Jankowski. Intrusion detection in software defined networks with self-organized maps., 2015. URL: https://core.ac.uk/download/pdf/235207356.pdf.

[12] A. Raschellà Q. Shi F. den Hartog J. Saldana R. Munilla J.Ruiz-Mas J. Fernández-Navajas J. Almodovar N. van Adrichem F. Bouhafs, M. Mackay. Wi-5: A programming architecture for unlicensed frequency bands.

[13] Open Networking Foundation. *SDN Architecture Overview.* Open Networking Foundation, December 2013. URL: https://opennetworking.org/wp-content/uploads/2013/02/TR_SDN_ARCH_1.0_06062014.pdf.

[14] J. L. Salazar Riaño J.P. Javaudin J.M. Bonnamy M. Le Dizes J. Saldana, J. Fernández-Navajas. Attention to wi-fi diversity: Resource management in wlans with heterogeneous aps. Jan 2021.

[15] P. Isolani C. Both J. Rochol L. Granville J. Wickboldt, W. De Jesus. Software-defined networking: management requirements and challenges. 2015.

[16] T. Huang R. Xie J. Liu C. Wang Y. Liu Junfeng Xie, F. R. Yu. A survey of machine learning techniques applied to software defined networking (sdn): Research issues and challenges.

[17] Kristián Košťál, Rastislav Bencel, Michal Ries, Peter Trúchly, and Ivan Kotuliak. High performance sdn wlan architecture. *Sensors*, 19(8), 2019. URL: https://www.mdpi.com/1424-8220/19/8/1880, https://doi.org/10.3390/s19081880 doi:10.3390/s19081880.

[18] C. Jacquenet M. Boudacair. Software-defined networking: A perspective from within a service provider environment, March 2014. URL: https://datatracker.ietf.org/doc/html/rfc7149#section-1.

[19] Roberto Morabito, Jimmy Kjällman, and Miika Komu. Hypervisors vs. lightweight virtualization: A performance comparison. March 2015. https://doi.org/10.1109/IC2E.2015.74 doi:10.1109/IC2E.2015.74.

[20] E. Zegura N. Feamster, J. Rexford. The road to sdn: An intellectual history of programmable networks. URL: https://www.cs.princeton.edu/courses/archive/fall13/cos597E/papers/sdnhistory.pdf.

[21] P. Pinto L. Bernardo J. Tavares H. s. Mamede P. Amaral, J. Dinis. Machine learning in software defined networks:data collection and traffic classification. Nov 2016.

[22] Ivan Pogarcic, David Krnjak, and Davor Ozanic. Business benefits from the virtualization of an ict infrastructure. *International Journal of Engineering Business Management*, 4:1, 11 2012. https://doi.org/10.5772/51603 doi:10.5772/51603.

[23] J. Mogul H. Frystyk L. Masinter P. Leach R. Fielding, J. Gettys and T. BernersLee. Hypertext transfer protocol – http/1.1. technical report rfc 2616. 1999. URL: http://www.ietf.org/rfc/rfc2616.txt.

[24] T.S. Rappaport S. Shakkottai. Research challenges in wireless networks: a technical overview. 2010.

[25] T.Bray . The javascript object notation (json) data interchange format. rfc 8259, Dec 2021. URL: https://datatracker.ietf.org/doc/html/rfc8259.

[26] ONF Tr. Sdn architecture. 2016. URL: https://opennetworking.org/wp-content/uploads/2013/02/TR_SDN_ARCH_1.0_06062014.pdf.

[27] VMWare. Making the business case for virtualization.

Anexos

Anexo I

Instancias virtuales e imágenes

En este anexo se detalla la gestión de las imágenes virtuales, su interoperabilidad con GNS3, además de métodos de almacenamiento y gestión.

I.1. Imágenes host GNS3

Por defecto, GNS3 almacena las imágenes de las distintas maquinas que se introducen en los proyectos como snapshots. Esto quiere decir que cuando se arranque una máquina que haya sido modificada con anterioridad, el hipervisor (KVM en este caso) lanzará primero la imagen base de la máquina, definida en el template, y después cargará el snapshot correspondiente a la imagen modificada.

Este proceso es muy eficiente en cuanto a tiempos de carga y gestión de recursos y memoria, puesto que ahorra tener que salvar la misma imagen varias veces.

Las instancias virtuales se almacenan en la siguiente ruta del servidor GNS3:

```
1 /home/tfggns3/GNS3/images/QEMU
```

Mientras que los snapshots de las máquinas, se almacenan en los directorios que se crean automáticamente para los proyectos:

```
1 /home/tfggns3/GNS3/projects/.../project-files/qemu/
```

I.2. Actualización imágenes base

Aunque no es el modo común de operación, el cliente de GNS3 permite modificar una máquina virtual QEMU directamente en su imagen base, evitando crear snapshots de esta.

Este modo de configuración resulta útil cuando queremos replicar una configuración aplicada a la máquina a todas las que se vayan a utilizar en el futuro. Para ello, en el menú de configuración avanzada del template, debe desactivarse la opción "Use as a linked base VM"

Figura I.1: Opciones avanzadas de un template

Mientras dicha opción esté desactivada, todos los cambios realizados sobre la máquina se almacenarán en la imagen base. Sin embargo, esto presenta limitaciones: únicamente se permite disponer de un equipo basado en una imagen sin enlazar.

Es recomendable desactivar esta opción una vez se hayan introducido las modificaciones deseadas.

Anexo II

Preparación de la máquina Debian

II.1. Consola Telnet

Para permitir el acceso a la línea de comandos mediante una consola serie, se debe modificar el fichero en la ruta:

```
/etc/default/grub
```

Y se añaden las siguientes líneas:

```
/GRUB_CMDLINE_LINUX="console=tty0 console=ttyS0,115200n8"
GRUB_TERMINAL=serial
GRUB_SERIAL_COMMAND="serial --speed=115200 --unit=0
\ --word=8 --parity=no --stop=1"
```

Una vez introducido el contenido, debe actualizarse el archivo de configuración de grub:

```
update-grub
```

Finalmente, se debe reiniciar la máquina para acceder al guest mediante la consola Telnet.

II.2. Redimensionado

Inicialmente, las maquinas creadas en el servidor tienen asignadas un espacio en disco reducido que no bastará para alojar los paquetes y añadidos de la máquina.

Para incrementar el tamaño en disco de una imagen el cliente GNS3 dispone de la instrucción resize, en el apartado HDD de cada template. Sin embargo, esto no es suficiente para asignar el espacio requerido. Desde la máquina se debe validar la configuración de dicha partición.

En primer lugar, se debe diferenciar la partición asignada con el espacio inicial, puesto que la modificación realizada desde GNS3 aun no es efectiva. Para ello se hace uso de cualquiera de los siguientes comandos:

```
df -l
lsblk
fdisk -l
```

Una vez localizada la partición correspondiente, a modo ilustrativo se amplía el disco sda1 en este ejemplo, se asigna el espacio indicado en disco por GNS3:

```
growpart /dev/sda 1
resize2fs /dev/sda1
```

II.3. Actualización kernel y controladores.

En este proyecto, se ha usado el adaptador de red RTL8812AU de Realtek. Los controladores de algunos dispositivos presentan errores e incompatibilidades con el Kernel instalado en la maquina Debian. Para solucionar dichos fallos, debemos actualizar el Kernel a una versión compatible.

Como primer paso, es recomendable actualizar la lista de fuentes de apt, puesto que las actualizaciones y cambios en los paquetes de Debian son comunes y puede resultar útil tener acceso a paquetes anteriores. El fichero que se debe modificar es:

```
/etc/apt/sources.list
```

Añadiendo las siguientes líneas:

```
deb http://deb.debian.org/debian stretch main
deb-src http://deb.debian.org/debian stretch main
deb http://deb.debian.org/debian-security/
\ stretch/updates main
deb-src http://deb.debian.org/debian-security/
\ stretch/updates main
deb http://deb.debian.org/debian stretch-updates main
deb-src http://deb.debian.org/debian stretch-updates main
```

Tras modificar el fichero de fuentes, debe actualizarse *apt*:

```
sudo apt update
sudo apt upgrade
```

También se requiere la instalación de *dkms*, el módulo encargado de la gestión de actualizaciones del Kernel:

```
apt-get install dkms
```

Una vez actualizados, se procede a instalar la nueva versión del Kernel:

```
sudo apt-get install linux-headers-4.19.0-14-common
```

Por último, se procede a clonar el repositorio de los controladores de la tarjeta de red:

```
git clone -b v5.6.4.2
\ https://github.com/aircrack-ng/rtl8812au.git
```

Desde el directorio creado, se montan e instalan los controladores en el sistema:

```
make && make install
```

II.4. Configuración de red

La configuración de red de la máquina Debian va a modificarse conforme se avanza en la configuración.

En un principio, únicamente se requiere salida al exterior para descargar los paquetes y repositorios necesarios, con una configuración sencilla será suficiente:

```
auto ens3
iface ens3 inet static
        address 155.210.157.170
        netmask 255.255.255.0
        gateway 155.210.157.254
        dns-nameservers 8.8.8.8 1.1.1.1
```

Sin embargo, una vez configurado el punto de acceso, se modificará esta configuración para crear un puente entre la interfaz de la maquina Debian y la interfaz inalámbrica sobre la que se monta el AP. Para ello, se utilizan 2 ficheros diferentes. El primero, será el fichero de arranque del demonio Hostapd. En este fichero se debe indicar la ruta absoluta del fichero de configuración de Hostapd para arrancar el servicio desde el lanzamiento de la máquina. El segundo fichero, se creará para configurar el puente entre interfaces con el demonio de init desde el proceso de *boot*. El fichero se guardará en la siguiente ruta:

```
/etc/network/interfaces.d/br0
```

Y el contenido es el siguiente:

```
auto br0
iface br0 inet static
        address 155.210.157.170
        broadcast 155.210.157.255
        netmask 255.255.255.0
        gateway 155.210.157.254
        dns-nameservers 8.8.8.8 1.1.1.1
        bridge_ports ens3
        bridge_stp off       # Sin Spanning Tree Protocol
                bridge_waitport 0    # No delay
                bridge_fd 0
```

Es importante remarcar que la opcion *bridge_ports* incluye únicamente la interfaz cableada y no la inalámbrica. En el proceso de lanzamiento, hostapd añadirá automáticamente la interfaz WiFi al puente, que devolvería un error si ya estuviese ocupado. Debido a esto, es necesario que durante el inicio de la máquna, Hostapd se lanze después de levantarse las interfaces de red.

Anexo III

Creación de un punto de acceso con Hostapd

III.1. Captura puertos USB

Como paso previo a la cesión del control de los puertos USB, debe modificarse la configuración del servidor, ya que, por defecto, las reglas definidas no permiten operaciones como estas. Estas reglas predeterminadas pueden consultarse en la ruta *"/usr/lib/udev/rules.d"*

Para sobrescribir dichas reglas, basta con crear un nuevo fichero en una ruta más prioritaria:

```
/etc/udev/rules.d/usb.rules
```

Y añadir la regla:

```
SUBSYSTEM="usb", MODE="0666"
```

En el entorno actual, esta regla no presenta elevados riesgos. Sin embargo, en un sistema diferente al actual sería considerada demasiado permisiva e insegura, por lo que una regla más aceptable sería ceder el control únicamente cuando se conecte un determinado dispositivo y a un determinado grupo de usuarios:

```
ATTRS{idVendor}=="0bda", ATTRS{idProduct}=="8812",
\ MODE = "0660", GROUP="tfggns3"
```

Una vez configurada la regla, se debe modificar el fichero de lanzamiento de QEMU/KVM de la maquina Debian. Para ello, se añaden las siguientes instrucciones a la configuración avanzada de maquina desde el cliente de GNS3:

```
-device ich9-usb-ehci1,id=usb,bus=pci.0
\ -device usb-host,vendorid=0xAAAA,productid=0xBBBB,
```

```
\ id=hostdev0,bus=usb.0
```

Los identificadores vendorid y productid pueden obtenerse ejecutando *lsusb* en el Host GNS3, apareciendo en formato hexadecimal AAAA:BBBB.

Dicha regla puede modificarse para transferir el control de un puerto en concreto, independientemente del dispositivo que se encuentre conectado, o de uno o más buses USB/PCI.

III.2. Configuración Hostapd

Para la creación y configuración de un punto de acceso, se ha utilizado la herramienta Hostapd. Este software habilita un demonio que, a través de una interfaz de red física, permite configurar un punto de acceso inalámbrico.

En primer lugar, se debe descargar del repositorio:

```
sudo apt-get install hostapd
```

Para permitir el arranque del servicio, bloqueado en un principio por el sistema, deben ejecutarse las siguientes instrucciones:

```
sudo systemctl unmask hostapd
sudo systemctl enable hostapd
```

Seguidamente, deben instalarse en el sistema los controladores necesarios para el dispositivo sobre el que va a montarse el punto de acceso. Dependiendo del adaptador de red utilizado, los controladores pueden descargarse del repositorio de recursos no libres:

```
sudo apt-get install firmware-misc-nonfree
```

Los controladores catalogados como no libres cambian a menudo y puede darse el caso de que determinados controladores pasen a ser libres. Este ejemplo se ha realizado con un adaptador Realtek RTL8812AU cuyos controladores no se encuentran en los repositorios. Además, para poder instalar estos controladores, debe haberse actualizado el Kernel del sistema.

En primer lugar, se debe clonar el repositorio de los controladores de la tarjeta de red:

```
git clone -b v5.6.4.2 \
https://github.com/aircrack-ng/rtl8812au.git
```

Desde el directorio creado, se montan e instalan los controladores en el sistema:

```
make && make install
```

Una vez instalados los componentes necesarios, se configuran los ficheros de Hostapd en la siguiente ruta:

```
/etc/hostapd/hostapd.conf
```

El contenido que debe introducirse en el fichero es:

```
country_code=ES
interface=wlx00c0caa4737b
bridge=br0
ssid=DEBIAN_AP_TEST
hw_mode=g
channel=7
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=1
wpa=2
wpa_passphrase=password
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Para habilitar el lanzamiento del demonio de en el arranque de la máquina, debemos indicarse la ruta del fichero de configuración en:

```
 /etc/default/hostapd
```

Y añadir la siguiente línea:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

III.3. Útiles para depuración

Las siguientes instrucciones muestran eventos detectados por el kernel del sistema, como la detección de un dispositivo conectado en un puerto USB. Estos comandos son útiles para comprobaciones de errores en las configuraciones anteriores:

```
Udevadm monitor --kernel
Udevadm monitor --udev
```

Muestra las reglas ejecutadas y acciones tomadas cuando el kernel detecta un evento. En nuestro caso, la conexión de un dispositivo en un puerto USB dispararía un evento:

Comprobar el funcionamiento correcto del fichero de configuración de Hostapd:

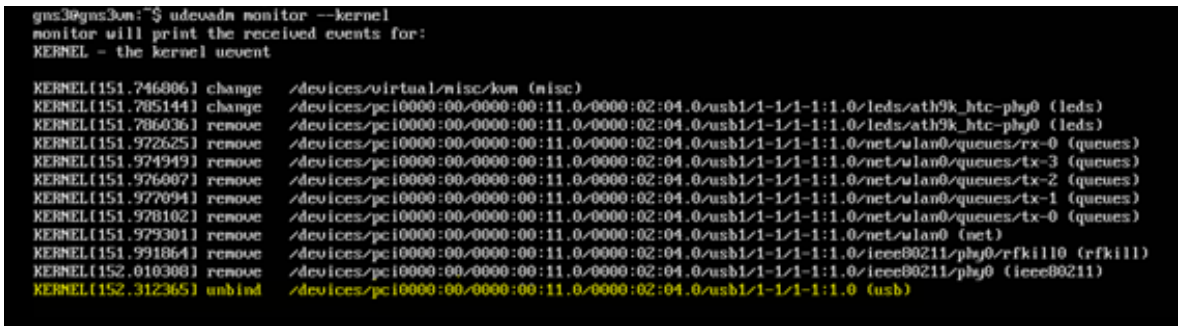

Figura III.1: Eventos del kernel

```
1 hostapd -d /etc/hostapd/hostapd.conf
```

La salida de la instrucción anterior devuelve información acerca de la ejecución del demonio.

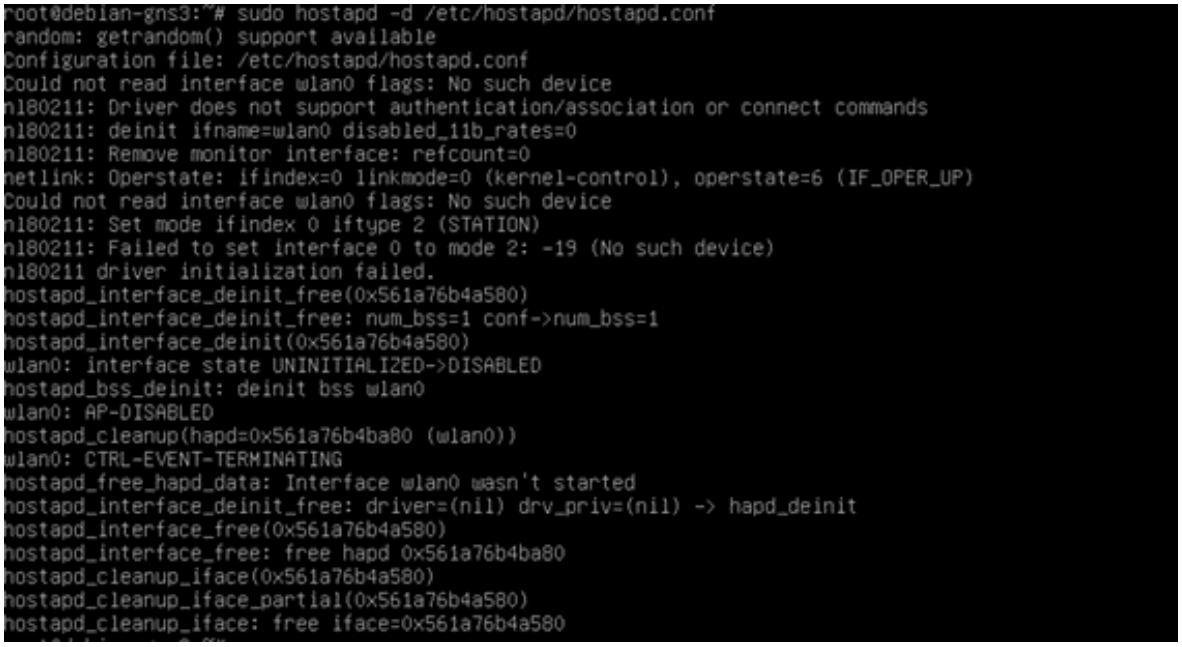

Figura III.2: Debug de la ejecución de Hostapd

Anexo IV

Configuración Host GNS3

GNS3 requiere que se disponga de un servidor que se utilizará para correr las distintas imágenes de los escenarios desarrollados. En este proyecto, se ha elegido correr este servidor sobre una distribución Ubuntu Server 20.04.1.

IV.1. Configuración de red Ubuntu Server

La configuración de red en la versión 20.04 de Ubuntu se realiza a partir de **Netplan**. Basta con crear un nuevo fichero en la ruta **/etc/netplan** con extensión **.yaml** para configurar el direccionamiento estático:

```
network:
  version: 2
  ethernets:
    enp0s25:
      addresses:
        - 155.210.157.162/24
      gateway4: 155.210.157.254
      nameservers:
          addresses: [8.8.8.8, 1.1.1.1]
```

Las versiones de Ubuntu con interfaz gráfica utilizan por defecto **Network-Manager** para controlar y modificar las interfaces de red del sistema. **Network-Manager** presenta algunas configuraciones por defecto que podrían interferir con la configuración de red deseada para el sistema, por ello, es recomendable desactivar o desinstalar **Network-Manager** y aplicar las configuraciones de red a través del Daemon **networkd**.

IV.2. Descarga e instalación del servidor GNS3

Como primer paso, se añade el repositorio de GNS3 a lista de fuentes y se procede a descargar el servidor de GNS3. En este caso, basta con descargar el servidor GNS3 en lugar de instalar también GNS3 GUI, puesto que este servidor será utilizado de forma remota por un cliente GNS3.

```
sudo add-apt-repository ppa:gns3/ppa
sudo apt update
sudo apt install gns3-server
```

El sistema Ubuntu, no dispone de ninguna versión de Docker instalada por defecto, se debe descargar la versión más actualizada del repositorio oficial.

Algunos repositorios de Linux/GNU usan GPG (Gnu Privacy Guard) para encriptar y firmar el código de algunos de sus repositorios. Por tanto, se debe descargar la clave GPG que permite descargar y desencriptar el package de Docker.

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg
\ | sudo apt-key add
```

Una vez se ha habilitado la clave **GPG**, se añade el repositorio oficial de Docker a la lista de fuentes de **apt**:

```
sudo add-apt-repository "deb [arch=amd64] \
https://download.docker.com/linux/ubuntu
\ $(lsb_release -cs) stable"
```

Se actualiza la base de datos de paquetes a partir del nuevo repositorio y se procede a la instalación:

```
sudo apt update
sudo apt install docker-ce
```

Como último paso, debe incluirse al usuario por defecto, o uno de nueva creación, a los grupos que tendrán permitido lanzar imágenes virtuales y contenedores de Docker. Si se desea crear un nuevo usuario:

```
sudo adduser tfggns3
```

Se incluye el usuario en los grupos necesarios:

```
sudo usermod -aG ubridge tfggns3
sudo usermod -aG libvirt tfggns3
sudo usermod -aG kvm tfggns3
sudo usermod -aG docker tfggns3
```

Una vez se ha configurado el servidor según las instrucciones anteriores, está preparado para correr el Host GNS3.

IV.3. Archivos de configuración servidor GNS3

La instalación por defecto del servidor no tiene configurado un demonio que permita su arranque, por lo que debe ser configurado manualmente mediante la creación de un fichero de inicio en la siguiente ruta:

```
/lib/systemd/system/gns3.service
```

Cuyo objetivo principal será definir el usuario que lanzará el servicio, la ruta absoluta en la que se localiza el fichero y las condiciones previas del demonio **systemd** para lanzar el servicio:

```
[Unit]
Description=GNS3 server
Wants=network-online.target
After=network.target network-online.target
[Service]
Type=forking
User=tfggns3
Group=tfggns3
PermissionsStartOnly=true
ExecStartPre=/bin/mkdir -p /var/log/gns3 /run/gns3
ExecStartPre=/bin/chown -R tfggns3:tfggns3 /var/log/gns3
\ /run/gns3
ExecStart=/usr/bin/gns3server --log /var/log/gns3/gns3.log \
--pid /run/gns3/gns3.pid --daemon
ExecReload=/bin/kill -s HUP $MAINPID
Restart=on-abort
PIDFile=/run/gns3/gns3.pid
[Install]
WantedBy=multi-user.target
```

La versión del servidor GNS3 instalado es la 2.2.19, aunque el repositorio de gns3 cuenta con versiones anteriores y posteriores que pueden ser instaladas clonando dicho repositorio. La configuración de otras versiones sigue el mismo proceso que el descrito anteriormente.

Una vez configurado el demonio de GNS3, se debe crear el fichero de configuración, extensión ".conf" , para la versión del servidor. Dependiendo de la versión del servidor utilizada, la ubicación de este fichero es diferente. Para la versión 2.2.19, la ruta es la

siguiente:

```
/home/tfggns3/.config/GNS3/2.2/gns3_server.conf
```

El contenido del fichero es el siguiente:

```
[Server]
; IP where the server listen for connection
host = 155.210.157.162
; HTTP port for controlling the servers
port = 3082
; Path where images of devices are stored
images_path = /home/tfggns3/GNS3/images
; Path where user project are stored
projects_path = /home/tfggns3/GNS3/projects
; Send crash to the GNS3 team
report_errors = True
;First port of the range allocated to devices
;telnet console
console_start_port_range = 2001
; Last port of the range allocated to devices telnet console
console_end_port_range = 5000
; First port of the range allocated to communication
;between devices. You need two port by link
udp_start_port_range = 10000
; Last port of the range allocated to communication
;between devices. You need two port by link
udp_end_port_range = 20000
; Path of the ubridge program
ubridge_path = /usr/bin/ubridge
; Boolean for enabling HTTP auth
auth = True
; Username for HTTP auth
user = proyecto
; Password for HTTP auth
password = proyecto

[Qemu]
;!! Remember to add the gns3 user to the KVM group,
;otherwise you will not have r/w permssions to /dev/kvm !!
enable_kvm = True
; Require KVM to be installed in order to start VMs
;(Linux has priority over require_hardware_acceleration)
require_kvm = True
; Enable hardware acceleration (all platforms)
enable_hardware_acceleration = True
; Require hardware acceleration in order to start VMs
```

```
;(all platforms)
require_hardware_acceleration = False
```

La configuración introducida en el archivo determina la dirección y el puerto en las que el servidor escuchará posibles conexiones entrantes, las rutas de los directorios de almacenamiento de imágenes y proyectos, además de habilitar autenticación para las conexiones. También habilita la virtualización de máquinas QEMU usando el hipervisor KVM.