Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

Maquetas para el aprendizaje de redes de ordenadores

Grado en Ingeniería Mecánica

ALUMNO: Iael Pilar Salafranca Francés

DIRECTORES: Norberto Fernández García

Miguel Rodelgo Lacruz

CURSO ACADÉMICO: 2021-2022

Universidade Vigo

Centro Universitario de la Defensa
en la Escuela Naval Militar

TRABAJO FIN DE GRADO

Maquetas para el aprendizaje de redes de ordenadores

Grado en Ingeniería Mecánica

Intensificación en Tecnología Naval

Cuerpo General

Universidade Vigo

RESUMEN

La enorme importancia de las redes de ordenadores en el panorama militar actual conlleva la necesidad de formar al personal de la Armada en este ámbito. Sin embargo, los estudiantes de la asignatura de ´Redes de Ordenadores´ que se imparte en quinto curso del Grado en Ingeniería Mecánica en la ENM no disponen de un escenario de pruebas de laboratorio desde el punto de vista experimental de construcción física de redes. Este es el porqué de construir diversas maquetas de redes con la finalidad de obtener un instrumento con el que fomentar el aprendizaje activo y complementar el aprendizaje teórico de una asignatura de redes.

En este TFG se proponen cinco montajes (maquetas) con equipos reales: rúteres MikroTik, conmutadores y equipos finales Raspberry Pi 3B con el fin de comprobar físicamente el funcionamiento de protocolos como ARP, HTTP, DHCP, 802.1q y OSPF. Se utilizan también herramientas auxiliares en el ámbito de la gestión y análisis de redes como el GNS3 y Cisco Packet Tracer a la hora de planificar el diseño, además de Wireshark para capturar e inspeccionar el intercambio de paquetes de los protocolos.

PALABRAS CLAVE

Maquetas, Protocolos, Analizadores de paquetes, Simuladores de redes, Aprendizaje activo

AGRADECIMIENTOS

En primer lugar, agradecer a mis padres el apoyo continuo que recibí de su parte durante los cinco años en la ENM. Gracias.

En segundo lugar, quería agradecer a mis dos tutores D. Norberto Fernández García y D. Miguel Rodelgo Lacruz toda la ayuda prestada en el desarrollo del proyecto.

CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Introducción y motivación

En 1969 el *Department of Defense* (DoD) de los Estados Unidos a través de la organización militar *Advanced Research Projects Agency* (ARPA) consiguió conectar por primera vez diferentes sitios informatizados en una red llamada ARPANET. En octubre de 1969 la red constaba de dos nodos y a finales de 1971 tenía 15 nodos.

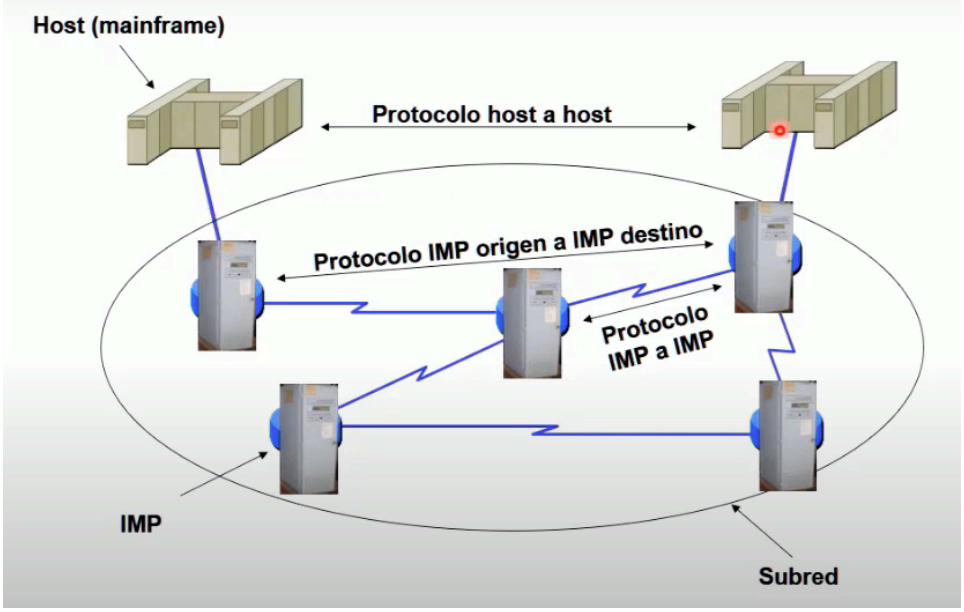


Figura 1-1 Diseño de la ARPANET original [2]

ARPA [2] investigó en protocolos de redes, en 1976 se sustituye el protocolo NCP (*Network Control Program*) original de ARPANET y es reemplazado por TCP/IP (*Transmission Control Protocol/ Internet Protocol*). En la actualidad estos protocolos son utilizados por cualquier dispositivo que quiera conectarse a Internet. En los años 80 con la aparición de los primeros ordenadores personales (PC) y la extensión de las redes de área local cobran especial importancia los sistemas distribuidos.

Uno de los principales usos de la Internet actual es el intercambio de información a través de la *World Wide Web* (o Web), los navegadores nos proporcionan un interfaz de utilización de aquella en la capa de aplicación y operando en la parte superior de los protocolos de Internet. La Figura 1-2 refleja la evolución del porcentaje de personas que la utilizan en estas últimas décadas.

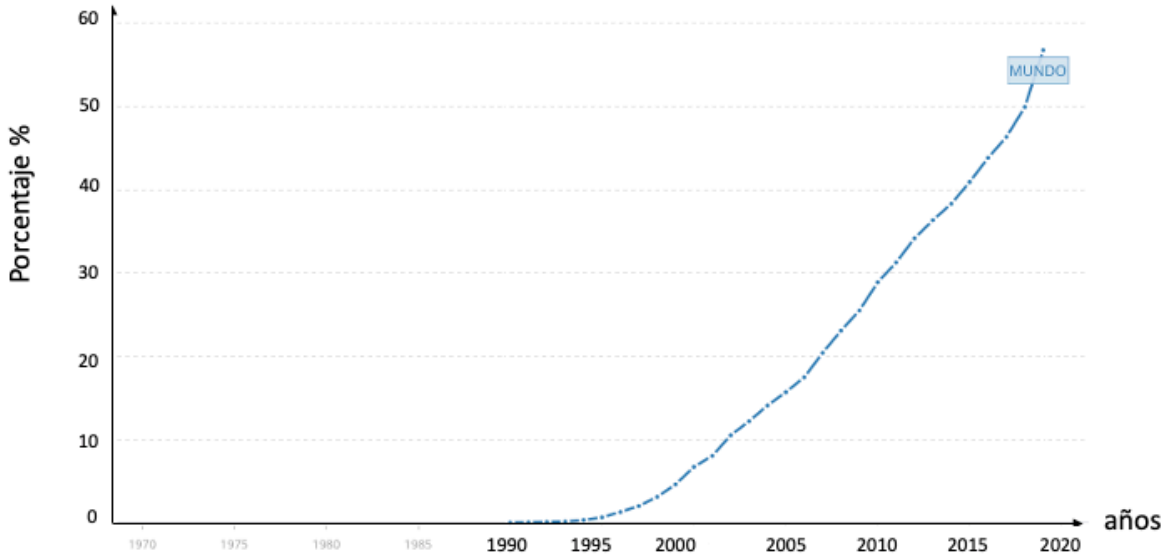


Figura 1-2 Porcentaje de personas que usan la Web [26]

En el ámbito militar las redes son imprescindibles hasta el nivel de que el Ciberespacio se ha convertido en un nuevo dominio de operaciones militares. El Ministerio de Defensa dispone de una red de mando y control para conectar nodos y unificar las comunicaciones, debe ser de alta seguridad, con requisitos más completos, ya que existen sistemas clasificados hasta el nivel de secreto de Estado. Tiene también la red WAN-PG (de propósito general) con topología de estrella siendo el centro el Ministerio de Defensa, con el propósito de que toda la información sea interdepartamental.

1.2 Arquitectura de redes: capas y protocolos

En 1984 la ISO **(***International Organization for Standardization*) estandarizó lo que debía ser una comunicación de red completa entre dos equipos sin importar quién fuera el fabricante. OSI (*Open Systems Interconnection)* es su modelo de interconexión de sistemas abiertos. Es una comunicación basada en capas que describen como construir e interpretar paquetes. Cada capa dispone de cometidos específicos comunicándose con las inmediatamente adyacentes y en ambas direcciones mediante el uso de protocolos. OSI es un modelo teórico que aún se estudia, aunque la pila real de Internet es TCP/IP. Ambos modelos se pueden ver en la Figura 1-3.

Figura 1-3 Modelo OSI Y Modelo TCP/IP [29]

El Modelo TCP/IP utiliza 4 capas, las de aplicación y transporte sólo se usan en el *host* origen y *host* terminal. Mientras que las otras dos, red y acceso a la red deben de estar presentes también en otros equipos de interconexión, como los rúteres. En cada una de ellas se utilizan diversos formatos de mensajes o paquetes que en la parte llamada cabecera define las acciones a tomar en lo que se conoce como protocolo.

Las funciones de estas 4 capas se describen a continuación:

- Capa de enlace. Las funciones principales de esta capa son: identificación de equipos mediante direcciones MAC (*Media Access Control*), control de errores mediante cadenas de redundancia CRC (*Cyclic Redundancy Check*), recuperación ante fallos mediante ARQ (*Automatic Repeat Request)*, así como funcionalidades de gestión de enlace y control de acceso al medio. Los protocolos más habituales son Ethernet, 802.11 (Wi-Fi), PPP (*Point to Point Protocol*) y *frame-relay.*

- Capa de red. Su función sería determinar el camino a seguir mediante las direcciones IP de origen y destino final. El control de errores se aplica solo a la cabecera del mensaje y no está

orientado a conexión. El protocolo IP (v4 o v6) es el que se utiliza junto con los protocolos de enrutamiento dinámico externo y interno. Los más utilizados serán OSPF (*Open Shortest Path First*) para el enrutamiento interior y BGP (*Border Gateway Protocol*) para el exterior. Protocolos auxiliares como el ICMP (*Internet Control Message Protocol*) envían mensajes de error e información sobre situaciones excepcionales o anómalas sobre la capa de red.

- Capa de transporte. Su función consiste en establecer una conexión de extremo a extremo entre aplicaciones (software) identificadas por puertos (es un número de 16 bits que generalmente es proporcionado por el sistema operativo o, en el caso de servidores, usando puertos estándar bien conocidos). Los principales protocolos son TCP y UDP. TCP (*Transmission Control Protocol*) está orientado a conexión, es un protocolo fiable con control de errores mediante sumas de comprobación (*checksum*) y recuperación de errores mediante ARQ (*Automatic Repeat Request*) contando también con control de flujo y congestión. Mediante UDP (*User Datagram Protocol*) las aplicaciones envían datagramas encapsulados sin conexión, aunque tiene control de errores con *checksum*.

- Capa de aplicación. Su función son los servicios de red a aplicaciones e incluye protocolos como Telnet, SSH (*Secure Shell*), HTTP (*Hypertext Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*), DHCP *(Dynamic Host Configuration Protocol*), DNS (*Domain Name System*), etc. Existen unas API (interfaces de programación estandarizadas) que son los *sockets* de Berkeley que se utilizan en todas las aplicaciones para programar sobre TCP o UDP, permitiendo al programador escribir código portable sin preocuparse por la arquitectura del host a través de librerías.

En la Figura 1-4 podemos ver la pila de protocolos de un servicio web ejecutándose sobre el protocolo TCP y Ethernet.

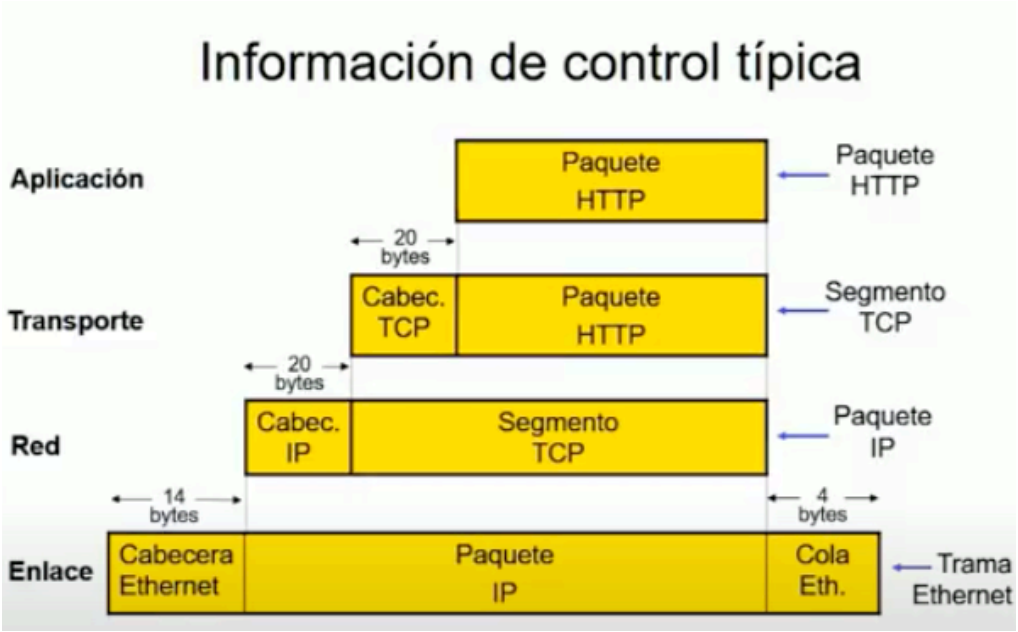


Figura 1-4 Información de control típica [29]

En la tabla 1-1 podemos ver las capas y algunos de los protocolos más utilizados en cada una de ellas.

Aplicación	WEB (HTTP), Transferencia de Ficheros (FTP)	DHCP, Resolución nombres (DNS)
Transporte	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
Red	IP (Internet Protocol) IPv4 e IPv6	
Enlace	Ethernet, WIFI, ADSL, CATV	
Física	Cable o Fibra, radio, cable telefónico, cable coaxial	

Tabla 1-1 Algunos protocolos utilizados actualmente en Internet (Autoría propia)

En este proyecto se construirán maquetas físicas en donde se ejecutarán diversos protocolos para estudiar e ilustrar su funcionamiento mediante analizadores de tráfico (*packet sniffers*). En concreto se considerarán los siguientes protocolos:

- ARP (*Address Resolution Protocol*). Determina la dirección MAC (dirección física) de un nodo a partir de su dirección IPv4 en una red.
- DHCP *(Dynamic Host Configuration Protocol)*. Es un protocolo sobre la capa de aplicación de tipo cliente/servidor mediante el cual un servidor DHCP asigna dinámicamente una dirección IP y otros parámetros de configuración de red a dispositivos de una red en situación de clientes.
- HTTP (*Hypertext Transfer Protocol*) protocolo utilizado para acceso a páginas web.
- ICMP (*Internet Control Message Protocol*). Detecta y registra condiciones de error en la red, tales como, por ejemplo: destino inalcanzable (*destination unrechable*) o tiempo excedido (*time exceded*) entre otros.
- 802.1q utilizado en las VLAN (*Virtual Local Area Network*).
- OSPF (*Open Shortest Path First*). Es un protocolo de encaminamiento interior IGPs (*Interior Gateway-Routing Protocols*) entre rúteres.

Internet es la capa física o red formada por conmutadores (*switches*), rúteres y otros equipos con la función de transportar información de un punto a otro de forma rápida, fiable y segura, se organiza [2,3] sobre un conjunto de sistemas autónomos (*Autonomous Systems*, AS) interconectados por un protocolo de enrutamiento dinámico EGP (*Exterior Gateway Protocol*), utilizando lo que se conoce como *Path-Vector* expresado en el protocolo BGP (*Border Gateway Protocol*). Los sistemas autónomos son un conjunto de redes IP y rúteres que se encuentran bajo el control de una misma entidad denominada habitualmente ISP (*Internet Service Provider*) y que poseen una política de enrutamiento concreta. A diferencia del protocolo externo de enrutamiento mundial BGP, los AS utilizan protocolos de enrutamiento interno. OSPF es uno de ellos el cual proveyéndose del algoritmo de Dijkstra encuentra lo que se conoce como *Shortest Path.* El OSPF no es el único protocolo IGP (*Interior Gateway Protocol*), hay otros como el RIPv2 utilizado solo para pequeñas redes o el EIGRP de Cisco.

El intercambio de tráfico de red IP a nivel regional, nacional e internacional se conoce como *peering*. Un punto de intercambio IXP [4] es una instalación en la interconexión de redes, generalmente un conmutador Ethernet en que todos los participantes establecen sesiones BGP entre sus redes. La Internet actual tiene 325 IXP en el mundo y más de 100.000 AS.

1.3 Docencia de redes

La enorme importancia de las redes de ordenadores en el panorama militar actual conlleva la necesidad de formar al personal de la Armada en este ámbito. Está para ello la asignatura de 'Redes de ordenadores' que se imparte en 5º curso del grado de Ingeniería Mecánica con intensificación naval en la ENM. En las prácticas de la asignatura los alumnos desde la aplicación de virtualización VirtualBox y con la máquina virtual residente en ella con sistema operativo Linux (Ubuntu 18.04 LTS Desktop 64 bits), abordan la configuración y exploración de red mediante las herramientas: ifconfig, nmap y traceroute. En el laboratorio de ciberseguridad de la asignatura se estudia cifrado de información con los programas GPG (AES) y ROT13, el programa *John the Ripper* para romper contraseñas, la detección de intrusiones con el comando sha1sum de Ubuntu que permite calcular la huella de un fichero y comprobar así si el contenido ha sido modificado. Sin embargo, los estudiantes no disponen de un escenario de pruebas de laboratorio desde el punto de vista experimental de construcción física de redes, la actividad más similar que se realiza actualmente es el uso del software de simulación GNS3 en modo local en rúteres IOS de Cisco para configurar una VLAN en otra de las actividades de laboratorio de la asignatura.

En los centros politécnicos de nuestro país el software para la simulación de redes varía desde el *Riverbed Modeler* [30], en su versión académica (*Riverbed Modeler Academic Edition*) utilizado en la UVa, hasta los más utilizados *Cisco Packet Tracer* [22] en la UPV entre otras, junto con el GNS3 [8] que se está convirtiendo en dominante al ser *Open Source* y posibilitar la simulación de los sistemas operativos virtualizados de los rúteres. En los analizadores de paquetes es predominante el uso de *Wireshark* [12].

Por otra parte, debido a la relevancia de las metodologías activas de aprendizaje, como el Aprendizaje Basado en Proyectos (ABP) [17] que involucran al estudiante en el proceso consiguiendo fomentar la motivación, en las escuelas de ingeniería se utiliza cada vez más. En la ETSETB (*Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona)* tienen el modelo docente CDIO [17] (*concebre-dissenyar-implementar-operar*). Consiste en desarrollar en equipo soluciones a problemas complejos, definiendo un plan de negocio y realizando prototipos implementados físicamente de la solución propuesta. El CDIO trata de generar un entorno próximo al ejercicio profesional de la ingeniería como contexto ideal para el aprendizaje de la ingeniería. En todo el mundo, los centros colaboradores de la iniciativa CDIO la han adoptado como marco de referencia para la planificación curricular y la evaluación basada en los resultados de aprendizaje. El CDIO nació en 2000 en el MIT [31] con el objetivo de reducir la distancia existente entre el perfil de salida de sus ingenieros y las necesidades de la industria, actualmente hay más de 50 instituciones en más de 25 países que son miembros en diferentes grados de la iniciativa.

El uso de equipos reales de redes en la formación académica entraría dentro de lo que se conoce como aprendizaje basado en problemas y casos concretos, al ser un escenario más realista que el uso de los simuladores. Este es el porqué de construir diversas maquetas de redes en donde comprobar experimentalmente el funcionamiento de los protocolos más importantes con la finalidad de obtener un instrumento con el que complementar el aprendizaje teórico de una asignatura de redes.

1.4 Objetivos del trabajo

El objetivo principal consiste en diseñar montajes (maquetas) con equipos reales de rúteres MikroTik, conmutadores (s*witches*) y equipos finales representados por Raspberry Pi (RPi) (alguna funcionando como servidor), que permitan a los estudiantes comprobar físicamente en ellas el funcionamiento de protocolos como ARP, HTTP, DHCP, 802.1q, OSPF, etc. Utilizaremos además como herramientas auxiliares otras del ámbito de la gestión y análisis de redes como GNS3 a la hora de planificar el diseño de los montajes o Wireshark para capturar e inspeccionar el intercambio de paquetes de los protocolos. Para alcanzar este objetivo principal se empleará una metodología ágil, basada en pequeños desarrollos de complejidad incremental y reuniones de coordinación periódicas con los tutores, permitiendo avanzar a través de una serie de actividades a realizar, entre las que se incluyen:

- Estudiar el estado del arte de las herramientas de apoyo al aprendizaje de redes.
- Familiarizarse con los mecanismos de configuración de rúteres MikroTik ofrecidos por su sistema operativo RouterOS.
- Diseñar montajes en los que se pueda observar el funcionamiento de protocolos tales como: ARP, DHCP, HTTP, ICMP, 802.1q y OSPF.
- Validar los montajes previamente diseñados mediante su implementación física y análisis de su funcionamiento.

1.5 Estructura de la memoria

La presente memoria se estructura en varios capítulos:

- En el capítulo 1: Introducción y Objetivos. Se busca contextualizar el trabajo, así como exponer los objetivos de éste. Se hace una introducción a los orígenes de Internet para pasar después a la arquitectura de redes (capas y protocolos). Se discuten metodologías activas de aprendizaje de redes basadas en problemas y casos concretos. Se propone construir diversas maquetas de redes en donde comprobar experimentalmente el funcionamiento de los protocolos más importantes, con la finalidad de obtener un instrumento con el que complementar el aprendizaje teórico de una asignatura de redes.
- En el capítulo 2: Estado del arte. Afrontaremos, en primer lugar, el estudio de los principales simuladores de redes y de analizadores de paquetes con una comparativa entre ellos. A continuación, se exponen los cursos de formación de aprendizaje de enrutamiento y manejo de redes cableadas e inalámbricas de las empresas MikroTik y Cisco. Para terminar con la descripción de los equipos MikroTik y su sistema operativo RouterOS que es el que se utilizará en el proyecto.
- En el capítulo 3: Diseño y desarrollo. Dado que se muestra el funcionamiento y validación de cada maqueta justo después de su diseño se ha decidido integrar en un solo capítulo los temas de desarrollo y prueba. Se diseñan 5 montajes (maquetas) con equipos reales de rúteres MikroTik, conmutadores y equipos finales Raspberry Pi. Probándose en ellos experimentalmente los protocolos: ARP, DHCP, HTTP, 802.1q en VLAN y OSPF con un analizador de paquetes.

- En el capítulo 4: Conclusiones y líneas futuras. Se describen las principales conclusiones alcanzadas con el trabajo y se proponen nuevas maquetas como líneas futuras de ampliación en las que probar otros protocolos.

2 ESTADO DEL ARTE

Como parte del proceso de documentación para realizar este TFG, se ha investigado qué opciones hay disponibles en la actualidad sobre herramientas de simulación de redes y analizadores de paquetes. Se compararán las características, ventajas e inconvenientes de los diversos programas informáticos que nos permiten implementar teóricamente modelos de redes con muchos dispositivos que sería imposible montar en el aula o laboratorio. Los diversos programas que controlan y analizan el tráfico de red (*sniffers)* son un complemento necesario a los simuladores pues generan información de errores, fallos de seguridad, cuellos de botella y detección de intrusos, incluso para la docencia resultan imprescindibles. De los rúteres MikroTik se tratarán sus prestaciones y su sistema operativo además de los cursos de certificaciones MikroTik que se imparten, finalmente se describirán los que imparte Cisco.

2.1 Herramientas de simulación de redes

Los simuladores de redes [20] se emplean en la etapa de diseño antes de que el sistema sea construido, o para intentar predecir el efecto de un cambio o variante que queramos introducir en un sistema ya construido. La simulación de diversos escenarios nos permite validar si el diseño teórico es correcto.

Los rúteres en su configuración virtualizada utilizan el mismo sistema operativo que el del fabricante en sus dispositivos físicos. Por tanto, disponen de las mismas posibilidades en cuanto a protocolos de enrutamiento, configuración de interfaces, VPNs, NAT para acceder a Internet, DHCPs como servidor o cliente en sus interfaces, seguridad, cortafuegos (*firewall*)…etc.

2.2 Estudio de simuladores de redes

Estos son los principales simuladores de redes que se han encontrado, de los cuales vamos a analizar algunos.

- CISCO VIRL
- CISCO PACKET TRACER
- GNS3
- NETSIM
- NETSIMK
- NS-2 y NS-3
- JIMSIM
- CORE

- KIVA NS
- CLOONIX
- MININET
- RIVERBED MODELER
- OMNET++
- MARIONNET

2.2.1 CISCO VIRL (VIRTUAL INTERNET ROUTING LAB)

Cisco VIRL [21] está destinado a estudiantes que quieran obtener las certificaciones Cisco para dedicarse al entorno profesional de instalación y mantenimiento de redes. VIRL usa su propia gama de rúteres y conmutadores de la empresa. Tiene una comunidad virtual que proporciona foros de discusión entre sus miembros para la solución de problemas. Usa el sistema operativo Cisco IOS en sus rúteres. Cuenta con una amplia documentación VIRL como vídeos de demostración y guías. Usa como aplicación de virtualización VM Ware Fusion (Mac) o Workstation. Ni VIRL ni VMWare son gratuitos, hay un pago por uso de licencias que incluyen varias posibilidades de contratación.

Figura 2-1 Logo Virtual Internet Routing Lab & VMware [21,11]

2.2.2 CISCO PACKET TRACER

Es una potente plataforma de simulación de redes [22] [23] con objeto de ser utilizada como complemento a laboratorios de equipos reales. Es ideal para motivar a estudiantes en el diseño de redes. Este programa es utilizado en muchas universidades en su asignatura de redes y compite con el *GNS3*. Es fácil de manejar y permite ver el desarrollo por capas al tener un *sniffer* propio, diferente del *GNS3* en el que hay que instalar un analizador de paquetes (Wireshark). La disposición de los diferentes equipos en un entorno espacial de simulación ayuda en una representación visual acabada. El programa es gratuito, de fácil instalación, adaptado a plataformas Windows, Linux y funciona perfectamente en Mac sobre la aplicación de virtualización VMWare.

Figura 2-2 Logo Cisco [23]

2.2.3 GNS3

GNS3 [8] es utilizado por profesionales para configurar, probar y solucionar problemas de redes virtuales y reales. Es una aplicación *open source* con lo que su código es accesible al público: todos pueden ver, modificar y distribuir el código de la forma que consideren conveniente.

La arquitectura de *GNS3* consta de dos componentes de software:

i. El software GNS3-all-in-one que contiene la interfaz gráfica del usuario (GUI) y es la parte cliente.
ii. La máquina virtual GNS3 (VM) hay que ejecutarla usando un software de virtualización de VMware o VirtualBox. En ella se instalarán las imágenes de los sistemas operativos de rúteres, u otros equipos de interconexión o finales.

Figura 2-3 Logo GNS3 [8]

2.2.4 Otras herramientas

- NETSIM [32] simula el funcionamiento de software de Cisco. Nos permite conectar el simulador a hardware real e interactuar con las aplicaciones a través de una red virtual.

- NETSIMK [20] es un simulador para la enseñanza y el aprendizaje del rúter Cisco. Se centra en enseñar cosas básicas sin necesidad de configurar opciones innecesarias de dispositivos. La red puede tener conectada tantos terminales y rúteres como sea necesario. Todos los aspectos más importantes del IOS (sistema operativo de los rúteres de Cisco) están implementados, permitiendo así que los usuarios que lo deseen puedan obtener las certificaciones de Cisco practicando sobre ellos.

- NS-2 y NS-3 [33] han sido desarrollados por universidades y están dirigidos a la investigación de redes. NS-2 se creó en 1989, está programado en C y se puede instalar en los sistemas operativos Unix y Linux (Debian y Ubuntu). Existe ahora el NS-3 que está implementado en C++ desarrollado también por universidades y ofrece un mayor número de eventos, pero es diferente de NS-2, los proyectos no se pueden portar de uno al otro.

- JIMSIM SIMULATOR [34] su código está en Java lo que lo hace independiente de la plataforma. Es capaz de hacer procesos de simulación en tiempo real, aunque tiene una realización parcial de la emulación de la red.

- CORE (*Common Open Research Emulator*) [35] ha sido desarrollado por la división de tecnología de *U.S Naval Research Laboratory*. Puede emular redes y conectarlas a otras que se ejecutan en tiempo real y es muy configurable ejecutando aplicaciones y protocolos sin modificarlos.

- KIVANS (*Kiva Network Simulator*) [36] es muy útil para la simulación de datagramas y el encaminamiento de estos por la red, aunque también se utiliza para la arquitectura TCP/IP. Está escrito en Java y funciona en múltiples sistemas operativos.

- CLOONIX [37] ofrece simulación de rúteres y huéspedes. Da la posibilidad de comprender protocolos como el DNS y su interfaz gráfica es amigable.

- MININET [38] está orientado a la investigación y al aprendizaje. Es un simulador muy útil, gratuito, fácil de instalar y de configurar.

- OMNET++ [39] se utiliza para modelar protocolos, redes de comunicación y colas. Está escrito en C++ y tiene su propia biblioteca.

- MARIONNET [40] de propósito docente en universidades y fácil de configurar. Se comporta muy bien con redes complejas.

- RIVERBED MODELER [30] investigadores del Instituto de Tecnologías de Massachusetts lo desarrollaron a través de la teoría de redes de colas. Se ofrece de manera gratuita a las universidades en su versión *Riverbed Modeler Academic Edition,* pero no tiene los servicios

de la versión de pago. Es por esto por lo que éstas prefieren utilizar GNS3 u otros simuladores de redes.

2.2.5 Comparativa de herramientas

A continuación, se comparan los simuladores de redes. (Tabla 2-1 y Tabla 2-2)

Características / Simuladores	Licencia	Sistema Operativo	Desarrollador	Lenguaje de programación	Subnetting	OSPF	VLAN
Cisco Packet Tracer	Propietario	Linux, Windows	Cisco	-	sí	sí	sí, permite trunk
Cisco Virl	Propietario orientado al aprendizaje	Windows y con la máquina virtual en Windows y Linux	Cisco Systems	Java	sí	sí	sí, trunk limitado
GNS3	Software libre	Multiplataforma: Windows, Mac y Linux	-	Python	sí	sí	sí, permite trunk
Riverbed Modeler	Propietario	Windows	Riverbed	C++	sí	sí	sí, permite trunk
NETSIM	Versiones Standard, Pro y Academic	Windows	-	Python	sí	sí	sí, permite trunk
NETSIMK	Versiones de evaluación gratuita	Windows	Cisco	C++	sí	no	sí, permite trunk
NS-2 y NS-3	Licencia pública GNU	Windows Y Unix	Comunidad	C++ y Python	sí	sí	sí

Tabla 2-1 Comparación de simuladores de redes parte I (Autoría propia)

Características / Simuladores	Licencia	Sistema Operativo	Desarrollador	Lenguaje de programación	Subnetting	OSPF	VLAN
JIMSIM	Código abierto	Windows y Linux	Universidad de West Bohemia República Checa	C++ y Python	sí	sí	sí, similar a la tecnología CISCO sobre VLAN
CORE	Open Source	Linux	U.S Naval Research Laboratory	Python	sí	sí	sí
KIVA NS	Código abierto	Windows y Linux	Universidad de Alicante	Java	sí	no	no
CLOONIX	Código abierto	Linux	CLOONIX	Java	sí	sí	sí
MINIMET	Open Source	Windows, Mac y Linux	Bob Lantz	Python	sí	sí	sí
OMNET++	Licencia Pública	Multiplataforma	Opensim	C++	sí	sí	sí, tiene librerías para implementación
MARIONNET	GNU	Windows y Linux	Jean Loddo	Ocaml	sí	sí	sí, a nivel de puerto

Tabla 2-2 Comparación de simuladores de redes parte II (Autoría propia)

2.3 Analizadores de Paquetes

Son programas de captura de paquetes (*sniffer*) y se usan para analizar el tráfico en la red y a través de ellos podemos descubrir problemas en la red como cuellos de botella o detección de intrusos y analizar la información que se transmite convirtiéndolo todo en un formato entendible. Los principales son:

- Wireshark
- Microsoft Message Analyzer

2.3.1 Wireshark

Es el analizador de protocolos de red [12] más utilizado, aparte del análisis de tráfico es una excelente aplicación didáctica y sirve para la resolución de problemas de red. La Figura 2-5 corresponde a la pantalla inicial en la cual podemos seleccionar la interfaz de red que nos convenga para el análisis de los paquetes.

Figura 2-4 Logo Wireshark [12]

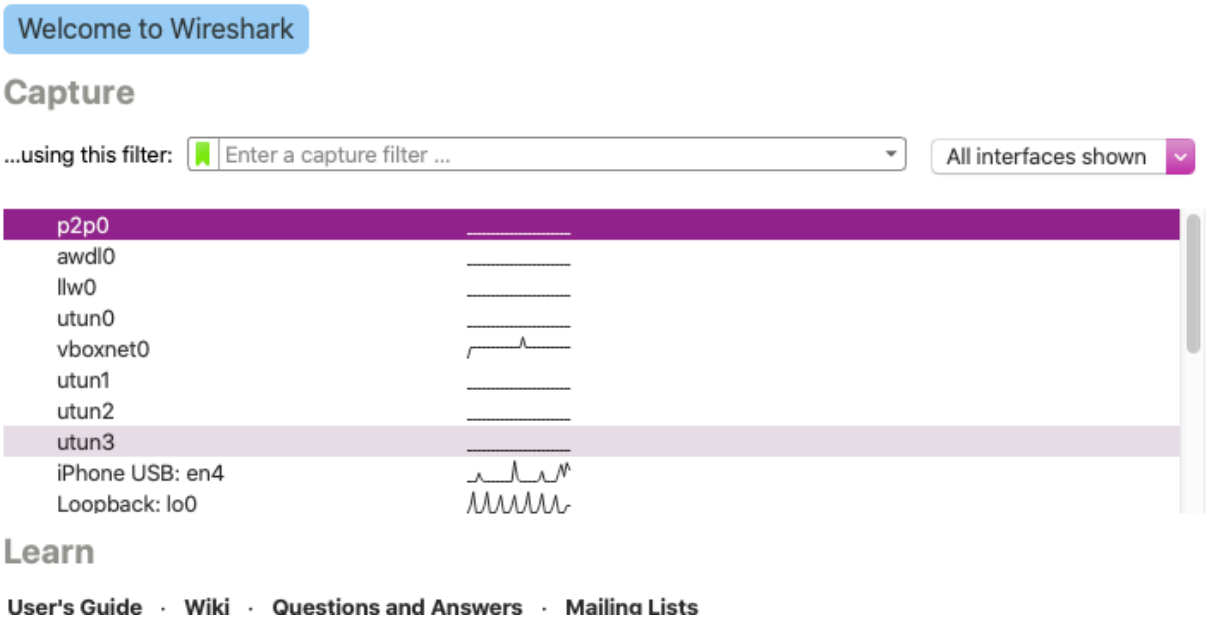


Figura 2-5 Captura con Wireshark (Autoría propia)

Seleccionando una interfaz nos aparece la pantalla principal mostrada en la Figura 2-6.

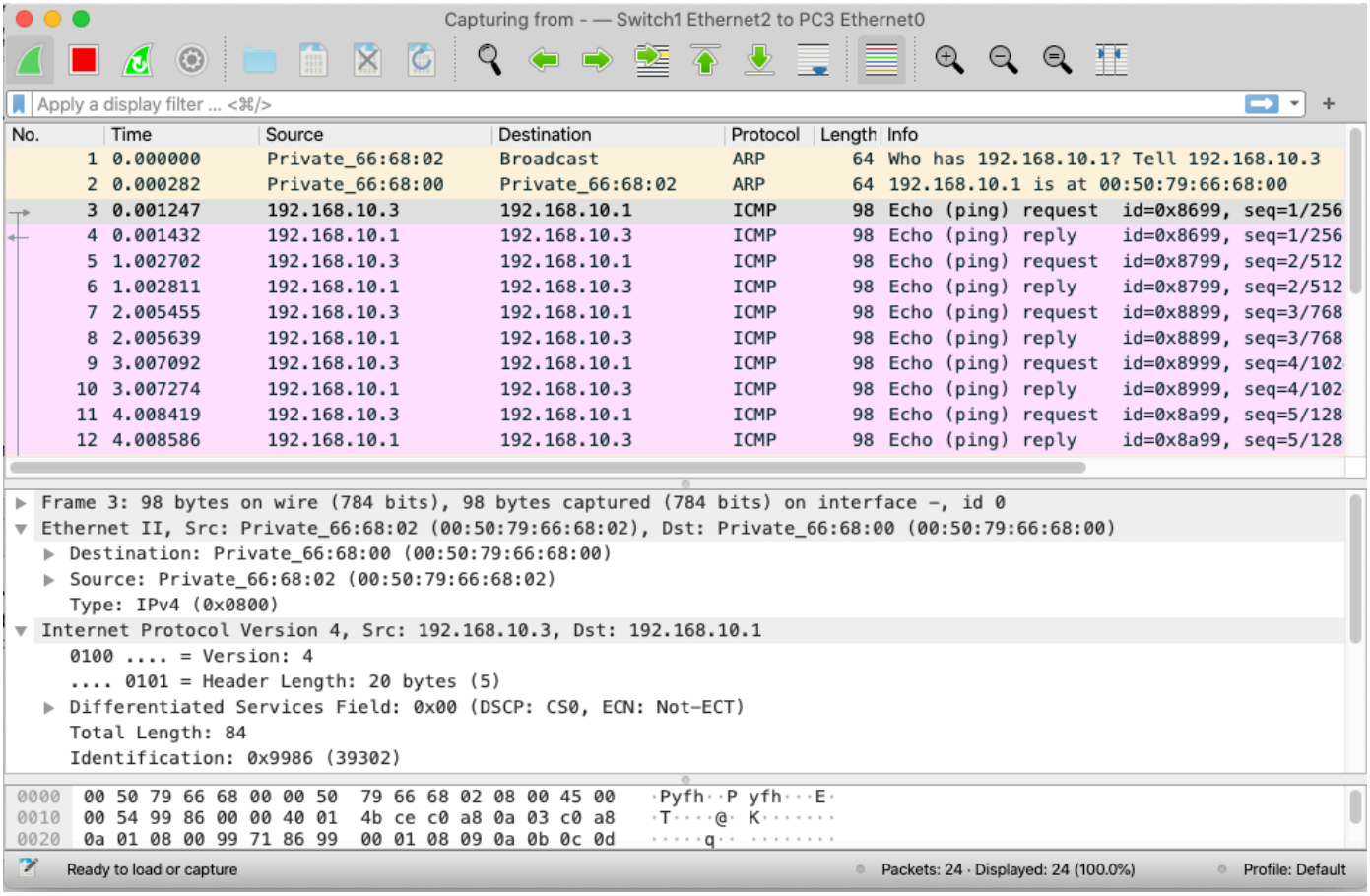


Figura 2-6 Captura de Wireshark en la interfaz virtual (Autoría propia)

La pantalla tiene tres partes o ventanas, la parte de arriba nos muestra los paquetes capturados en tiempo real desde el momento en que empieza la captura y en el orden de llegada para cada paquete. Nos muestra las IP de la fuente (*Source)* y del destino (*Destination),* el protocolo al que pertenece, la longitud y otra información adicional. Esta ventana primera es como una especie de resumen - cada paquete está representado por una línea -, si seleccionamos una línea nos muestra en la segunda ventana el contenido del paquete y su estructura. En esta ventana podemos expandir las líneas para obtener información adicional. La tercera ventana muestra el contenido de todo el paquete en hexadecimal y en ASCII, resaltando el campo seleccionado en la segunda ventana. No todos los caracteres hexadecimales tienen una correspondencia con los ASCII imprimibles, pero se pueden leer los mensajes de texto plano. Wireshark es una herramienta muy útil para analizar el contenido de los paquetes, y con la utilización de los filtros adecuados se puede diagnosticar un problema. Podemos ver por ejemplo los paquetes con una IP de destino o una MAC dadas.

A nivel estadístico, la aplicación Wireshark proporciona desde el menú Statistics información general de los paquetes capturados según sus protocolos. La opción es *Protocol Hierarchy* es mostrada en la Figura 2-7**.**

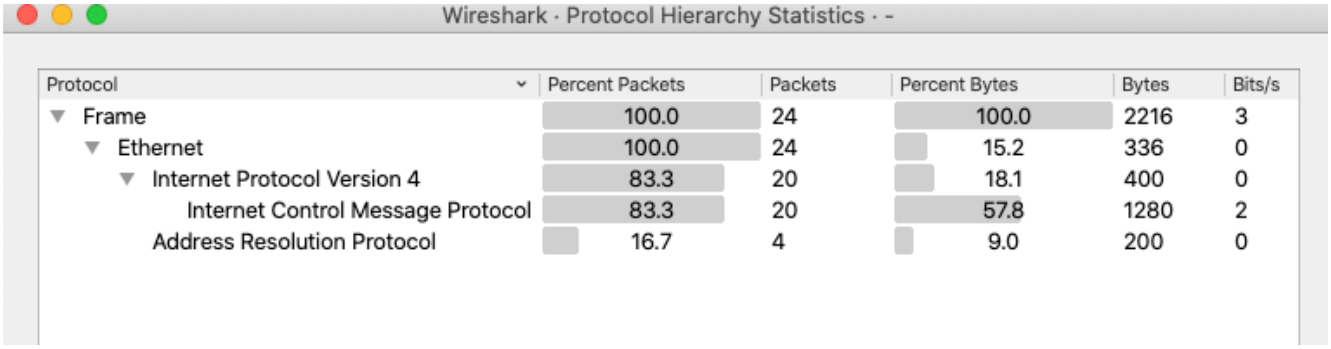


Figura 2-7 Estadísticas tras la realización de dos pings entre dos ordenadores. (Autoría propia)

2.3.2 Microsoft Message Analyzer

Microsoft Message Analyzer [41] es una herramienta para mostrar datos de registros o capturas de datos, seguimiento en una variedad de visores de datos y de vistas gráficas seleccionables. No solo es eficaz para situaciones de red, sirve también para probar y verificar implementaciones de protocolos. La aplicación se puede descargar de forma totalmente gratuita y debemos ejecutarla con permisos de administrador ya que, de lo contrario, no capturará paquetes.

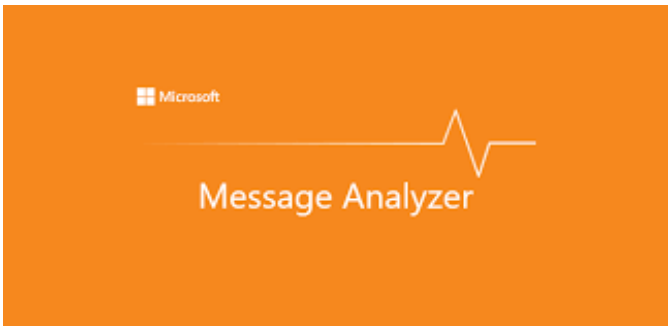


Figura 2-8 Logo Message Analyzer [41]

2.4 MikroTik Training and Cisco Courses

2.4.1 MikroTik Training Courses

MikroTik [13] ofrece cursos certificados que son reconocidos mundialmente para aquellos que quieran tener un buen conocimiento acerca de RouterOS. Existen centros donde se imparten cursos de formación de MikroTik en diversas partes del mundo. En estos cursos se aprende a enrutar y manejar redes cableadas e inalámbricas usando MikroTik RouterOS. En la Figura 2-9 se muestra la ubicación de los diversos sitios donde se imparten cursos MikroTik.

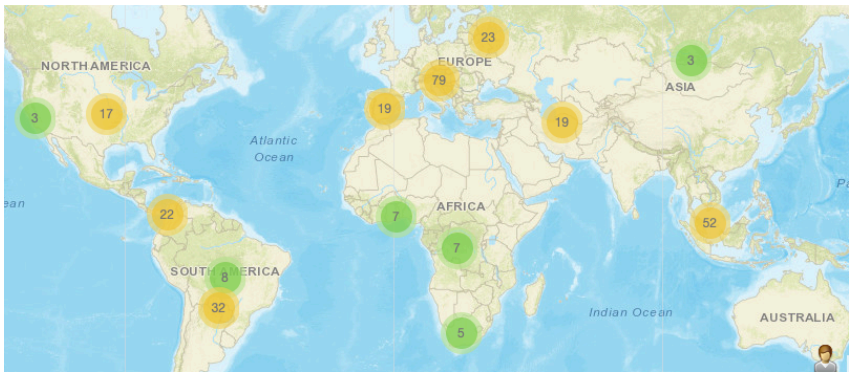


Figura 2-9 Distribución de los cursos certificados MikroTik [42]

Los cursos MikroTik [42] atraen a muchos estudiantes ya que sus certificados son reconocidos por muchas empresas a nivel mundial. Las clases consisten en lecciones teóricas y prácticas sobre distintos temas: manejo de redes e IP, así como su implementación en equipos MikroTik. El contenido total versa sobre una introducción a redes IP, MikroTik Esencial y MikroTik Wireless. Existen 10 niveles de certificaciones MikroTik:

- Certificado MTCNA (*Mikrotik Certified Network Assocciate)*
- Certificado MTCRE (*MikroTik Certified Routing Engineer*)
- Certificado MTCUME (*MikroTik Certified User Management Engineer*)
- Certificado MTCINE (*MikroTik Certified Inter Networking Engineer)*
- Certificado MTCWE (*MikroTik Certified Wireless Engineer*)
- Certificado MTCIPv6E (*MikroTik Certified IPv6 Engineer*)
- Certificado MTCSE (*MikroTik Certified Security Engineer*)
- Certificado MTCTCE (*MikroTik Certified Traffic Control Engineer*)
- Certificado MTCSWE (*MikroTik Certified Switching Engineer)*
- Certificado MTCEWE (*MikroTik Certified Enterprise Wireless Engineer*)

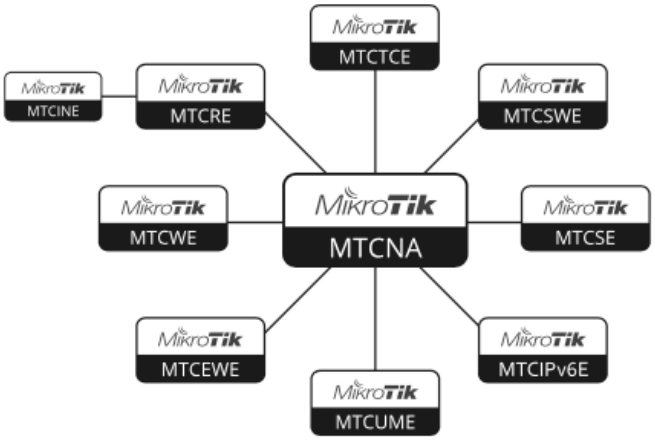


Figura 2-10 Certificados MikroTik [42]

La Certificación MTCNA es la primera que se obtiene y es requisito para obtener las siguientes certificaciones. Entre las certificaciones avanzadas se encuentra MTCRE para el enrutamiento. Se trabaja VLAN, direccionamiento punto a punto, túneles VPN y OSPF para el certificado MTCNA. Tal y como se muestra en la programación de la Figura 2-11.

Figura 2-11 Programación de la Certificación MTCNA [42]

2.4.2 Cisco Courses

Cisco también ofrece certificaciones [23] para aquellos que deseen alcanzar los conocimientos necesarios sobre redes. Estos certificados son reconocidos mundialmente y demuestran que se tienen habilidades para configurar y administrar redes Cisco, así como solucionar sus problemas.

Actualmente hay cinco niveles de certificados Cisco (Figura 2-12):

- Entry: CCENT (*Cisco Certified Entry Networking Technician)*
- Associate: CCNA (*Cisco Certified Networking Associate*)
- Professional: CCNP (*Cisco Certified Networking Professional)*
- Expert: CCIE (*Cisco Certified Internetwork Expert)*
- Architect: CCAr (*Cisco Certified Architect*)

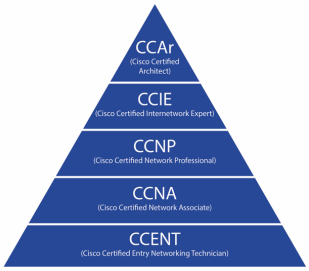


Figura 2-12 Niveles de Certificados Cisco [23]

La certificación más básica empieza por CCENT dirigida a estudiantes que se encuentran en la escuela secundaria o la universidad, el siguiente nivel sería CCNA dirigido a aquellos que empiezan en el mundo laboral y cuentan con dos años de experiencia mínimo y, posteriormente, con CCNP se acredita un nivel profesional cuando se tiene entre dos y cinco años de experiencia. Para aquellos que quieran acreditar un nivel experto está la certificación CCIE orientada a ingenieros de redes. Por último, está la certificación CCAr destinada a aquellos que manejen redes complejas y tiene como requisito tener la acreditación de nivel experto CCIE.

Un ejemplo del programa de CCNA que se imparte en Barcelona sería el mostrado en la Figura 2-13, tiene una duración de 120 horas y un precio de 1450€.

MÓDULO 1.- Conceptos básicos de Redes
1- Introducción a las Redes de Ordenadores
2- Conceptos de las Redes de Ordenadores
3- Medios de transmisión
4- Comprobación de los medios de transmisión
5- Instalación de cableado estructurado
6- Conceptos de Ethernet
7- Tecnologías Ethernet
8- Conmutación en Ethernet
9- Torre de protocolos TCP/IP y direccionamiento IP
10- Conceptos de routing y subredes
11- Nivel de Aplicación

MÓDULO 2.- Routers y Encaminamiento
1- Redes WAN y routers
2- Introducción a los routers
3- Configuración de routers
4- Otros dispositivos
5- Gestión del software IOS
6- Protocolos de encaminamiento
7- Protocolos de encaminamiento de tipo "Distance Vector"
8- Mensajes de error y de control
9- Búsqueda de errores en la configuración de los routers
10- TCP/IP
11- Listas de Control de Acceso (ACLs)

MÓDULO 3.- Introducción a la conmutación y routing avanzado
1- Introducción al encaminamiento sin clases
2- OSPF con una única área
3- EIGRP
4- Conceptos de conmutación
5- Conmutadores
6- Configuración de switches
7- Protocolo Spanning-Tree (STP)
8- Redes de Ordenadores virtuales (VLANs)
9- Protocolos de "VLAN Trunking"

MÓDULO 4.- Redes WAN
1- Gestión de direcciones IP
2- Tecnologías WAN
3- PPP
4- RDSI y DDR
5- Frame Relay
6- Introducción a la administración de redes

Figura 2-13 Programa del curso CCNA [24]

2.5 MikroTik y RouterOS

MikroTik fabrica el hardware y software en Letonia (*Latvia*) para sus rúteres. La compañía fue fundada en 1996 con el objeto de vender equipos de red en mercados emergentes y es popular en proyectos de bajo presupuesto.

2.5.1 Dispositivos MikroTik

RouterBoard es la familia de soluciones de hardware creada por MikroTik para responder a las necesidades de clientes a nivel mundial. En la hoja de características del fabricante (*data sheet)* podemos ver la capacidad de procesamiento (CPU), memoria y almacenamiento del equipo, características que determinan la elección de un equipo, aparte de las interfaces requeridas (Ethernet, de fibra o inalámbricas). En la Figura 2-14 tenemos su modelo más popular hAP series que será el que se utilizará en el laboratorio para desarrollar el proyecto.

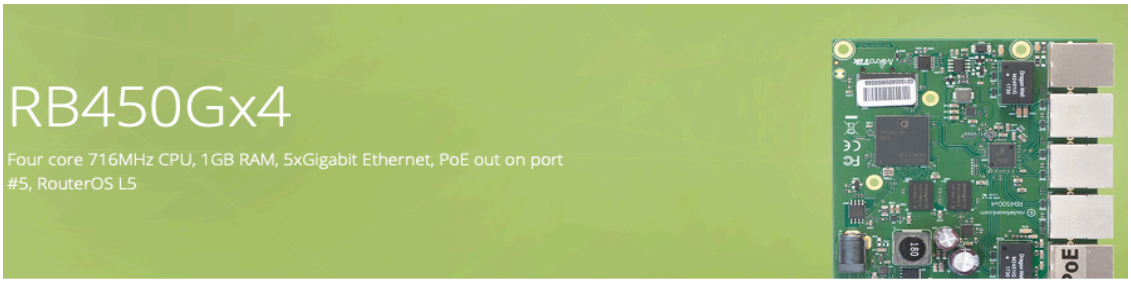


Figura 2-14 Dispositivo RB450Gx4 de RouterBoard de MikroTik [14]

2.5.2 RouterOS

Todo lo que se necesite se puede configurar en estos equipos. Su sistema operativo MikroTik RouterOS está basado en Linux Kernel v3.3.5. MikroTik tiene en sus rúteres todas las funciones habilitadas, lo único que se limita es la capacidad de procesamiento (memoria RAM y CPU) y la de brindar un servicio a una cantidad específica de usuarios y es por esto por lo que MikroTik diferencia entre sus equipos pequeños y grandes, proporcionándonos diferentes tipos de licencias MikroTik como podemos ver en la Figura 2-15, siendo ésta, otra característica en la que deberemos fijarnos para la elección de un equipo además de las mencionadas en el apartado 2.5.1 de capacidad de procesamiento, memoria e interfaces.

Level number	0 (Trial mode)	1 (Free Demo)	3 (WISP CPE)	4 (WISP)	5 (WISP)	6 (Controller)
Price	no key	registration required	volume only	$45	$95	$250
Initial Config Support	-	-	-	15 days	30 days	30 days
Wireless AP	24h trial	-	-	yes	yes	yes
Wireless Client and Bridge	24h trial	-	yes	yes	yes	yes
RIP, OSPF, BGP protocols	24h trial	-	yes(*)	yes	yes	yes
EoIP tunnels	24h trial	1	unlimited	unlimited	unlimited	unlimited
PPPoE tunnels	24h trial	1	200	200	500	unlimited
PPTP tunnels	24h trial	1	200	200	500	unlimited
L2TP tunnels	24h trial	1	200	200	500	unlimited
OVPN tunnels	24h trial	1	200	200	unlimited	unlimited
VLAN interfaces	24h trial	1	unlimited	unlimited	unlimited	unlimited
HotSpot active users	24h trial	1	1	200	500	unlimited
RADIUS client	24h trial	-	yes	yes	yes	yes
Queues	24h trial	1	unlimited	unlimited	unlimited	unlimited
Web proxy	24h trial	-	yes	yes	yes	yes
User manager active sessions	24h trial	1	10	20	50	Unlimited
Number of KVM guests	none	1	Unlimited	Unlimited	Unlimited	Unlimited

Figura 2-15 Tipos de Licencia MikroTik [14]

3 DISEÑO Y DESARROLLO

En este apartado vamos a desarrollar diversas maquetas de redes en donde se probarán diversos protocolos con equipos MikroTik y Raspberry Pi. Está compuesto por los siguientes subapartados:

i. Recursos empleados.
ii. Protocolo ARP.
iii. Protocolo DHCP.
iv. Protocolo HTTP y servidor web.
v. Protocolo 802.1q en VLAN.
vi. Interoperabilidad entre redes VLAN y redes no VLAN. Configuración estática y protocolo OSPF.

Dado que se muestra el funcionamiento y validación de cada maqueta justo después de su diseño se ha decidido integrar en un solo capítulo los temas de diseño, desarrollo y prueba.

3.1 Recursos empleados

Para el desarrollo del proyecto es necesario contar con los siguientes recursos hardware y sus sistemas operativos.

- Varias Raspberry Pi 3B (Figura 3-1) en donde se instalará el sistema operativo Raspberry Pi OS usando la Raspberry Pi Imager obtenida de su página web [28]. A continuación, se inserta una tarjeta microSD con la imagen en la Raspberry Pi 3B y el sistema operativo se inicia con la toma de corriente. Escribiendo en la terminal *sudo raspi-config* accedemos a *interfacing options* habilitando SSH, en *change user password* podemos introducir una contraseña (*password*) y en *Advanced Options>expand Filesystem* tomaremos todo el espacio de la tarjeta microSD.

Figura 3-1 Raspberry Pi 3B (Autoría propia)

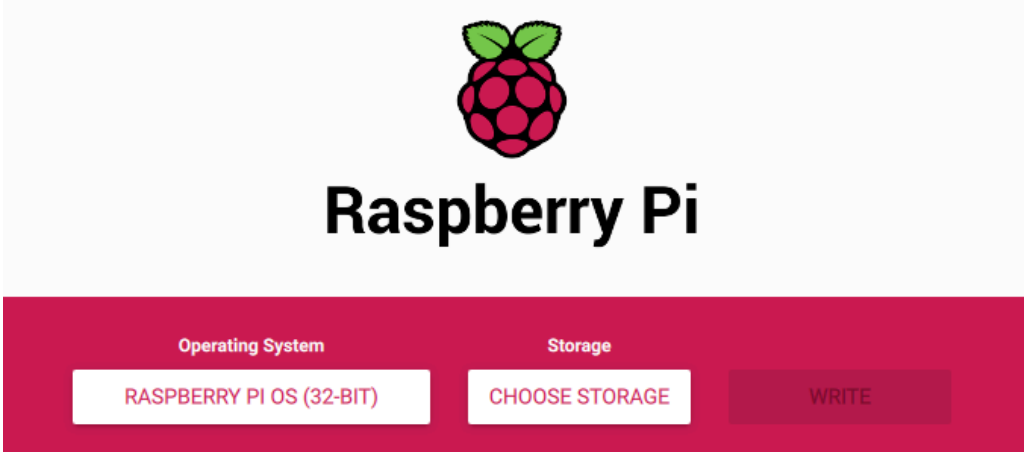


Figura 3-2 Raspberry Pi Imager (Autoría propia)

- Dispositivos Routerboard MikroTik hAP series con su sistema operativo Mikrotik RouterOS con su configuración por defecto (*default configuration*). Tiene 5 puertos y por defecto ether1 viene configurado para conectarse a WAN y el resto ether2-ether5 interconectados entre sí en modo puente. (Figura 3-3).

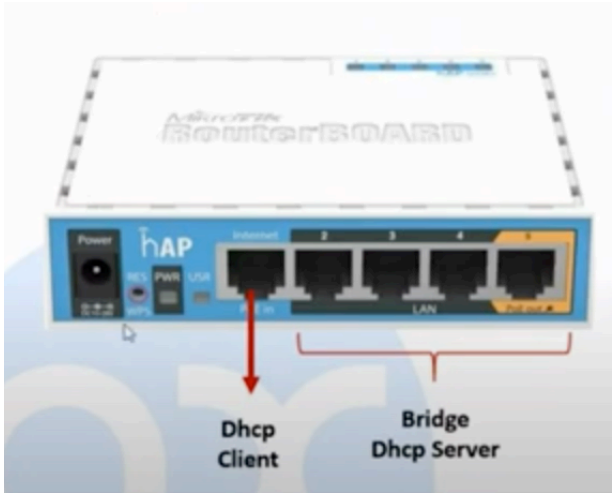


Figura 3-3 Configuración de los puertos MikroTik por defecto [13]

- Cables de red RJ45 Cat.6
- Switch TP-link 5-port Gigabit.

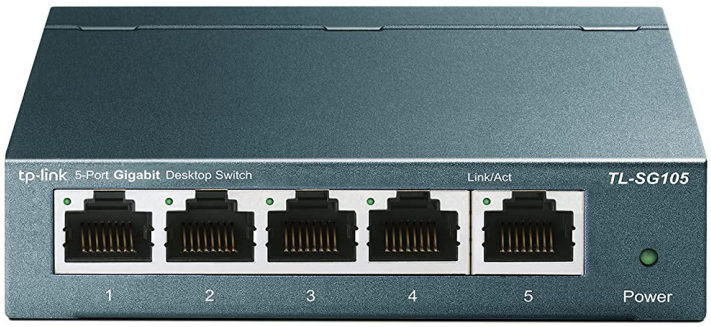


Figura 3-4 Switch TP-link-5-port Gigabit (Autoría propia).

3.2 Maqueta 1: Protocolo ARP

El protocolo ARP (*Address Resolution Protocol*) estandarizado en el RFC 826 [43], es un protocolo de comunicaciones sobre la capa de enlace encargado de determinar la dirección MAC que se corresponde a una dirección IP. Cuando una máquina pretende enviar información a otra, conociendo su dirección IP pero no su dirección MAC, antes de proceder al envío actúa el protocolo ARP en la siguiente forma:

- Estamos en el *host* de origen y queremos enviar información al *host* de destino, como no conocemos la dirección MAC del *host* de destino y el protocolo a nivel de enlace funciona solo con las direcciones MAC, el *host* de origen es incapaz de iniciar la comunicación con nadie y es por esto que ejecuta el protocolo ARP.
- El *host* de origen envía un paquete ARP Request usando como dirección de destino la de difusión (*broadcast*) FF:FF:FF:FF:FF:FF a todos los hosts de la red, preguntando quién tiene la dirección MAC que se corresponde con la IP a la que quiero mandar la información.
- El ARP Request lo reciben todos los hosts de la red local y solo el que tiene la IP pedida (*host destino*) le responde al *host origen* que esta IP es local y que corresponde a una dirección MAC de su interfaz Ethernet.
- Con esto se puede iniciar la comunicación entre los host origen y destino a nivel de enlace. La información obtenida se almacena en una zona de memoria (*caché*) por si necesita consultarse en el futuro próximo (de forma que se evita repetir peticiones ARP innecesarias).

En la Figura 3-5 se muestra el funcionamiento del protocolo ARP. Vemos que el *host* origen es la Raspberrypi2 y el de destino es la Raspberrypi1, el *Request* son los sobres en rojo y *Reply* en color verde.

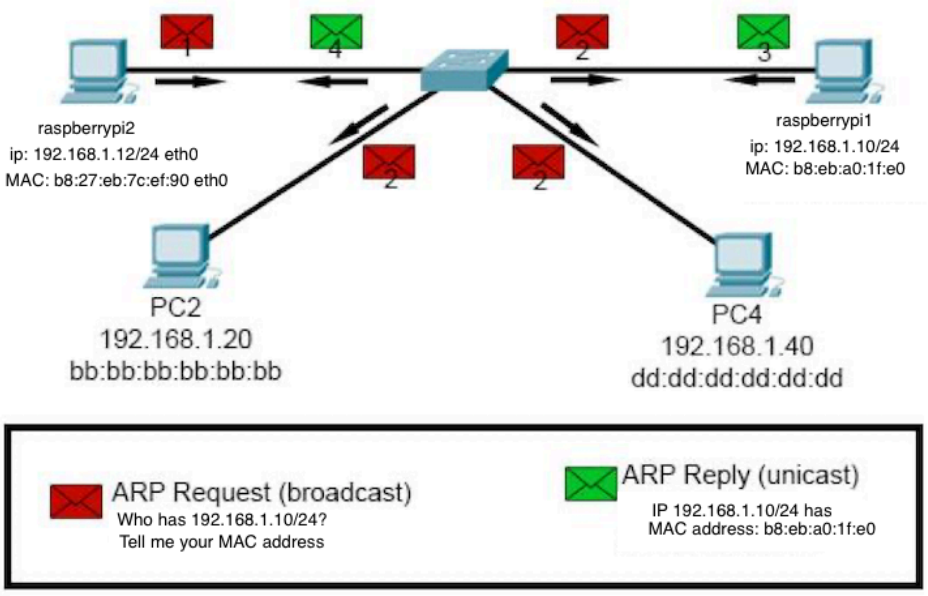


Figura 3-5 Protocolo ARP (Autoría Propia)

Para mostrar el funcionamiento del protocolo se montará una maqueta que constará de los siguientes elementos:

- 2 Raspberry Pi con el programa Wireshark instalado.
- 1 conmutador (*switch*) TP-link.
- 2 cables RJ45 Ethernet.

Vamos a omitir los 2 PC de la Figura 3-5 y a considerar 2 *hosts* en el mismo segmento de red dado que ARP es un protocolo de red local. La que implementaremos en el laboratorio será la Figura 3-6.

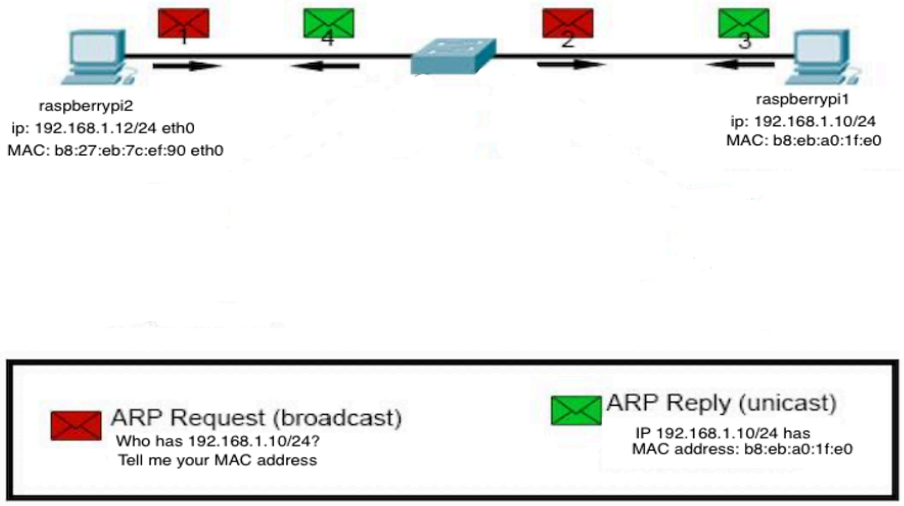


Figura 3-6 Maqueta a construir en el laboratorio para el protocolo ARP (Autoría propia)

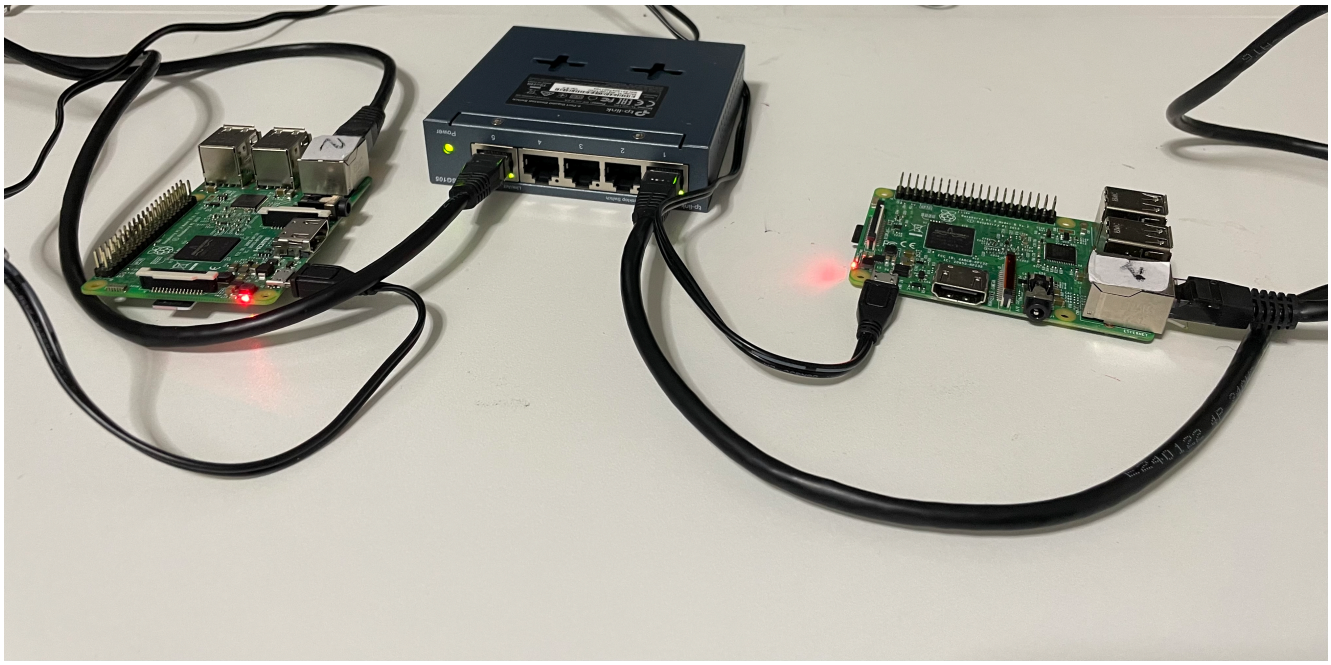


Figura 3-7 Maqueta construida para la experimentación en el laboratorio del protocolo ARP (Autoría propia)

3.2.1 Configuración IP en la Raspberry e instalación del programa Wireshark

Tras darle el nombre de Raspberrypi2 a una de las Raspberry Pi procedemos con el terminal a abrir el fichero de configuración DHCP. Dado que no se usará este protocolo, debemos asignar la configuración de red a mano, para ello editamos el fichero con *sudo nano /etc/dhcpcd.conf* y una vez dentro introducimos para la interfaz ether0 (el nombre que tiene por defecto, o el que le hubiéramos puesto de haberlo cambiado) la dirección estática IP/red en notación CIDR (*Classless Inter-Domain Routing*) 192.168.1.12/24 y guardamos el fichero con el mismo nombre (Figura 3-8). Como el fichero lo lee el sistema operativo al iniciarse (en algunas versiones del sistema operativo basta reiniciar la interfaz), efectuamos un reinicio en la terminal con el comando *sudo reboot.* Una vez reiniciado el sistema operativo comprobamos con el comando *ifconfig* desde la terminal si se han realizado los cambios en la interfaz Ethernet (Figura 3-9). El mismo proceso se hace con la otra Raspberry Pi, con el nombre Raspberrypi1 y la configuración IP/red 192.168.1.10/24.

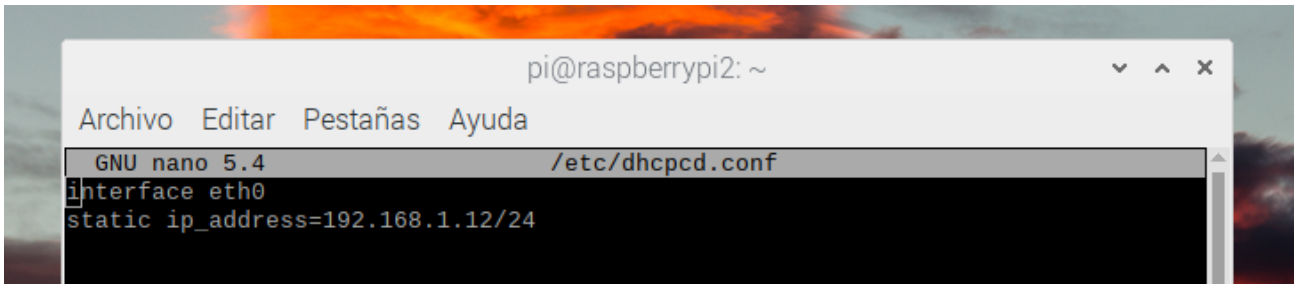


Figura 3-8 Configuración de la IP en la interfaz Ethernet en la Raspberrypi2 (Autoría propia)

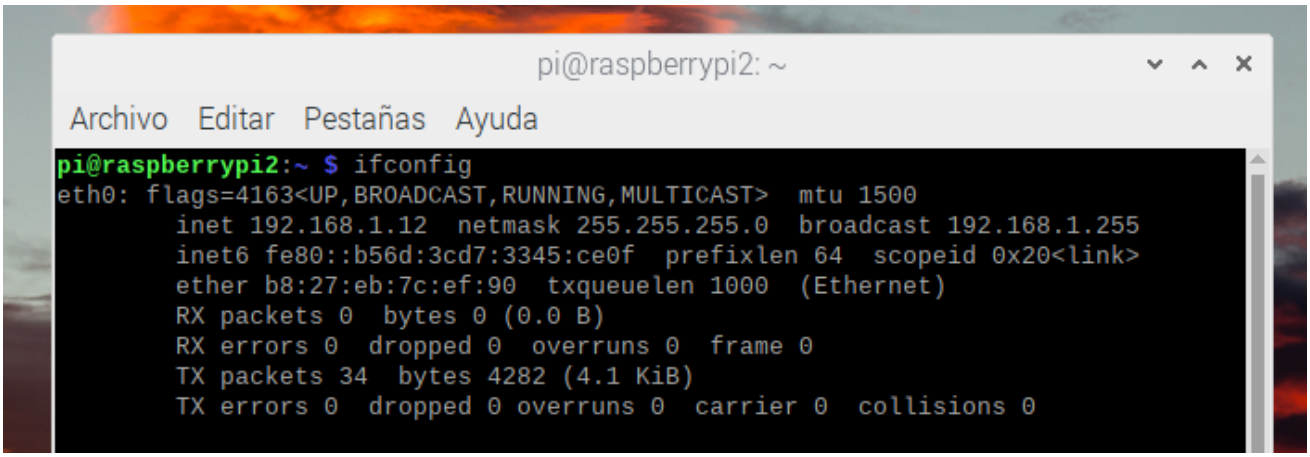


Figura 3-9 Comprobación de la IP en la Raspberrypi2 (Autoría propia)

A continuación, utilizaremos el comando *sudo apt-get install wireshark* para instalar el analizador de paquetes. El programa *arping* (que ya viene instalado por el propio sistema operativo de la Raspberry Pi) sirve para testear si un host es alcanzable dentro de la red, opera a nivel de enlace y generalmente no está bloqueado por los firewalls de los equipos o de la interfaz del rúter si la tuviera. A diferencia del protocolo *ICMP Request* (*ping*) que sí es un protocolo a nivel de red y permite alcanzar cualquier interfaz de cualquier red para testearla, mandando de vuelta un *ICMP Reply*. En este caso, aunque tenemos instalado el *arping* no lo vamos a usar porque la raspberry no va a bloquear el *ping*.

3.2.2 Comprobación experimental del protocolo ARP

Una vez configuradas, conectamos las 2 raspberry al conmutador (*switch*) a través de un cable Ethernet por sus puertos. Lo que queremos es testear la red Ethernet, para ello lo haremos desde la interfaz Ethernet de la Raspberrypi2 y comprobaremos si la interfaz Ethernet de la Raspberrypi1 es alcanzable mediante el comando *ping*.

Lo primero de todo es poner en marcha el simulador de paquetes Wireshark en la terminal de la Raspberrypi2 mediante el comando *sudo wireshark* (Figura 3-10), comprobamos que ni recibe ni se manda paquetes a través de la Raspberry2, que tiene su tabla caché ARP vacía (Figura 3-11).

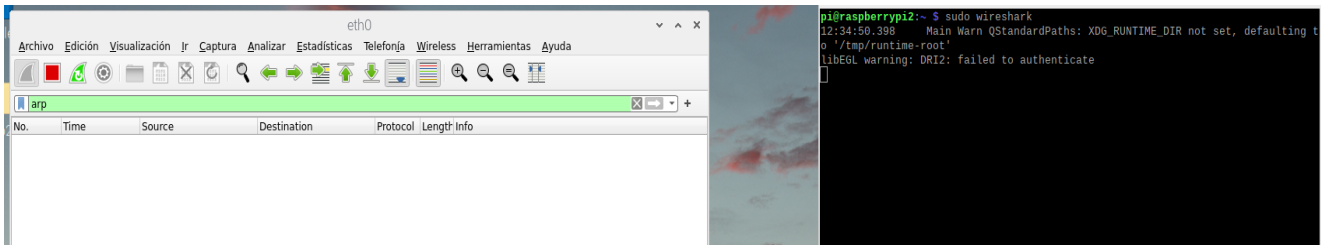


Figura 3-10 Captura Wireshark inicial (Autoría propia)

Figura 3-11 Tabla ARP Raspberrypi2 vacía (Autoría propia)

A continuación, desde la Raspberrypi2 mandamos un *ping -c 1 192.168.1.10* con un solo paquete a la dirección 192.168.1.10 que corresponde a la interfaz Ethernet de la Raspberrypi1 (Figura 3-12).

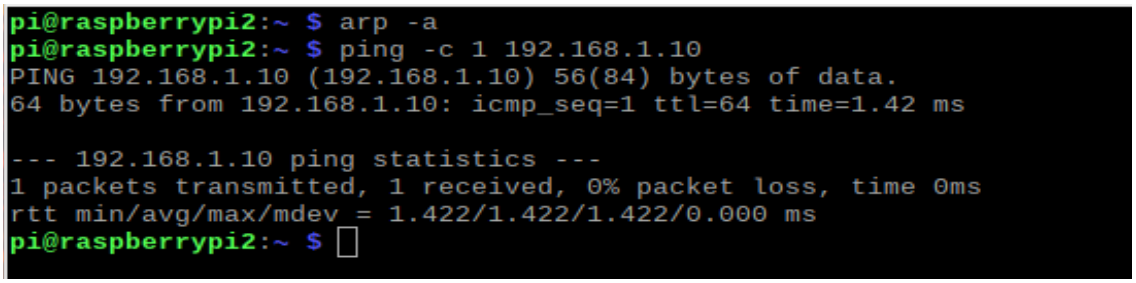


Figura 3-12 Comando ping a la interfaz de la Raspberrypi1 (Autoría propia)

Como la Raspberrypi2 hemos visto que tiene la tabla ARP vacía, no sabe la correspondencia de la IP 192.168.1.10 a la que hacemos el ping con la dirección MAC de la interfaz del dispositivo al que se dirige. Por lo que se ve obligada a ejecutar el protocolo ARP que averigüe la MAC asociada a la IP 192.168.1.10. Lo hace mediante difusión (*broadcast)* del mensaje ARP Request, la Raspberrypi1 contesta mediante un ARP Reply que esta IP está asignada a su interfaz Ethernet. Tras lo cual la Raspberrypi2 actualiza su tabla de correspondencia dirección IP-dirección MAC. Anteriormente, la Raspberrypi1 ha hecho lo mismo al recibir el ARP Request. Seguidamente la Raspberrypi2 ejecuta el comando ICMP Request (ping) y la Raspberrypi1 le responde con una ICMP Reply. (Figura 3-13).

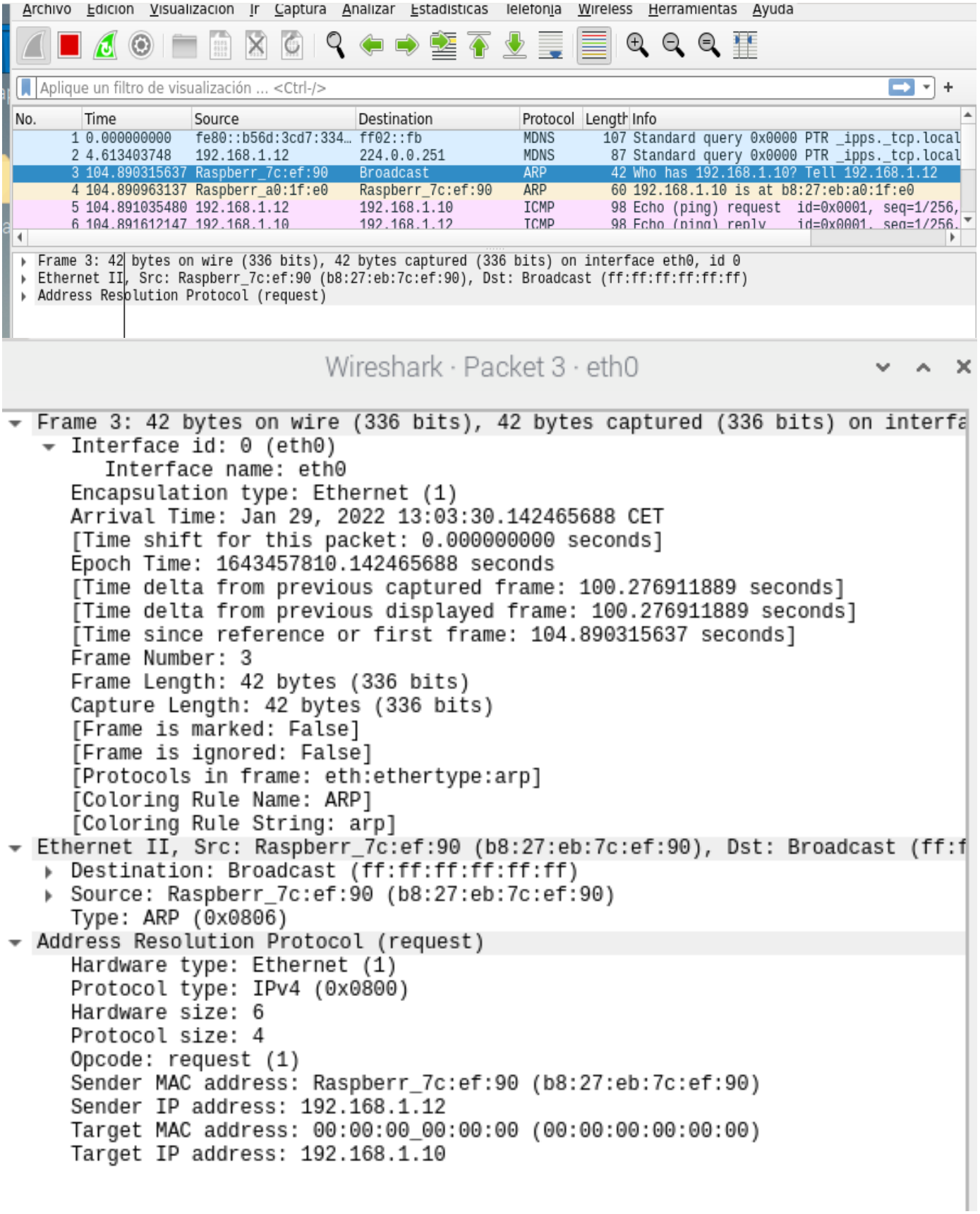


Figura 3-13 Captura mediante Wireshark del protocolo ARP (Autoría propia)

Podemos ver que la captura ocupa 42 bytes, aunque en realidad lo mínimo de la trama Ethernet son 64 bytes y esto es así porque el analizador de paquetes Wireshark automáticamente nos omite los bytes de relleno y el CRC. Especialmente importante es el *Ethertype*, del tipo 800 que nos indica que el *Payload* viene de la capa de red, pero en este caso el *Payload* es sólo de Ethernet y es del tipo 806 con lo que debe ser interpretado como ARP (Tabla 3-1).

Bytes 6 **Target MAC address**	**Bytes 6** **Sender MAC address**	**Bytes 2** **Ethertype**	**Bytes 28**	**Bytes 18**	**Bytes 4**
ff:ff:ff:ff:ff:ff	b8:27:eb:7c:ef:90	0x806	Payload	relleno	CRC

Tabla 3-1 Formato Ethernet ARP (Autoría propia)

La distribución de los 28 Bytes del Payload del ARP sería en general (Figura 3-14)

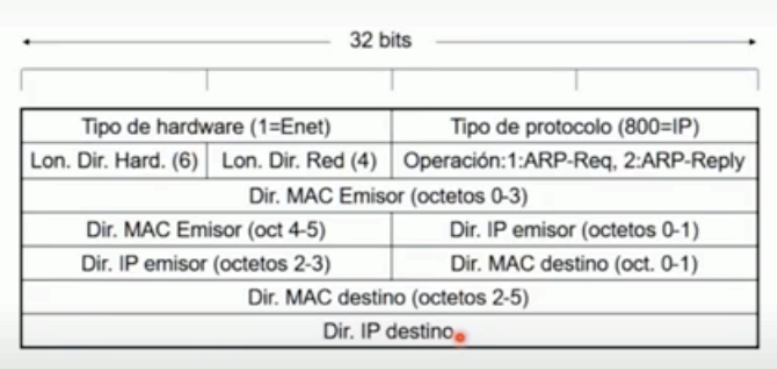


Figura 3-14 Protocolo ARP [25]

De forma similar en la captura de tráfico, podemos desplegar el mensaje *ARP Reply* (Figura 3-15).

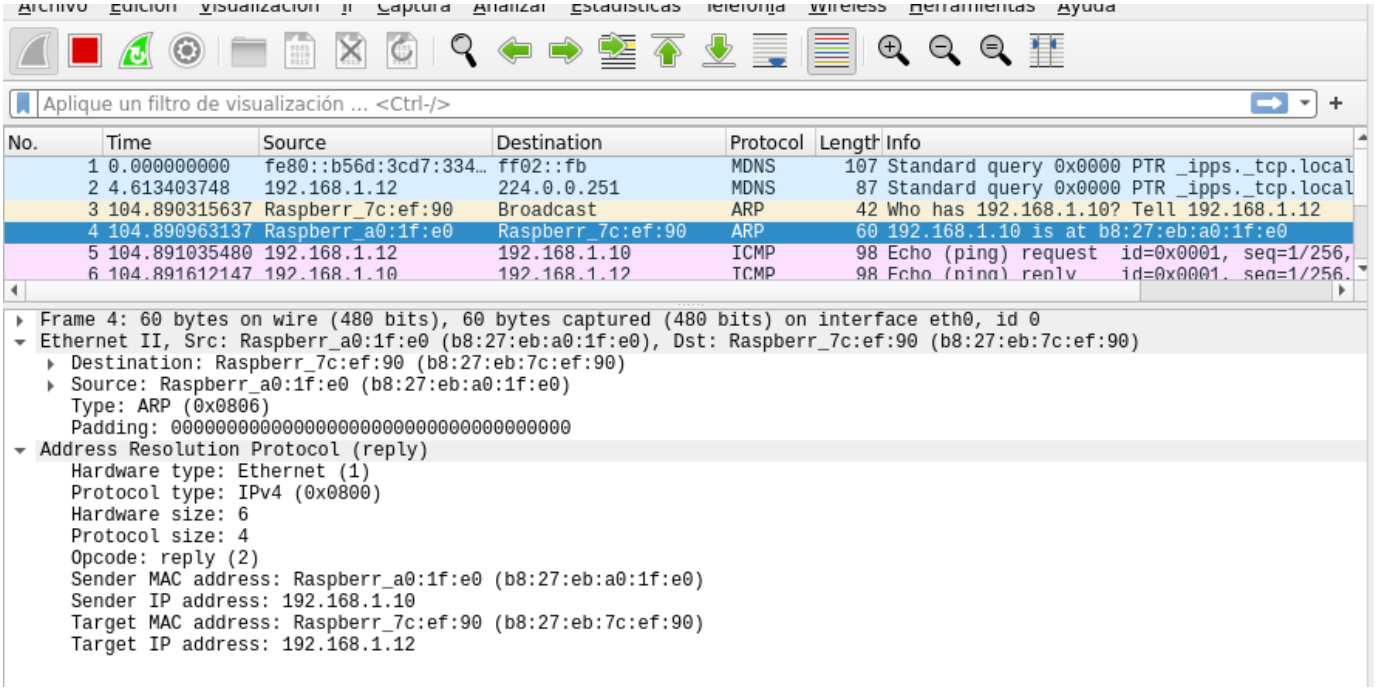


Figura 3-15 ARP Replay (Autoría propia)

A continuación, comprobamos en la Raspberrypi2 que su tabla caché de ARP ha sido actualizada, mediante el comando ARP -a. (Figura 3-16).

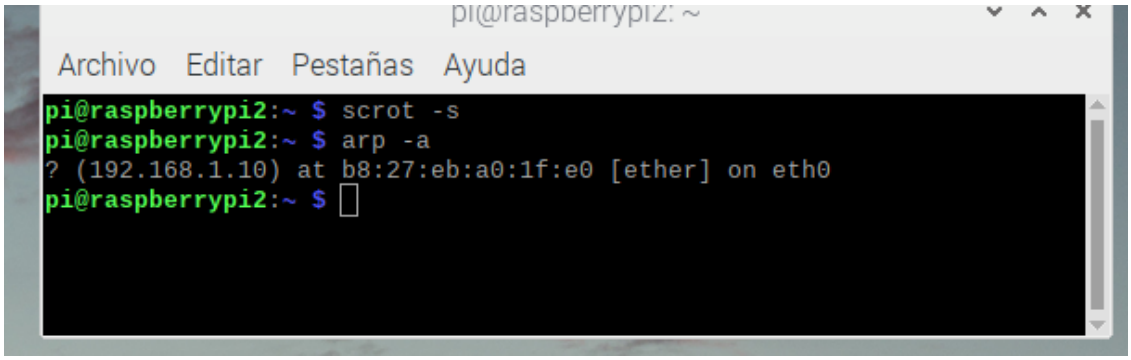


Figura 3-16 Actualización de la tabla ARP caché en la raspberrypi2 (Autoría propia)

Una vez ejecutado el protocolo ARP y con la tabla caché ARP rellenada en la Raspberrypi2 ya puede ejecutar el comando ping, el cual será devuelto con un ICMP *Reply* por la Raspberrypi1 porque también ella ha actualizado su ARP caché.

3.3 Maqueta 2: Protocolo DHCP

Dynamic Host Configuration Protocol (DHCP) es un protocolo de la capa de aplicación (operando sobre UDP en la capa de transporte) de tipo cliente/servidor mediante el cual un servidor DHCP asigna dinámicamente una dirección IP y otros parámetros de configuración de red a dispositivos de una red en situación de clientes. En nuestro caso configuraremos como servidor la interfaz de un rúter y como clientes 2 Raspberry Pi estando las tres interfaces conectadas a un conmutador (*switch*).

La maqueta (Figura 3-17 y Figura 3-18) constará de:

- 2 raspberry con el programa Wireshark instalado.
- 1 conmutador TP-link.
- 1 rúter MikroTik routerboard hAP series.
- 3 cables RJ45 Ethernet.

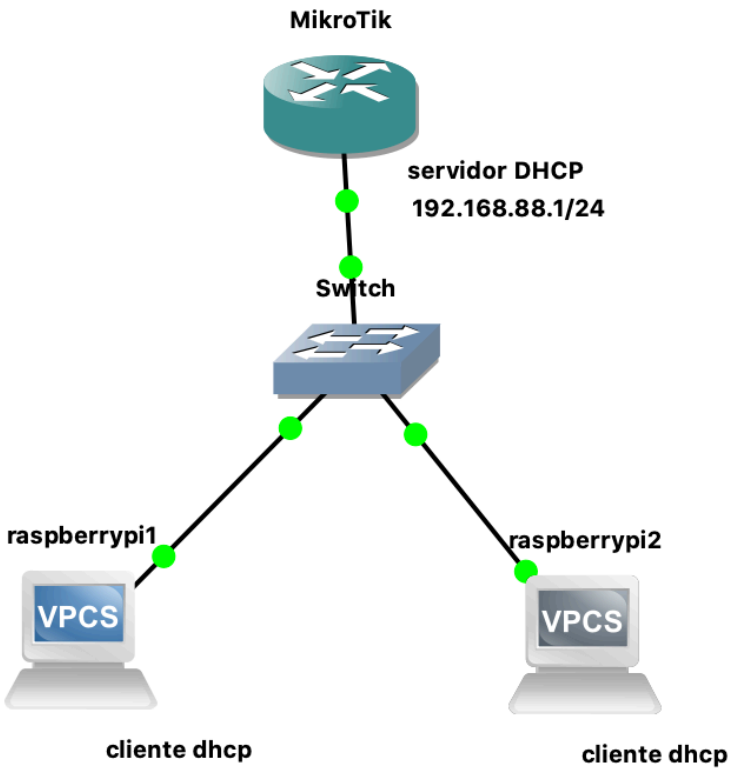


Figura 3-17 Maqueta a construir en el laboratorio para la comprobación del protocolo DCHP (Autoría propia)

Figura 3-18 Maqueta construida en el laboratorio para la comprobación del protocolo DHCP (Autoría propia)

3.3.1 Construcción de la maqueta

Conectamos con el cable Ethernet uno de los puertos Ethernet del rúter en nuestro caso el puerto 3 a un puerto cualquiera del *switch*. El único puerto Ethernet de la Raspberrypi1 lo conectamos a un puerto libre Ethernet del *switch* y lo mismo hacemos con la Raspberrypi2. Procedemos a encender las alimentaciones de todos los dispositivos.

Configuramos la interfaz Ethernet de las raspberry como clientes DHCP. Por defecto ya vienen configuradas como tales, pero como en el montaje del apartado 3.1 hemos añadido una configuración IP estática debemos eliminarla. Entramos con el comando *sudo nano /etc/dhcpcd.conf,* borramos lo añadido, lo guardamos y reiniciamos con el comando *sudo reboot*. La configuración de la interfaz del rúter ya viene determinada por una configuración por defecto de fábrica en modo puente entre las interfaces ether2, ether3, ether4 y ether5 con la IP 192.168.88.1/24, pero debemos de activar su DHCP. Para ello desde cualquier raspberry ponemos en su navegador la IP 192.168.88.1 y nos sale la página de configuración del rúter mostrada en la Figura 3-19 en la cual activamos el DHCP Server. En la Figura 3-20 se muestra el esquema de funcionamiento del DHCP, indicando el intercambio de paquetes que abordaremos con más detalle en la próxima sección.

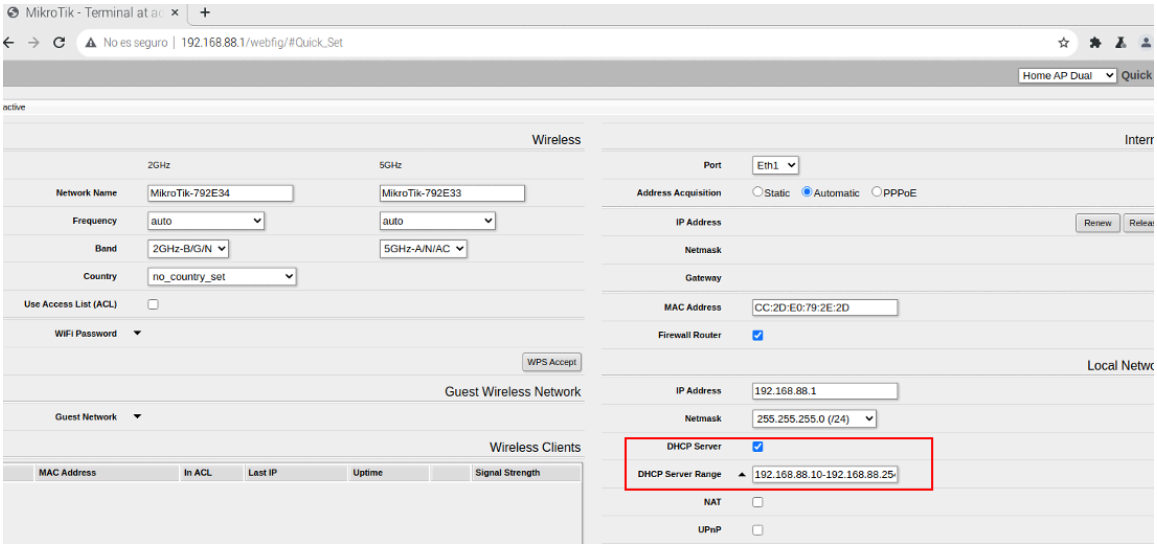


Figura 3-19 Webfig del rúter (Autoría propia)

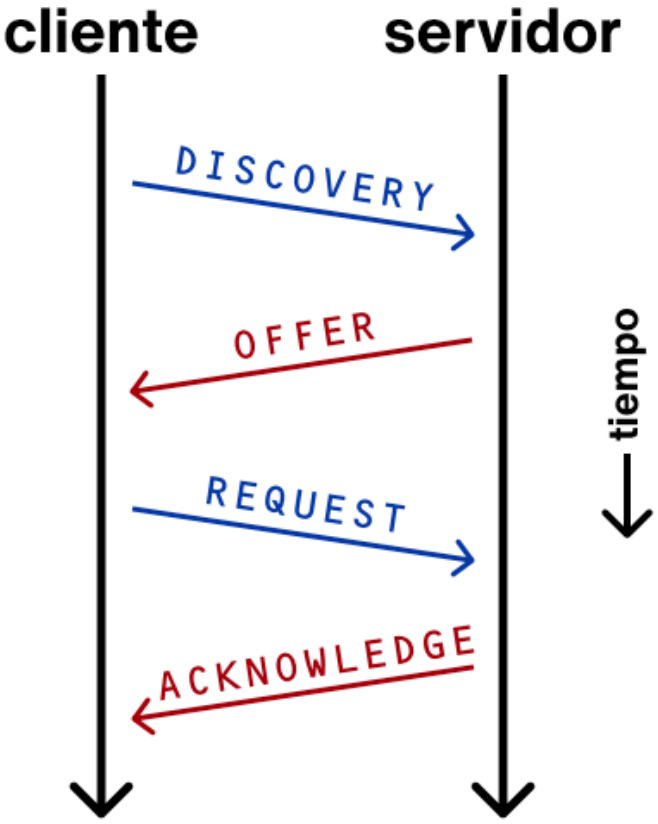


Figura 3-20 Resumen de la petición de un servicio DHCP [48]

3.3.2 Comprobación experimental del protocolo DHCP

Se ha realizado una captura con el programa Wireshark para explicar el funcionamiento del protocolo DHCP cuando un terminal en situación de cliente solicita los servicios de un servidor DHCP.

Cada dispositivo conectado a una red IP necesita una configuración IP (dirección IP, máscara de subred, rúter por defecto- pasarela o Gateway-, servidor DNS…). Aunque se puede utilizar una configuración manual en redes pequeñas o poco cambiantes y en equipos perfectamente identificados dentro de la red (servidores o equipos de interconexión), nosotros vamos a utilizar mecanismos de asignación dinámica (BOOTP, DHCP…) y en particular estudiaremos el DHCP porque es la alternativa más avanzada a otros protocolos de gestión de direcciones IP (como BOOTP o *Bootstrap Protocol*). DHCP funciona en modo cliente-servidor y está descrito en la RFC 2131 [44]. El funcionamiento de DHCP sería:

- DHCP DISCOVER
  El cliente DHCP envía un mensaje DHCP Discover en modo *broadcast* con lo cual lo reciben todos los equipos de la red local, incluidos los que actúan como servidores DHCP. El paquete está en la Figura 3-21.

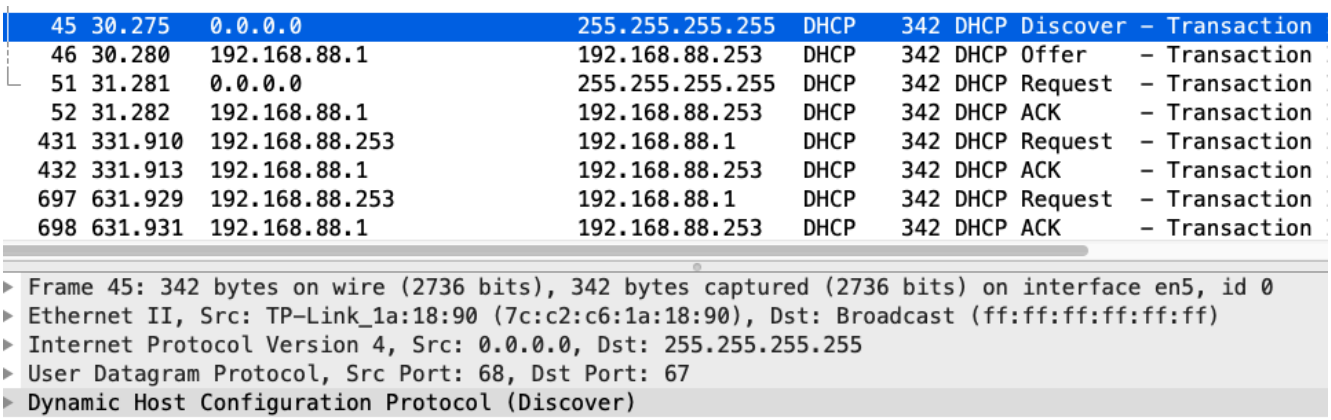


Figura 3-21 DHCP DISCOVER (Autoría propia)

- DHCP OFFER
  Los servidores, en base a los parámetros que le envía el cliente, que suele ser su dirección MAC, le hacen una propuesta. Los servidores DHCP determinan una posible configuración para el cliente. Esta configuración se la envían los servidores al cliente en modo *unicast* dirigido solamente al cliente, que es el mensaje DHCP OFFER. Pueden enviar una propuesta cada uno de los servidores DHCP que hay en la red, en caso de haber varios. El paquete está en la Figura 3-22.

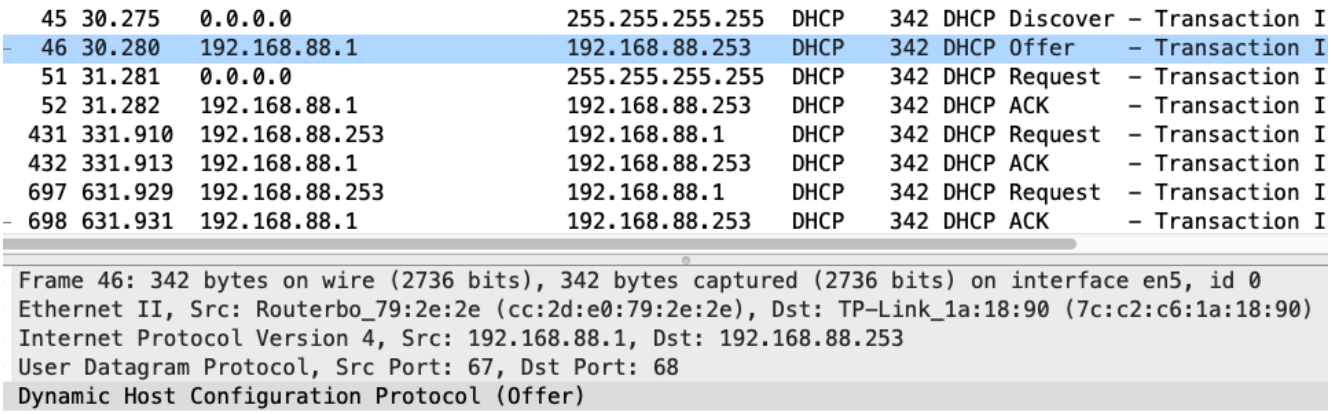


Figura 3-22 DHCP OFFER (Autoría propia)

- DHCP REQUEST

  El cliente procesa las ofertas de configuración de los servidores y selecciona una de ellas. Contestando que acepta esta configuración en modo *broadcast* para que lo reciban todos los servidores. De tal forma que todos los servidores se dan por enterados cual es la oferta aceptada por el cliente. (Figura 3-23).

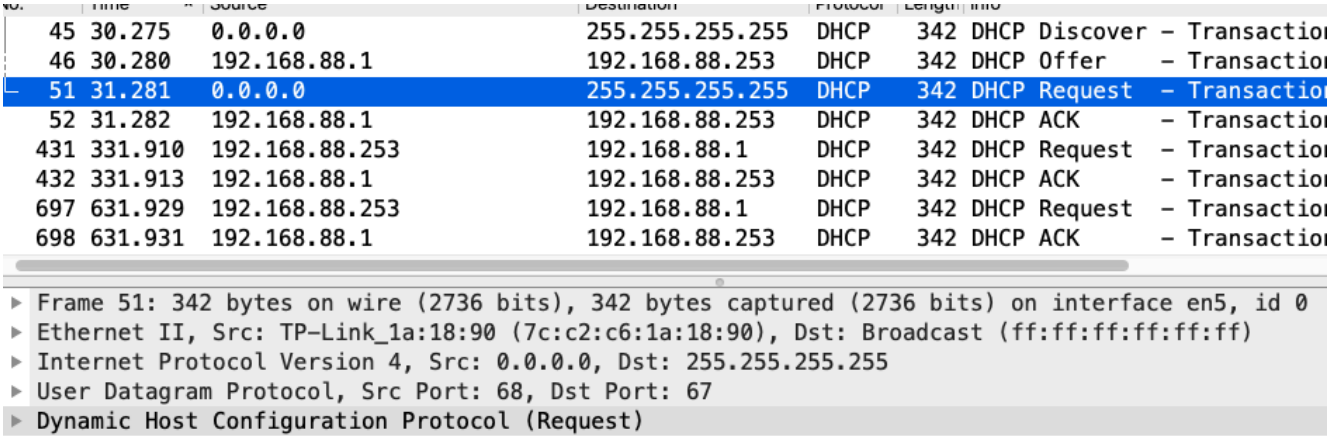
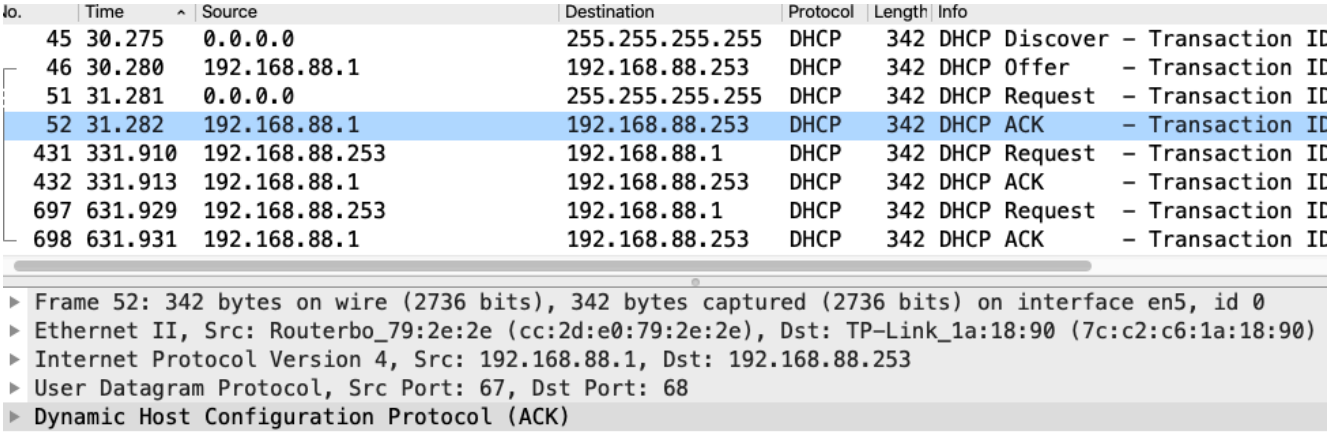


Figura 3-23 DHCP BROADCAST (Autoría propia)

- DHCP ACK

  El servidor seleccionado contestará en modo *unicast* con un mensaje DHCP ACK, reconociendo la oferta e indicando al cliente que ya puede empezar a utilizar la configuración indicada.

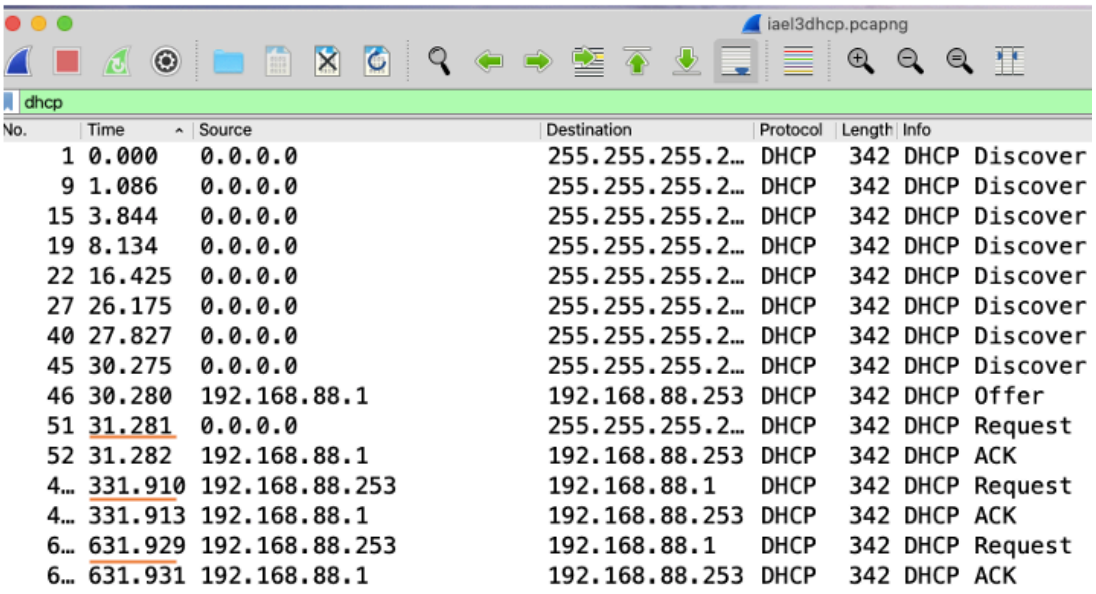


No.	Time	Source	Destination	Protocol	Length	Info
45	30.275	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID
46	30.280	192.168.88.1	192.168.88.253	DHCP	342	DHCP Offer - Transaction ID
51	31.281	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID
52	31.282	192.168.88.1	192.168.88.253	DHCP	342	DHCP ACK - Transaction ID
431	331.910	192.168.88.253	192.168.88.1	DHCP	342	DHCP Request - Transaction ID
432	331.913	192.168.88.1	192.168.88.253	DHCP	342	DHCP ACK - Transaction ID
697	631.929	192.168.88.253	192.168.88.1	DHCP	342	DHCP Request - Transaction ID
698	631.931	192.168.88.1	192.168.88.253	DHCP	342	DHCP ACK - Transaction ID

```
▸ Frame 52: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface en5, id 0
▸ Ethernet II, Src: Routerbo_79:2e:2e (cc:2d:e0:79:2e:2e), Dst: TP-Link_1a:18:90 (7c:c2:c6:1a:18:90)
▸ Internet Protocol Version 4, Src: 192.168.88.1, Dst: 192.168.88.253
▸ User Datagram Protocol, Src Port: 67, Dst Port: 68
▸ Dynamic Host Configuration Protocol (ACK)
```

Figura 3-24 DHCP ACK (Autoría propia)

Se muestra a continuación todas las capturas DHCP que corresponden a un tiempo de aproximadamente 11 min. (Figura 3-25). Vemos que en los 331 segundos se ha tenido que solicitar una renovación al servidor y otra a los 631 segundos, es decir cada 300 segundos (parámetro configurable). El diagrama de estados por los que pasa un cliente DHCP se representa en la Figura 3-26.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000	0.0.0.0	255.255.255.2…	DHCP	342	DHCP Discover
9	1.086	0.0.0.0	255.255.255.2…	DHCP	342	DHCP Discover
15	3.844	0.0.0.0	255.255.255.2…	DHCP	342	DHCP Discover
19	8.134	0.0.0.0	255.255.255.2…	DHCP	342	DHCP Discover
22	16.425	0.0.0.0	255.255.255.2…	DHCP	342	DHCP Discover
27	26.175	0.0.0.0	255.255.255.2…	DHCP	342	DHCP Discover
40	27.827	0.0.0.0	255.255.255.2…	DHCP	342	DHCP Discover
45	30.275	0.0.0.0	255.255.255.2…	DHCP	342	DHCP Discover
46	30.280	192.168.88.1	192.168.88.253	DHCP	342	DHCP Offer
51	31.281	0.0.0.0	255.255.255.2…	DHCP	342	DHCP Request
52	31.282	192.168.88.1	192.168.88.253	DHCP	342	DHCP ACK
4…	331.910	192.168.88.253	192.168.88.1	DHCP	342	DHCP Request
4…	331.913	192.168.88.1	192.168.88.253	DHCP	342	DHCP ACK
6…	631.929	192.168.88.253	192.168.88.1	DHCP	342	DHCP Request
6…	631.931	192.168.88.1	192.168.88.253	DHCP	342	DHCP ACK

Figura 3-25 DHCP (Autoría propia)

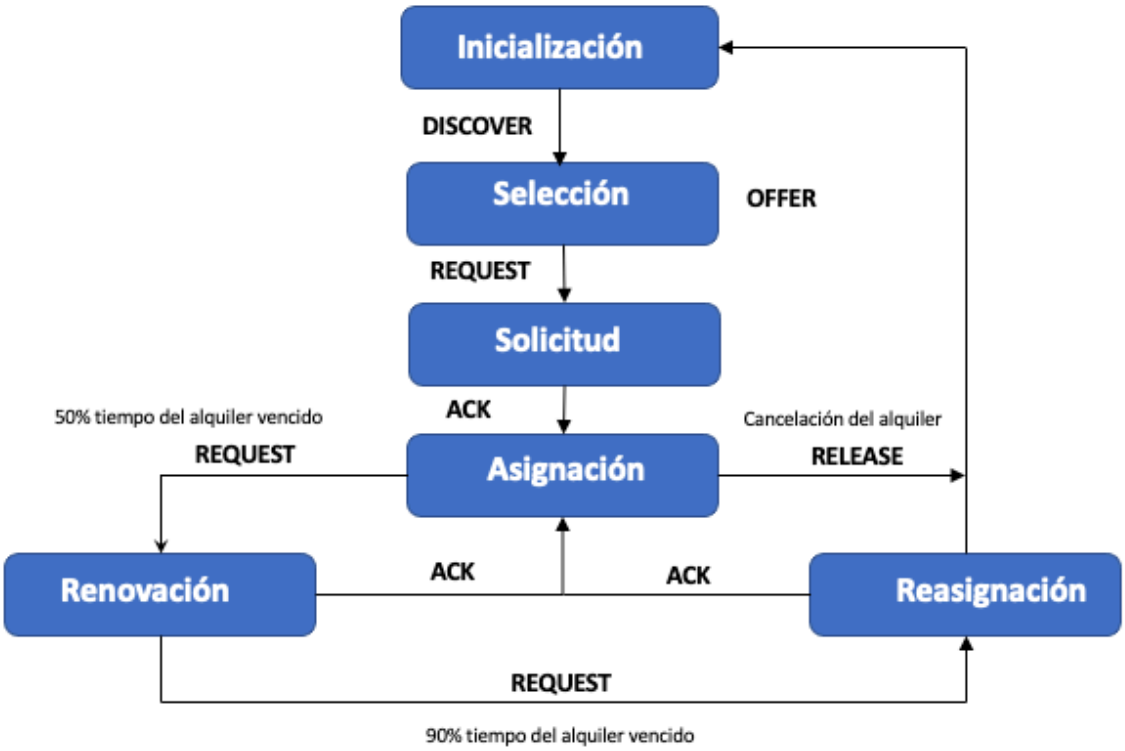


Figura 3-26 Diagrama de estados de un cliente DHCP (Autoría propia).

3.4 Maqueta 3: Protocolo HTTP

El protocolo HTTP (*Hypertext Transfer Protocol*) se ubica en la capa de aplicación y utiliza TCP (orientado a conexión y fiable) sobre la capa de transporte. Es un protocolo sin conexión y sin estado. En el HTTP 1.0 después de que el servidor haya respondido la petición del cliente, se rompe la conexión entre ambos, pero desde el HTTP 1.1 estandarizado con RFC 2616 [45] se puede mantener abierta la conexión, y por defecto no se guarda memoria del contexto de la conexión en las siguientes peticiones. HTTP utiliza tipos MIME (*Multipurpose Internet Mail Extension*) para la determinación del tipo de datos que transporta.

HTTP se basa en la estructura cliente-servidor intercambiándose datos como documentos HTML o texto plano a los cuales se puede añadir video (mp4…), imágenes (jpg…) y todo conformado con lo que se llama *layout CSS* (*Cascading Style Sheets)* que nos indicará la distribución de los diferentes elementos en la página. Utiliza *Uniform Resource Locato*r (URL) para localizar recursos, la forma general de una URL es *servicio://host/fichero.ext*. HTTP tiene diferentes mensajes de petición del recurso *HTTP Request Message*, así por ejemplo el método GET se usa para preguntar al servidor por un documento a través de su URL.

La maqueta a desarrollar (Figura 3-27 y Figura 3-28) constará de:

- 2 raspberry con los programas instalados wireshark (*sniffer*) y nginx (servidor web).
- 2 routers MikroTik routerboard hAP series.
- 3 cables RJ45 Ethernet.

Con ella además de probar el funcionamiento del protocolo HTTP estudiaremos la configuración estática de rutas en los equipos MikroTik.

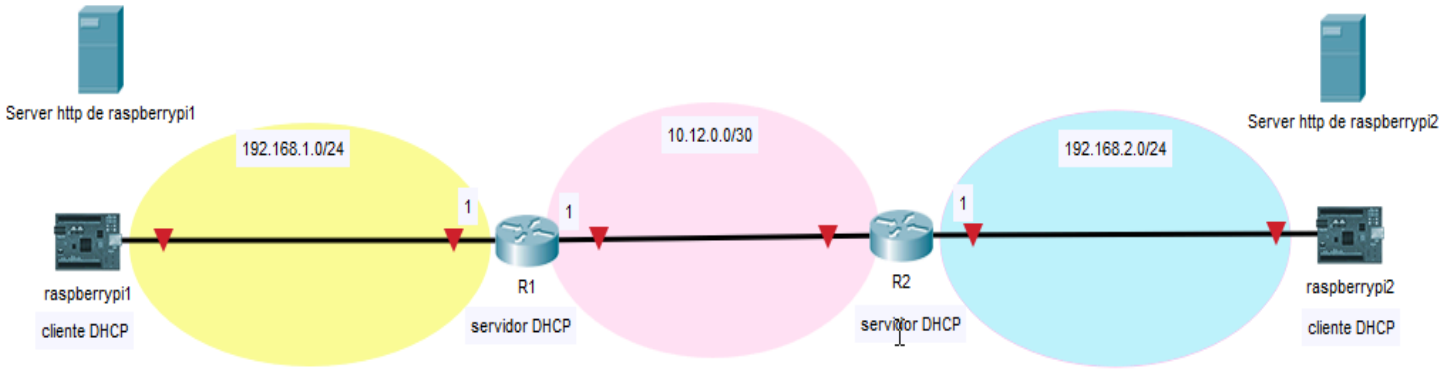


Figura 3-27 Maqueta a construir en el laboratorio para la comprobación experimental del protocolo HTTP y enrutamiento estático (Autoría propia)

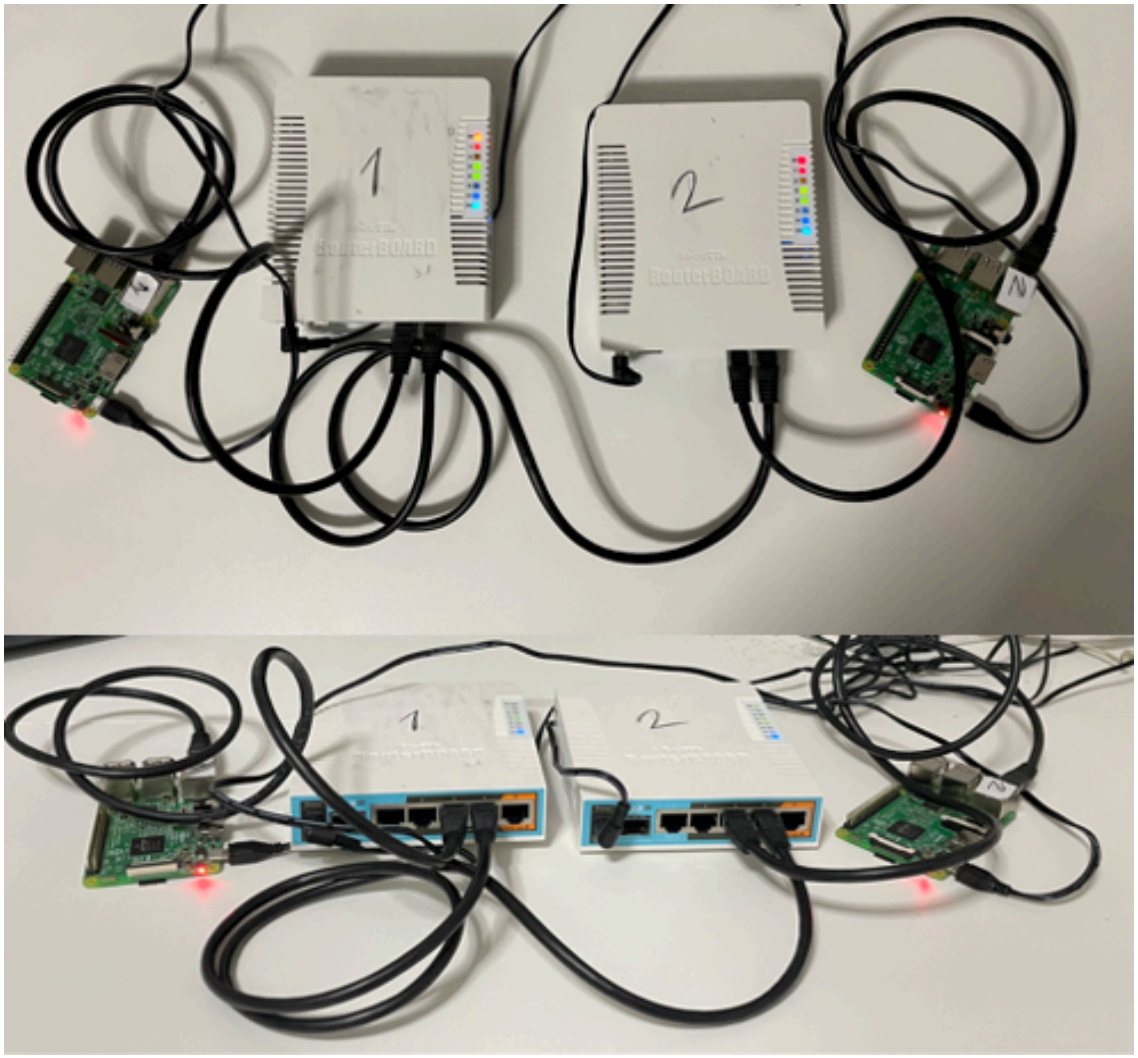


Figura 3-28 Maqueta construida en el laboratorio para la comprobación experimental del protocolo HTTP y configuración estática de enrutamiento (Autoría propia)

3.4.1 Construcción de la maqueta

El routerboard MikroTik hAP series lleva una configuración por defecto, los puertos Ethernet del 2-5 junto con las interfaces WLAN y el puerto SPF vienen configuradas en modo bridge (*type bridge*) con el nombre *bridge*, lo cual quiere decir que se comporta como un *switch*. Se accede a la configuración a través de uno de los anteriores puertos escribiendo la IP 192.168.88.1 en el navegador web. Además, el puerto 1 viene configurado por defecto como cliente DHCP con el *firewall* y NAT instalado, permitiendo así una conexión a Internet desde el puerto ether1 al ISP (*Internet Service Provider*).

A continuación, liberamos los puertos (sacándolos del *bridge* configurado por defecto) ether3 y ether4 para poder configurarlos en nuestra distribución. Es suficiente que el *master port* tenga el calificativo de *none* para los puertos 3 y 4 (con el comando */interface ethernet set ether3 master-port=none,* y equivalentemente para ether4) después se ha configurado una interfaz de tipo bridge con el nombre LocalRL1 para el primer router R1 y LocalRL2 para el otro. En ambos se han añadido el puerto ether3 quedando, así como si fuera un *switch* y así podremos conectar las raspberrys directamente al puerto ether3 sin necesidad de *switch*. En el puerto 4 Ethernet conectaremos los routers entre sí.

La configuración en el rúter R1, incluida la ruta estática, es la mostrada en la Figura 3-29, en la que se puede ver que se ha utilizado la consola de comandos para aplicarla.

```
[admin@R1 MikroTik] > interface bridge add name=LocalRL1
[admin@R1 MikroTik] > interface bridge port add interface=ether3 bridge=LocalRL1
[admin@R1 MikroTik] > ip address add address=192.168.1.1/24 interface=LocalRL1
[admin@R1 MikroTik] > ip dhcp-server setup
Select interface to run DHCP server on

dhcp server interface: LocalRL1
Select network for DHCP addresses

dhcp address space: 192.168.1.0/24
Select gateway for given network

gateway for dhcp network: 192.168.1.1
Select pool of ip addresses given out by DHCP server

addresses to give out: 192.168.1.2-192.168.1.4
Select DNS servers

dns servers: 8.8.8.8
Select lease time

lease time: 10m
[admin@R1 MikroTik] >
[admin@R1 MikroTik] >
[admin@R1 MikroTik] >
[admin@R1 MikroTik] >
[admin@R1 MikroTik] >
[admin@R1 MikroTik] >
[admin@R1 MikroTik] >
[admin@R1 MikroTik] >
[admin@R1 MikroTik] >
[admin@R1 MikroTik] >
[admin@R1 MikroTik] >
[admin@R1 MikroTik] > ip address add address=10.12.0.1/30 interface=ether4
[admin@R1 MikroTik] > ip route add distance=1 dst-address=192.168.2.0/24 gateway=10.12.0.2
```

Figura 3-29 Configuración del rúter R1 (Autoría propia)

El estado de los puertos en el rúter R1 con sus interfaces en modo *bridge* detalladas se puede observar en la Figura 3-30 y 3-31.

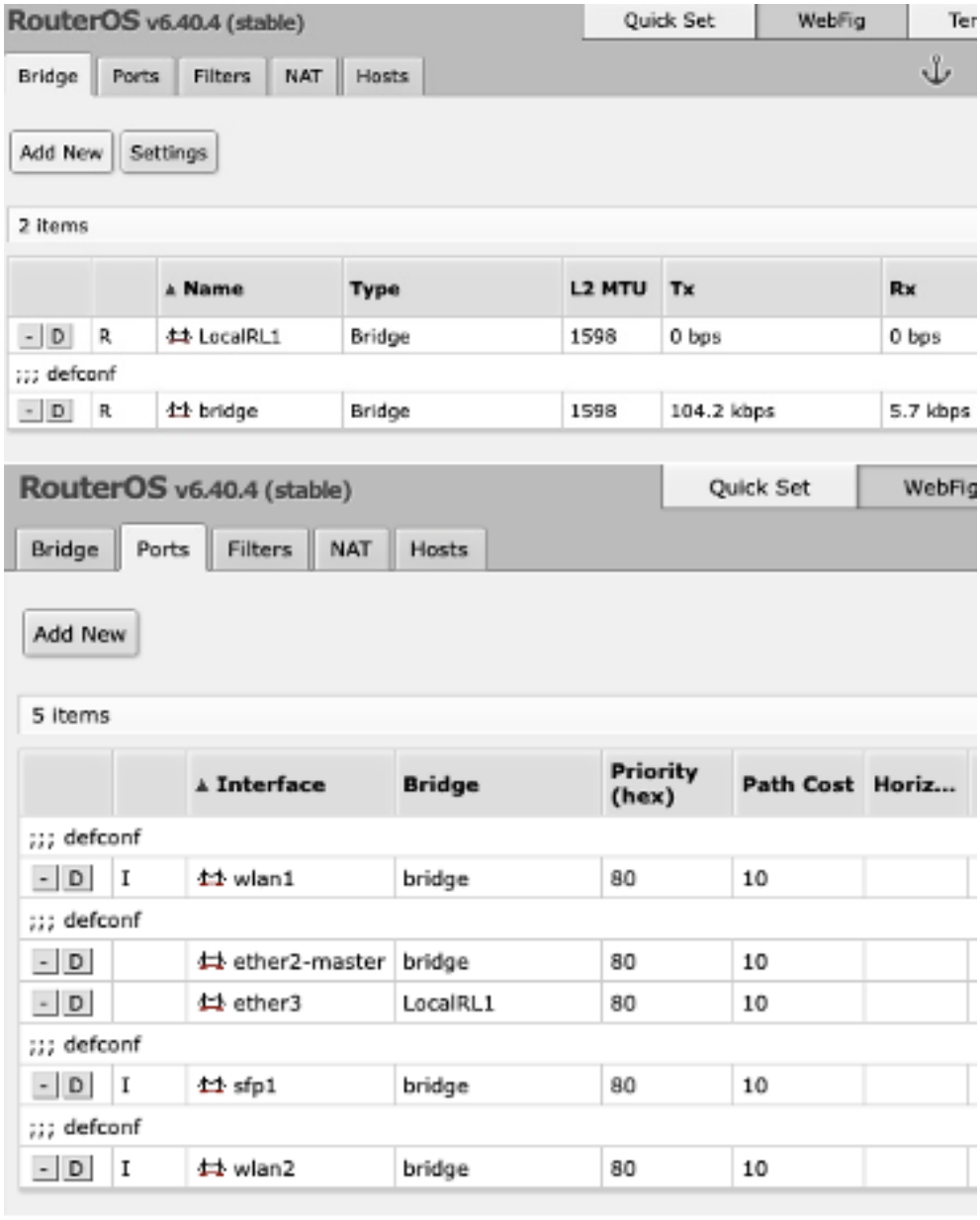


Figura 3-30 Distribución de los puertos bridge del rúter R1 (Autoría propia)

		Name	Type	Actual MTU	L2 MTU	Tx	Rx
- D	R	LocalRL1	Bridge	1500	1598	0 bps	0 bps
;;; defconf							
- D	R	bridge	Bridge	1500	1598	104.1 kbps	5.7 kbps
D		ether1	Ethernet	1500	1598	0 bps	0 bps
D	RS	ether2-master	Ethernet	1500	1598	104.5 kbps	9.2 kbps
D	RS	ether3	Ethernet	1500	1598	512 bps	0 bps
D	R	ether4	Ethernet	1500	1598	0 bps	0 bps
D		ether5	Ethernet	1500	1598	0 bps	0 bps
D	S	sfp1	Ethernet	1500	1600	0 bps	0 bps
D	S	wlan1	Wireless (Atheros AR9	1500	1600	0 bps	0 bps
D	S	wlan2	Wireless (Atheros AR9	1500	1600	0 bps	0 bps

Figura 3-31 Distribución de los puertos del rúter R1 (Autoría propia)

La asignación de direcciones y tabla de rutas del rúter R1 se muestra en la Figura 3-32.

```
admin@R1 MikroTik] /ip address> print
Flags: X - disabled, I - invalid, D - dynamic
#   ADDRESS            NETWORK         INTERFACE
0   ;;; defconf
    192.168.88.1/24    192.168.88.0    ether2-master
1   192.168.1.1/24     192.168.1.0     LocalRL1
2   10.12.0.1/30       10.12.0.0       ether4
admin@R1 MikroTik] /ip address> ..
admin@R1 MikroTik] /ip> route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
#      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
0 ADC  10.12.0.0/30       10.12.0.1       ether4                    0
1 ADC  192.168.1.0/24     192.168.1.1     LocalRL1                  0
2 A S  192.168.2.0/24                     10.12.0.2                 1
3 ADC  192.168.88.0/24    192.168.88.1    bridge                    0
admin@R1 MikroTik] /ip>
```

Figura 3-32 Asignación de direcciones y tabla de rutas del rúter R1 (Autoría propia)

Vemos que para acceder a la red 192.168.2.0/24 a través del rúter R1 debe hacerse con la IP 10.12.0.2 que se asignará al rúter R2 de forma estática.

Terminamos instalando un paquete de software en las Raspberrys para que pueda utilizarse como un servidor web con el comando desde la terminal: *sudo apt-get install nginx.* A continuación, procedemos a configurar de manera similar el rúter R2, quedando la asignación de direcciones (*address*) y la tabla de rutas como se muestra en la Figura 3-33.

```
[admin@RL2] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   ;;; defconf
     192.168.88.1/24    192.168.88.0    bridge
 1   192.168.2.1/24     192.168.2.0     LocalRL2
 2   10.12.0.2/30       10.12.0.0       ether4
[admin@RL2] > ip route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 ADC  10.12.0.0/30       10.12.0.2       ether4                    0
 1 A S  192.168.1.0/24                     10.12.0.1                 1
 2 ADC  192.168.2.0/24     192.168.2.1     LocalRL2                  0
 3 ADC  192.168.88.0/24    192.168.88.1    bridge                    0
[admin@RL2] >
```

Figura 3-33 Asignación de direcciones y tabla de rutas del rúter R2 (Autoría propia)

3.4.2 Comprobación experimental del protocolo HTTP

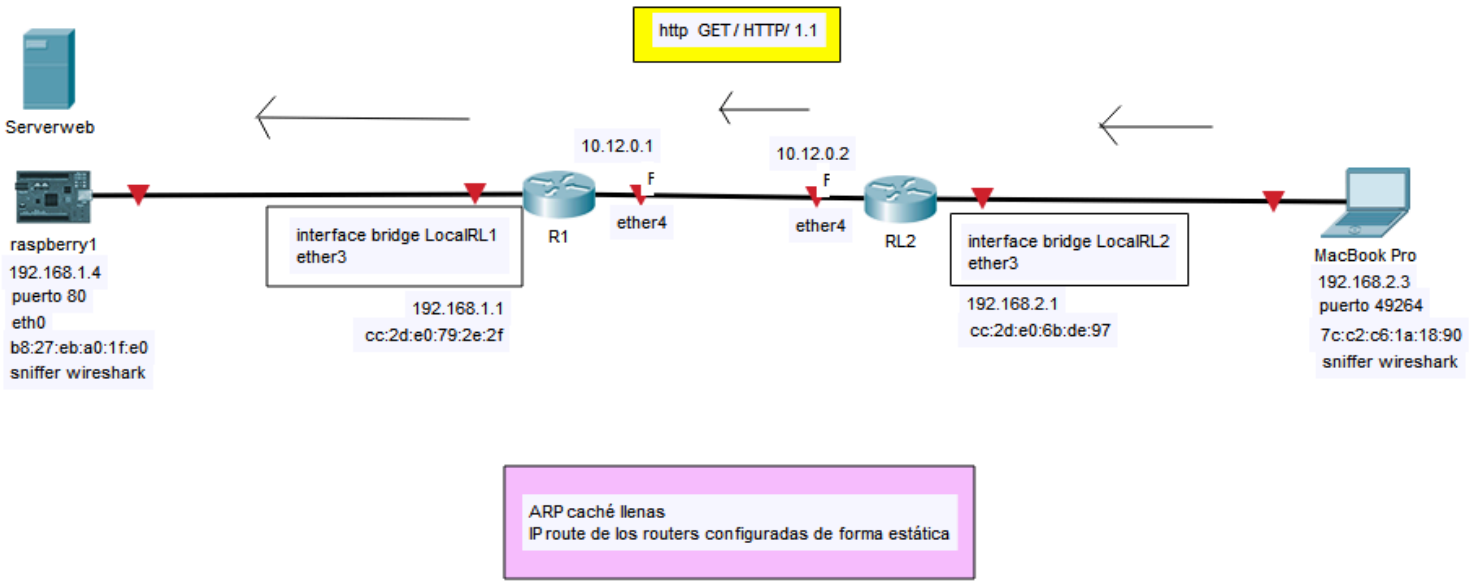


Figura 3-34 Petición de una página web del portátil al servidor web de la raspberry1 (Autoría Propia)

Se accederá a la Raspberrypi1 como servidor web desde un navegador instalado en la otra Raspberrypi2 o, en su lugar, en un portátil con el sistema operativo macOS. Para ello, averiguaremos las IPs de la Raspberrypi1 y del portátil, comprobando así el funcionamiento experimental del protocolo DHCP. Vemos que son las que se indican en la Figura 3-35.

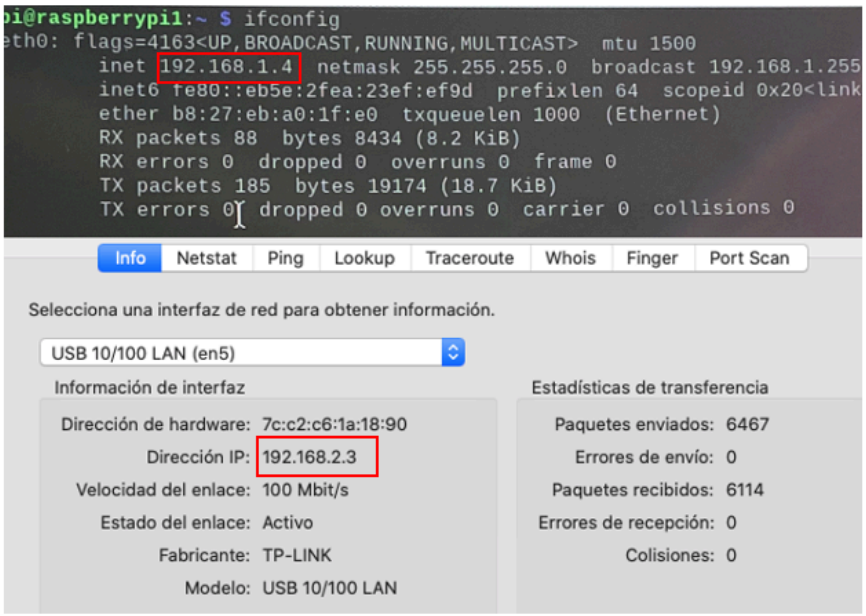


Figura 3-35 IPs asignadas a la raspberrypi1 y al portátil (Autoría propia)

El proceso de petición de una página web al servidor Raspberry1 consta de las siguientes partes:

1. Establecimiento de la conexión

    - Tenemos las tablas caché ARP rellenadas en la Raspberrypi1 y en el portátil MacBook Pro. Las rutas IP están configuradas en los rúteres de forma estática salvo las directamente conectadas.
    - Abrimos Wireshark en la Raspberrypi1 y en el MacbookPro configurando en ambos el filtro http.
    - Abrimos el navegador en el MacbookPro e introducimos en su barra de direcciones la IP 192.168.1.4 como podemos ver en la Figura 3-36. En realidad, lo que debemos introducir es la *Uniform Resource Locato*r (URL) que en este caso sería *servicio://host/fichero.ext*. *http://192.168.1.4/,* no habiendo rellenado el campo *fichero.ext* porque solamente tiene una página el servidor nginx (página por defecto que se crea al instalarlo).
    - Inicia el MacbookPro el establecimiento de la conexión con la Raspberry1. Podemos ver en la Figura 3-37 que estos son los paquetes 293-294-295. Los paquetes 293-294 son del tipo TCP-SYN que pertenecen propiamente al establecimiento sincronizado de la conexión. El último paquete 295 es del tipo TCP-ACK con lo que la conexión está ya establecida.
    - Observamos que el puerto de la Raspberry1 es el 80 que generalmente es el que está en modo *listen* en un servidor web (aunque no tiene que ser siempre así). El puerto del portátil es el 49264 y es el que ha tenido a bien asignarnos el sistema operativo macOS Catalina.

2. Petición de la página web

    - Ahora corresponde la comunicación entre cliente y servidor mediante el protocolo HTTP (véase paquete 296 en la Figura 3-37). El cliente solicita el documento al servidor y lo hace con GET / HTTP/1.1. El método GET indica el fichero que el cliente solicita y la versión de HTTP.
    - El portátil recibe a continuación un TCP-ACK del servidor confirmando que ha recibido la petición. Paquete 297 de la Figura 3-37.

3. Envío de la página web

    - El servidor responde mandando la respuesta HTTP y aunque debería de mandar el código de estado 200 para indicar que la petición del cliente ha sido procesada satisfactoriamente, el código de estado es el 304. El 304 es un código que se produce cuando se ha accedido a un recurso con anterioridad y éste se encuentra almacenado en la caché del navegador, con esto el servidor nos indica que el recurso no se ha visto modificado respecto al que tiene el cliente y que no se envía. (Paquete 298 de la Figura 3-37).
    - El cliente le envía al servidor un TCP-ACK para confirmar que ha recibido la respuesta del servidor. Paquete 299 de la Figura 3-37.

Figura 3-36 Acceso al servidor web de la aspberrypi1 (Autoría propia)

```
293 81.187816 192.168.2.3    192.168.1.4    TCP     78 49264 → 80 [SYN, ECN, CWR] Seq=0 Win=
294 81.189160 192.168.1.4    192.168.2.3    TCP     74 80 → 49264 [SYN, ACK, ECN] Seq=0 Ack=
295 81.189406 192.168.2.3    192.168.1.4    TCP     66 49264 → 80 [ACK] Seq=1 Ack=1 Win=131
296 81.409671 192.168.2.3    192.168.1.4    HTTP   502 GET / HTTP/1.1
297 81.410899 192.168.1.4    192.168.2.3    TCP     66 80 → 49264 [ACK] Seq=1 Ack=437 Win=64
298 81.413775 192.168.1.4    192.168.2.3    HTTP   246 HTTP/1.1 304 Not Modified
299 81.414027 192.168.2.3    192.168.1.4    TCP     66 49264 → 80 [ACK] Seq=437 Ack=181 Win=
```

Figura 3-37 Establecimiento de la conexión TCP, petición y entrega del recurso HTTP (Autoría propia).

Se muestra en las Figuras 3-38 en detalle el contenido de los paquetes con el filtro http en el programa Wireshark.

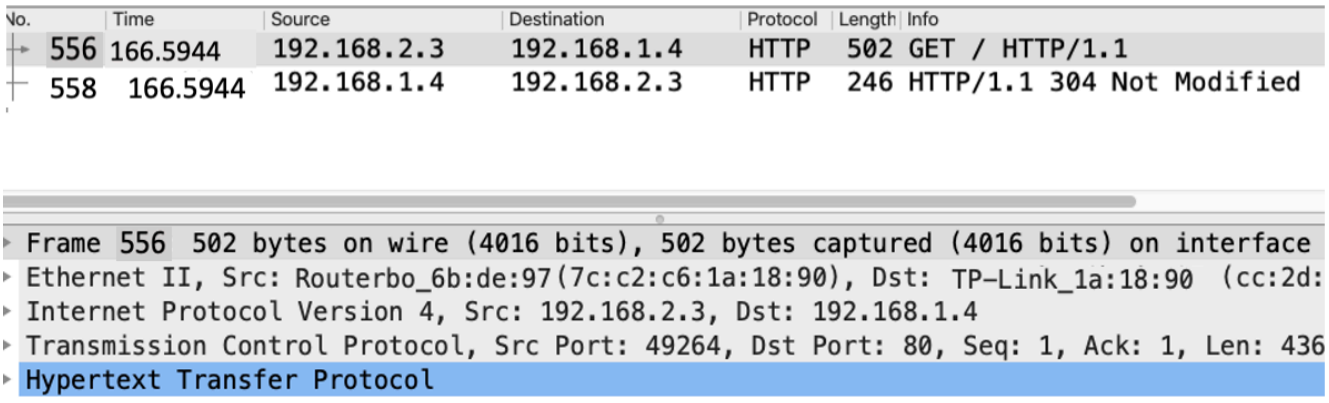


Figura 3-38 Wireshark Raspberrypi1 servidor web (Autoría propia)

3.5 Maqueta 4: Protocolo 802.1q en VLANs

El protocolo 802.1q estandarizado en el RFC 2674 [46] es un mecanismo que permite a múltiples redes compartir de forma transparente el mismo medio y sin problemas de interferencia. Está sobre el nivel de la capa de enlace y se diferencia sólo en cuatro bytes añadidos después del Source MAC. Los dos primeros indican en su Ethertype el valor hexadecimal 0x8100 que es un protocolo VLAN y los dos restantes bytes se interpretan en la forma: 3 bits de prioridad, 1 bit de indicador de forma canónica y 12 bits de identificador VLAN. Como se puede ver en la Figura 3-39 se añade un etiquetado, no encapsula por lo tanto la trama original.

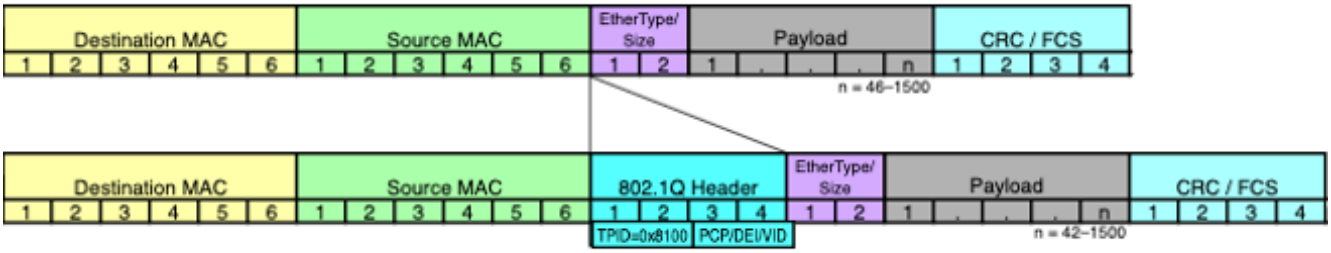


Figura 3-39 Etiquetado 802.1q [25]

Las VLANs limitan el tráfico *broadcast* en las redes locales y así evitan un elevado procesamiento en la CPU. Además, la división de una LAN en otras más pequeñas mejora la seguridad de todos los usuarios.

La maqueta a implementar en este caso (Figura 3-40 y Figura 3-41) [1] constará de:

- 2 raspberry con los programas instalados Wireshark (*sniffer*) y nginx (servidor web).
- 2 rúteres MikroTik routerboard hAP series.
- 3 cables RJ45 Ethernet.

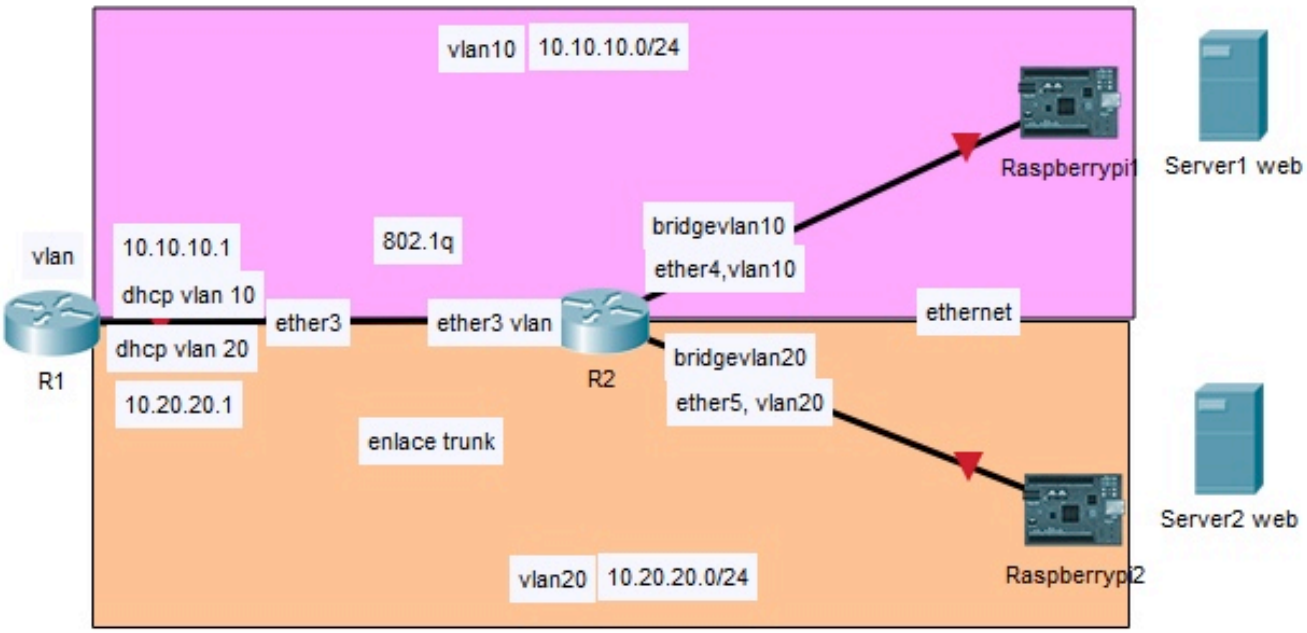


Figura 3-40 Maqueta a construir en el laboratorio para el protocolo 802.1q (Autoría propia)

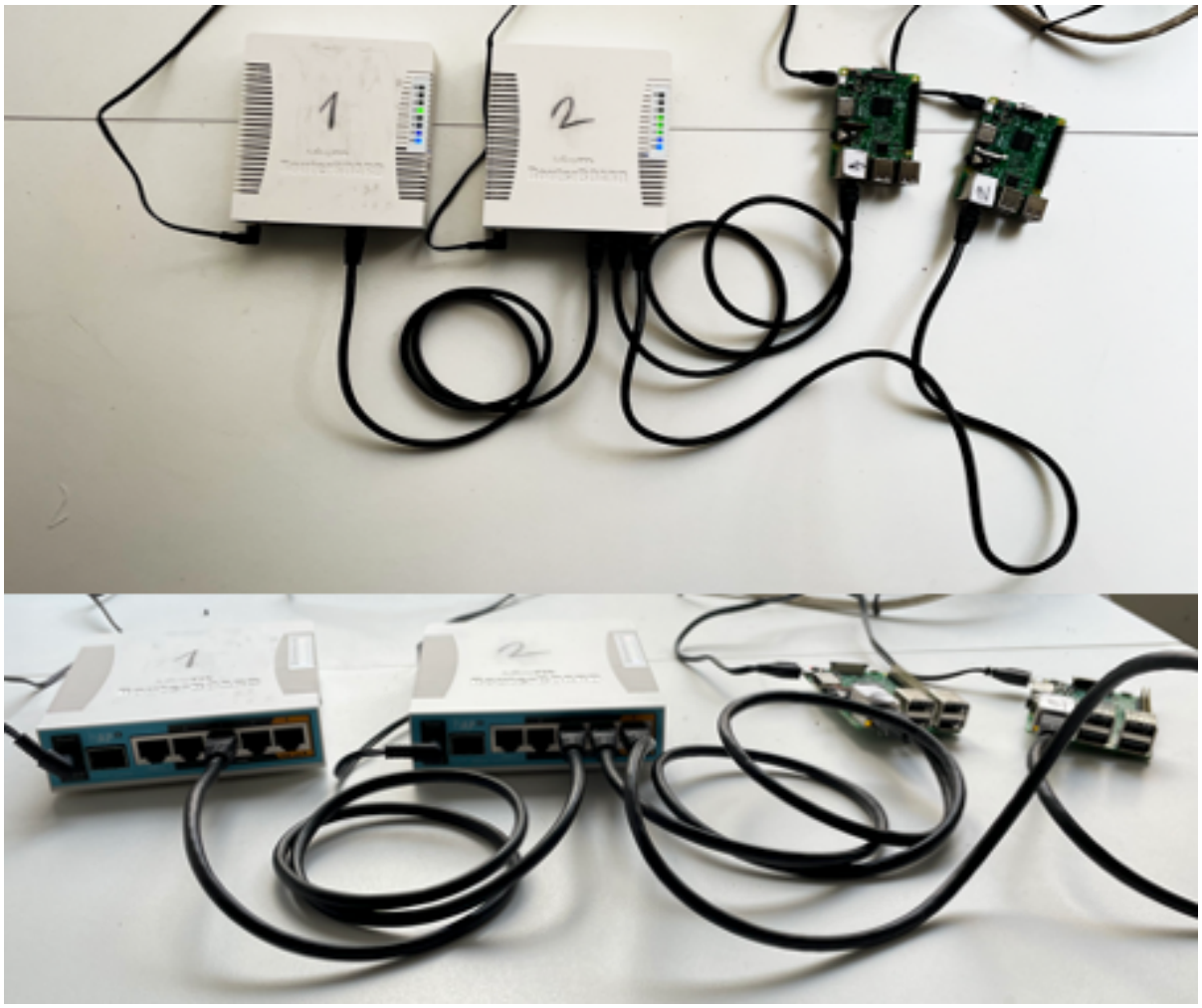


Figura 3-41 Maqueta construida en el laboratorio para la comprobación experimental del protocolo 802.1q (Autoría propia)

3.5.1 Construcción de la maqueta

- Tras resetear el rúter R2 y liberar los puertos ether3, ether4 y ether5 de la configuración de tipo *bridge* (estableciendo *master-port=none* como se indicó en la sección 3.4.1), pasamos a configurar el rúter sin IPs y sin tabla de enrutamiento. Crearemos dos interfaces de tipo bridge llamadas bridgevlan10 y bridgevlan20, añadiendo en la primera la interfaz ether4 y la vlan10 y en la otra ether5 y la vlan20. El ether4 lo conectaremos a la Raspberry1, el ether5 a la Raspberry2 y el ether3 al puerto ether3 del rúter R1. El rúter R2 trabaja solo a nivel de la capa de enlace, intercambiando tramas ethernet y 802.1q, en este aspecto se comporta como un *switch* configurable.

- En el rúter R1 tras liberar ether3, pasamos a construir una vlan10 (10.10.10.0/24) y otra vlan20 (10.20.20.0/24) las asociaremos a la interfaz ether3, quedando una VLAN construida que conectaremos al puerto ether3 del router R2, la conexión se suele llamar enlace troncal (*trunk*) puesto que por él circula tráfico de varias VLANs (vlan10 y vlan20 en nuestro caso). Asociaremos dos servidores DHCP a cada una de las VLAN además de su correspondiente *gateway*, para esto previamente hemos tenido que asociar unas IPs a las vlan10 (10.10.10.1) y vlan20 (10.20.20.1).

Podemos ver en la Figura 3-42 a través de la tabla de rutas (*ip-route*) del rúter R1 que las dos redes VLAN son alcanzables, la del *bridge* es la que lleva configurada por defecto y que no utilizamos.

		Dst. Address	Gateway	Distance	Routing Mark	Pref. Source	
-	DAC	10.10.10.0/24	vlan10 reachable	0		10.10.10.1	
-	DAC	10.20.20.0/24	vlan20 reachable	0		10.20.20.1	
-	DAC	192.168.88.0/2	bridge reachable	0		192.168.88.1	

Figura 3-42 Tabla de rutas del rúter R1 (Autoría propia)

Con la utilidad *Tools>IP Scan* del rúter R1 vamos a comprobar qué IPs han dado a las raspberrys los servidores DHCP. Se muestra el resultado en la Figura 3-43.

		0	10.10.10.4	B8:27:EB:A0:1F:E0
		1	10.10.10.1	
		2	10.20.20.5	B8:27:EB:7C:EF:90
		3	10.20.20.1	

Figura 3-43 IPs raspberry asignadas con los DHCP servers vlan (Autoría propia)

El código de configuración del sistema operativo MikroTikOS en los rúteres R1 y R2 se muestra en la Figura 3-44.

```
/system identity set name=R2

/interface vlan

add interface=ether3 name=vlan10 vlan-id=10
add interface=ether3 name=vlan20 vlan-id=20

/interface bridge

add fast-forward=no name=bridgevlan10
add fast-forward=no name=bridgevlan20

/interface bridge port

add bridge=bridgevlan10 interface=vlan10
add bridge=bridgevlan10 interface=vlan20
add bridge=bridgevlan20 interface=ether5
add bridge=bridgevlan10 interface=ether4


/system identity set name=R1

/interface vlan

add interface=ether3 name=vlan10 vlan-id=10
add interface=ether3 name=vlan20 vlan-id=20

/ip pool

add name=vlan10 ranges=10.10.10.100-10.10.10.200
add name=vlan20 ranges=10.20.20.100-10.20.20.200

/ip dhcp server

add address-pool=vlan10 disabled=no interface=vlan10 name=vlan10 gateway= 10.10.10.1
add address-pool=vlan20 disabled=no interface=vlan20 name=vlan20 gateway= 10.20.20.1


/ip address


add address=10.10.10.1/24 interface=vlan10 network=10.10.10.0
add address=10.20.20.1/24 interface=vlan20 network=10.20.20.0

add address=10.10.10.0/24 netmask=24
add address=10.20.20.0/24 netmask=24
```

Figura 3-44 Código de configuración de los rúteres (Autoría propia)

3.5.2 Comprobación experimental del protocolo 802.1q en VLAN

i. La Raspberrypi2 ejecuta una petición como cliente al servidor web de la raspberrypi1, lo que se consigue introduciendo en el navegador de la Raspberrypi2 la dirección IP del servidor (10.10.10.4) que hemos obtenido del listado de IPs asignadas por DHCP.

ii. Del *sniffer* Wireshark en la Raspberrypi2 y con el filtro http podemos ver que las tramas son Ethernet y no 802.1q. (Tabla 3-2)

Source	Destination	Protocol	Length	Info
10.20.20.5	10.10.10.4	HTTP	589	GET/HTTP/1.1
10.10.10.4	10.20.20.5	HTTP	246	HTTP/1.1 304

589 bytes on wire (4712 bits) on interface eth0
Ethernet II, Src: Raspberry (b8:27:eb:7c:ef:90), Dst: Routerbo (cc:2d:e0:79:2e:2f), Type:IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.20.20.5, Dst: 10.10.10.4
Transmission Control Protocol, Src Port: 35060, Dst Port:80, Seq:1, Ack:1, Len:523
Hypertext Transfer Protocol

Tabla 3-2 Datos capturados en Wireshark (Autoría Propia)

iii. En el rúter R1 con Tools>Packet Sniffer hemos capturado un archivo de tipo PCAP (*Packet Capture)* iaelpilar1.pcap (Figura 3-45) y se ha guardado en la carpeta *Files* del rúter R1. Tras extraerlo, lo hemos abierto en un portátil con Wireshark con el filtro http para ver si los paquetes son ya 802.1q a través del enlace troncal y qué etiquetas tienen.

		File Name	Type	Size
-		flash	disk	
-		flash/pub	directory	
-		flash/skins	directory	
-		iaelpilar1.pcap	.pcap file	61.1 KiB

Figura 3-45 Imagen del archivo para procesarlo con Wireshark (Autoría propia)

Podemos comprobar que tanto la petición HTTP como el envío se producen a través de la conexión *trunk vlan,* con el protocolo 802.1q (Figura 3-46 y Figura 3-47). Con las tramas etiquetadas con los identificadores de VLAN ID:10 e ID:20.

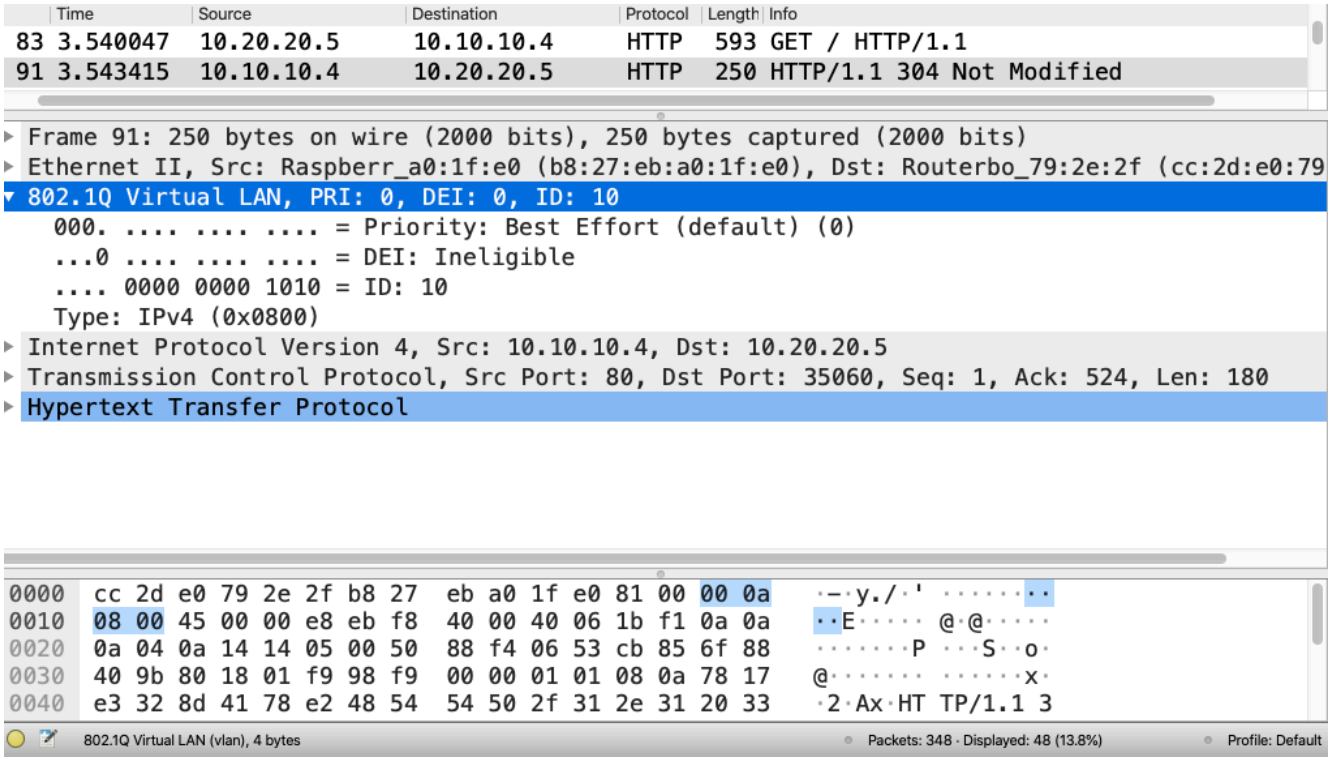


Figura 3-46 Captura del protocolo 802.1q de la trama ID:10 (Autoría propia)

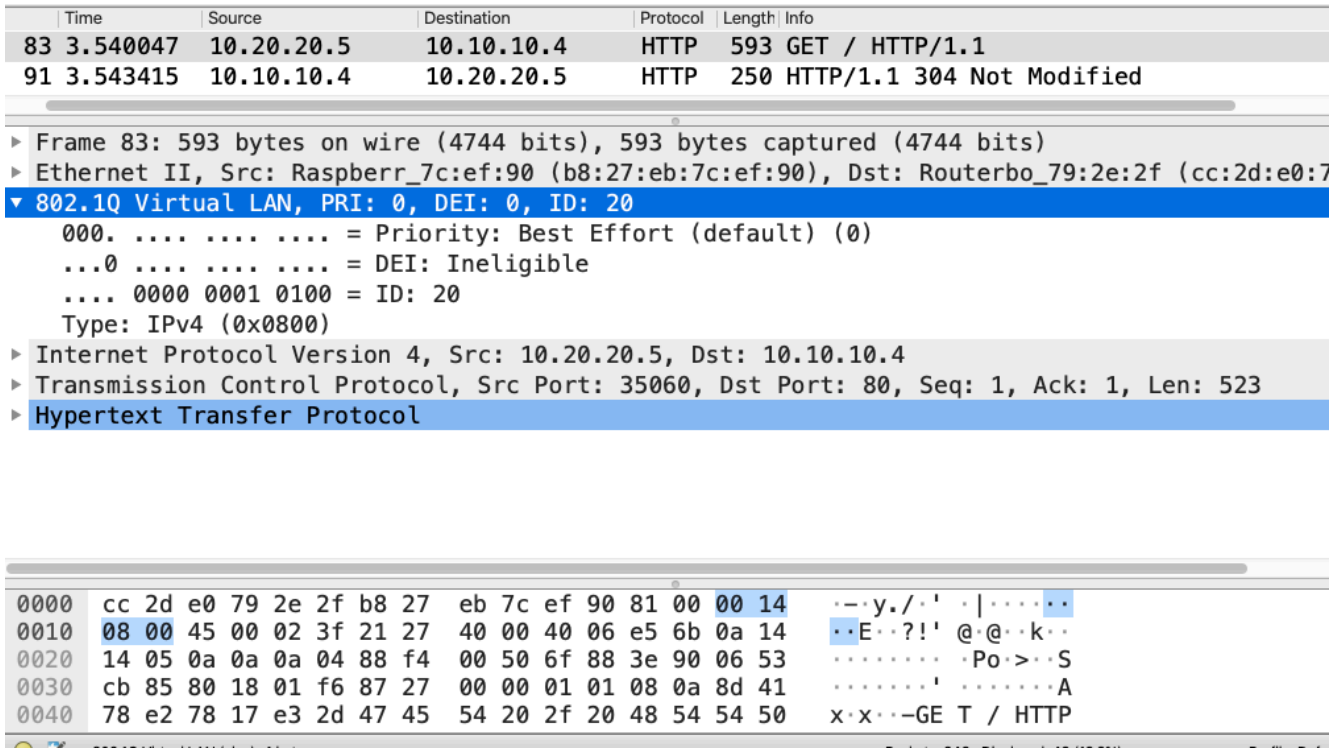


Figura 3-47 Captura del protocolo 802.1q de la trama ID:20 (Autoría propia)

iv. Vemos en detalle la distribución del paquete 802.1q para la ID:20. En el enlace troncal viajan mezcladas tramas de diferentes VLANs que han de ser marcadas antes de ser enviadas por el enlace. El Ethertype 8100 indica protocolo VLAN y los siguientes 2 bytes indican en conjunto la prioridad, forma canónica e identificador VLAN. (Tabla 3-3).

Dir. MAC Destino	Dir. MAC Origen	El Ethertype 8100 indica protocolo VLAN	Prioridad (3 bits) Canonical Format (1 bit), solo se usa en Token Ring Identificador VLAN (12 bits)	Ethertype/ Longitud 0800 indica que el payload es IP	Payload	CRC
6 bytes	6 bytes	2 bytes	2 bytes	2 bytes	593-18=575 bytes	4 bytes
cc:2d:e0: 79:2e:2f	b8:27:eb:7c: ef:90	81:00	00:14	08:00	45:...50	No sale en la captura

Tabla 3-3 Trama de la vlan20 (Autoría propia)

3.6 Maqueta 5: Interoperabilidad entre redes VLANs y no VLANs. Configuración estática y protocolo OSPF.

Se trata de mostrar experimentalmente que el acceso entre redes VLANs y las que no lo son es transparente para el usuario e indistinguible para él. Crearemos dos redes VLANs: la VLAN10 (192.168.10.0/24) y la VLAN20 (192.168.20.0/24), y también otras dos subredes no VLANs: (192.168.1.0/24 y 192.168.3.0/24).

La maqueta (Figura 3-48 y Figura 3-49) constará de:

- 2 Raspberry con los programas instalados Wireshark (*sniffer*) y nginx (servidor web).
- 2 PC.
- 3 routers MikroTik routerboard hAP series.
- 6 cables RJ45 Ethernet.
- 1 switch TP-Link .

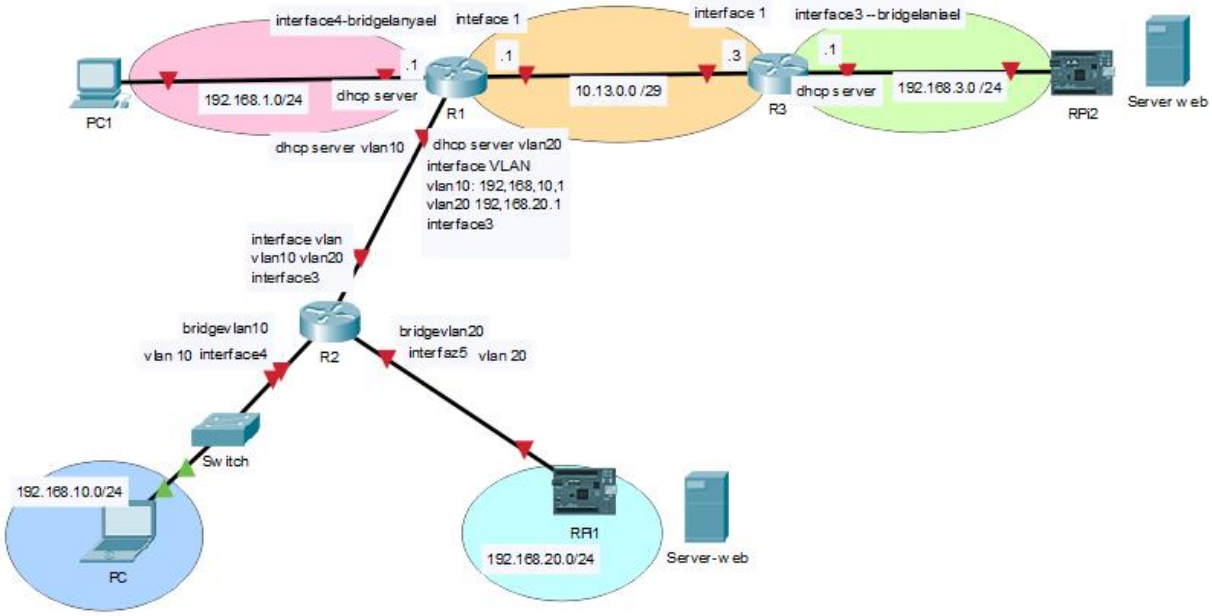


Figura 3-48 Maqueta a construir en el laboratorio de redes VLANs y no VLANs (Autoría propia)

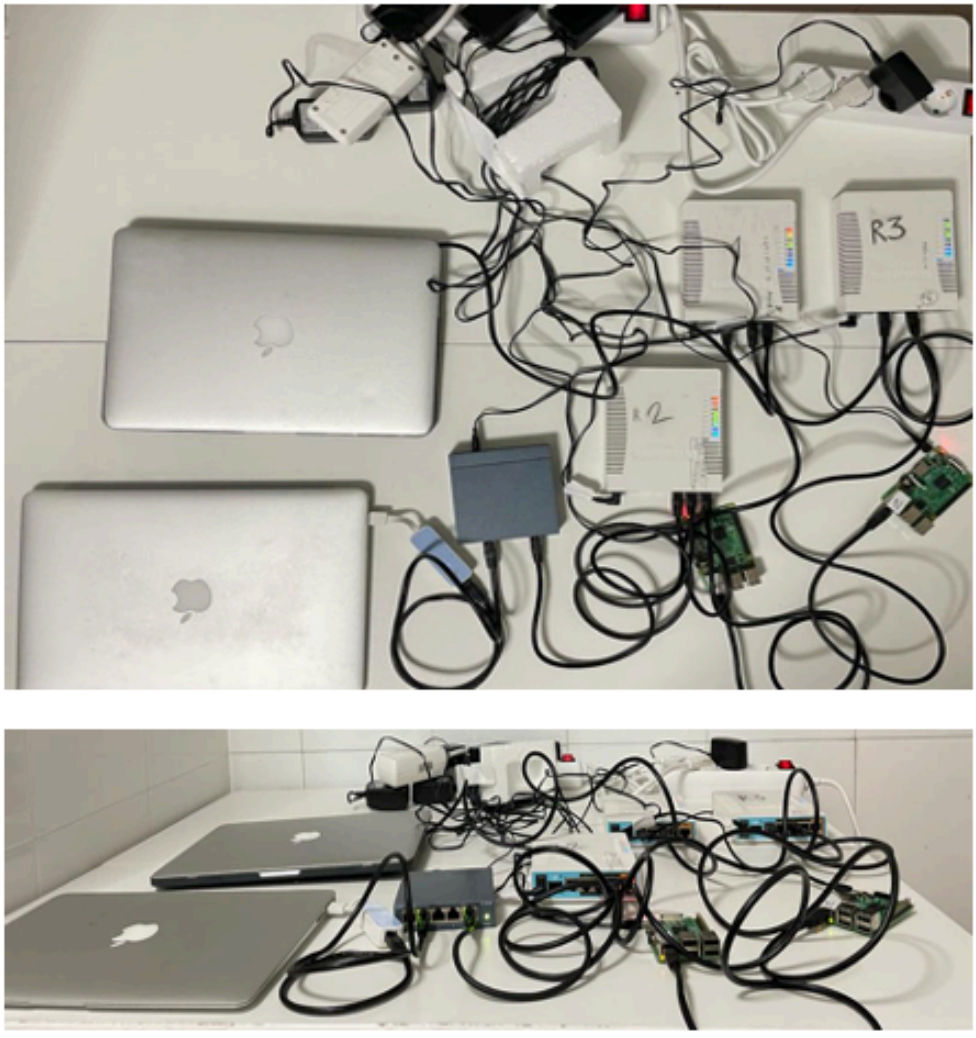


Figura 3-49 Maqueta construida en el laboratorio con redes VLANs y no VLANs (Autoría propia)

3.6.1 Construcción de la Maqueta

- En el rúter R1 tras liberar las interfaces ether3, ether4 y ether5 de la configuración original observamos que ether1 está en lo que MikroTik llama WAN. Se trata de una interfaz pensada para conectarse a un ISP o red de área amplia configurada como cliente de DHCP junto con una implementación NAT y cortafuegos. Por consiguiente, lo transformamos en una interfaz LAN y así estará listo para ser utilizado sin limitaciones que pudieran bloquear la comunicación. Para la conexión troncal creamos una vlan10 (192.168.10.0/24) y otra vlan20 (192.168.20.0/24) les ponemos direcciones IP y asociamos la VLAN al puerto ether3, quedando así la interfaz VLAN construida, además de habilitar dos servidores DHCP para las VLAN 10 y 20. También debemos construir otra interfaz de tipo *bridge*, en ella pondremos la interfaz ether4 y la dirección IP (192.168.1.1/24), asociando a la interfaz *bridge* un servidor DHCP. A la interfaz ether1 le asignamos la IP 10.13.0.1.
- El rúter R2 funciona sin direccionamiento IP estático, la configuración es idéntica a la maqueta realizada en el apartado 3.5 (donde hacía las funciones de un conmutador).
- Para el rúter R3 en la interfaz ether1 asignamos la dirección 10.13.0.3. Construimos una interfaz de tipo *bridge* en la cual se le introduce la interfaz ether3, le asociamos seguidamente a la interfaz de tipo *bridge* la dirección IP 192.168.3.1 y un servidor DHCP.
- Tanto en el rúter R1 y el rúter R3 introducimos las rutas estáticas para las redes que no están directamente conectadas.

Podemos observar en la Figura 3-50 y 3-51 que todas las redes son alcanzables entre sí. Vemos que R2 no tiene direcciones IP (la interfaz bridge de *type-bridge* es la que está configurada en el MikroTik por defecto y no debemos de considerarla, incluso la podríamos borrar).

		Dst. Address	Gateway	Distance
-	DAC	10.13.0.0/29	ether1 reachable	0
-	DAC	192.168.1.0/24	bridgelanyael reachable	0
- D	AS	192.168.3.0/24	10.13.0.3 reachable ether1	1
-	DAC	192.168.10.0/24	vlan10 reachable	0
-	DAC	192.168.20.0/24	vlan20 reachable	0
-	DAC	192.168.88.0/24	bridge reachable	0

Figura 3-51 Tabla de rutas del rúter R1 (Autoría propia)

-	DAC	10.13.0.0/29	ether1 reachable	0
- D	AS	192.168.1.0/24	10.13.0.1 reachable ether1	1
-	DAC	192.168.3.0/24	bridgelaniael reachable	0
- D	AS	192.168.10.0/2	10.13.0.1 reachable ether1	1
- D	AS	192.168.20.0/2	10.13.0.1 reachable ether1	1
-	DAC	192.168.88.0/2	bridge reachable	0

Figura 3-50 Tabla de rutas del rúter R3 (Autoría propia)

3.6.2 Comprobación experimental de la alcanzabilidad entre redes VLANs y no VLANs.

Vamos a comprobar si la raspberrypi2 conectada a la red vlan20 es alcanzable desde las otras tres redes. Para ello, primero averiguaremos la IP de la raspberrypi2 con el comando *ifconfig.* (Figura 3-52). A continuación, desde las otras tres redes mandaremos paquetes ICMP, habiendo comprobado previamente si los servidores DHCP funcionan y se han asignado correctamente direcciones IP. (Figura 3-53, Figura 3-54, Figura 3-55 y Figura 3-56).

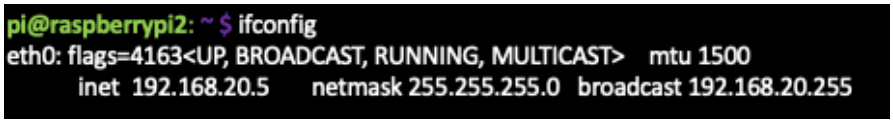


Figura 3-52 IP Raspberrypi2 en la vlan20 (Autoría propia)

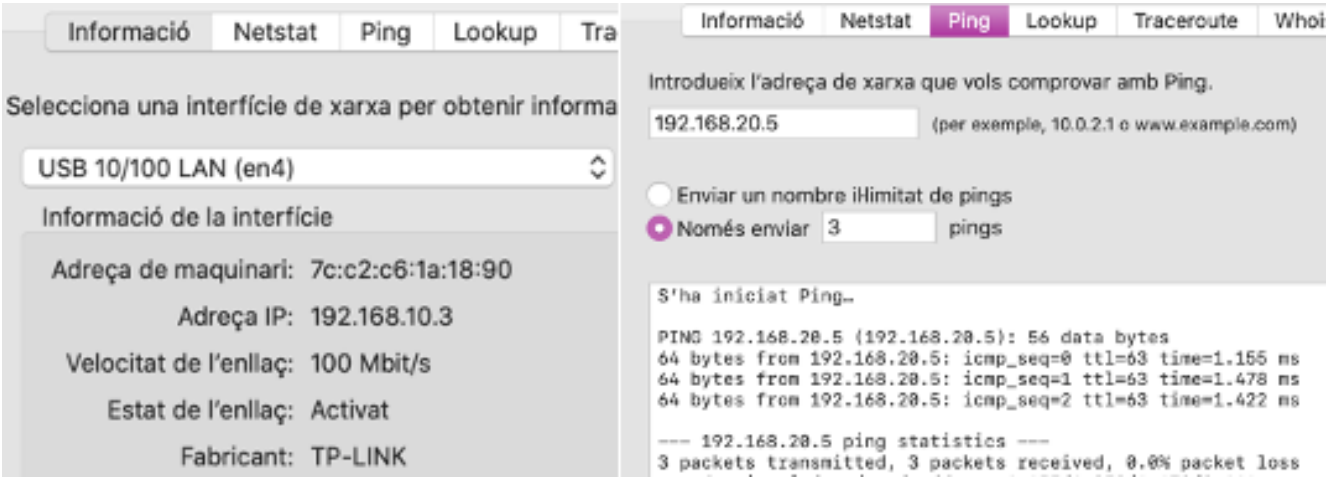


Figura 3-53 ping vlan10 a vlan20 (Autoría propia)

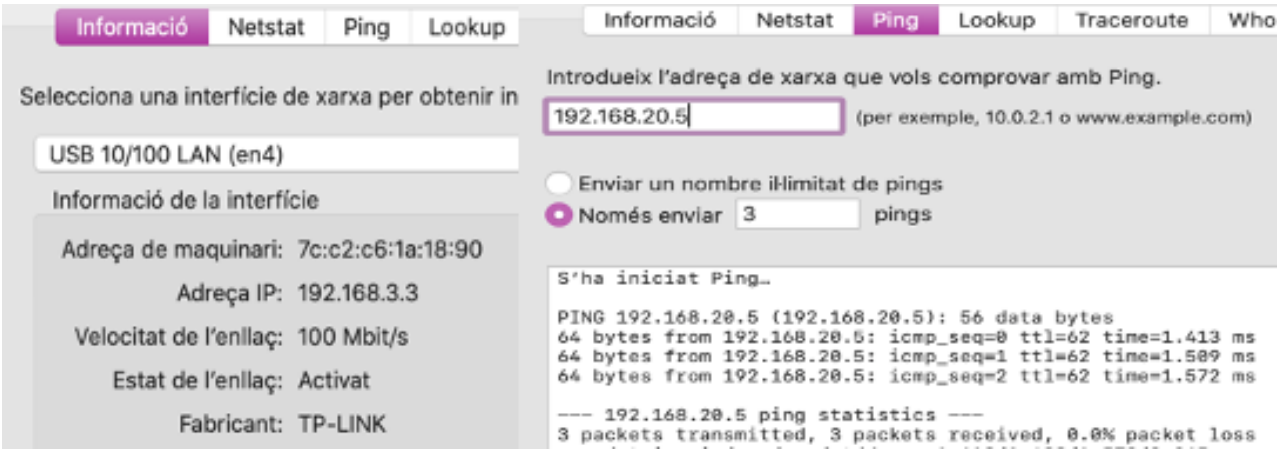


Figura 3-54 ping de la red 192.168.3.0/24 a la vlan20 (Autoría propia)

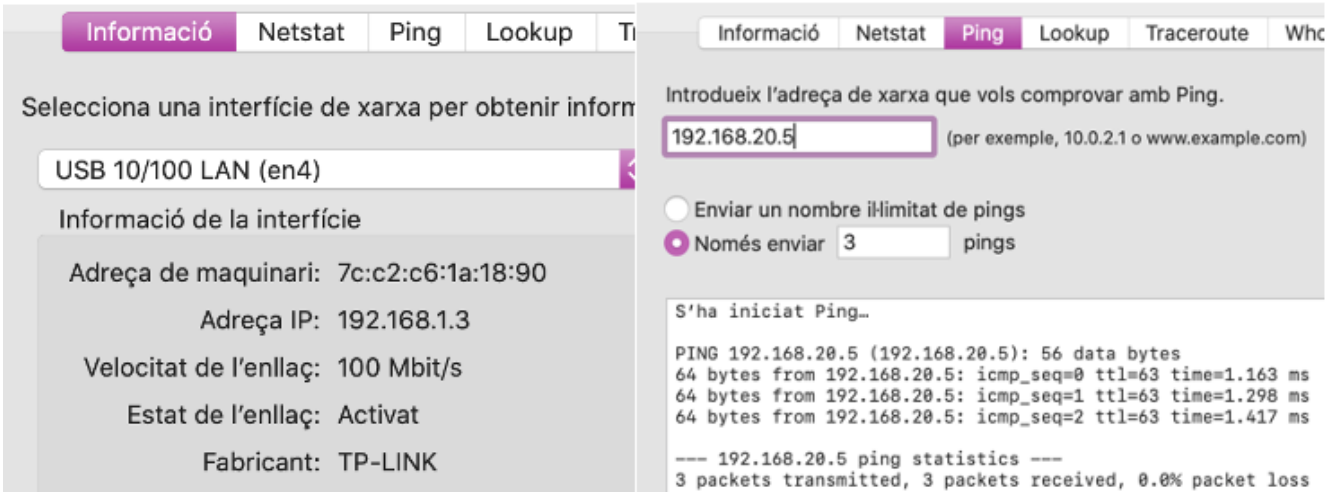


Figura 3-55 ping de la red 192.168.1.0/24 a la vlan20 (Autoría propia)

Comprobamos ahora la conexión de la VLAN 20 con las otras tres redes conectando un PC a ésta y solicitando una página a los servidores web en las otras redes. Al contar solo con dos servidores web instalados en las Raspberrys resulta necesario ir moviendo éstas de red en red. El resultado se muestra en la Figura 3-56.

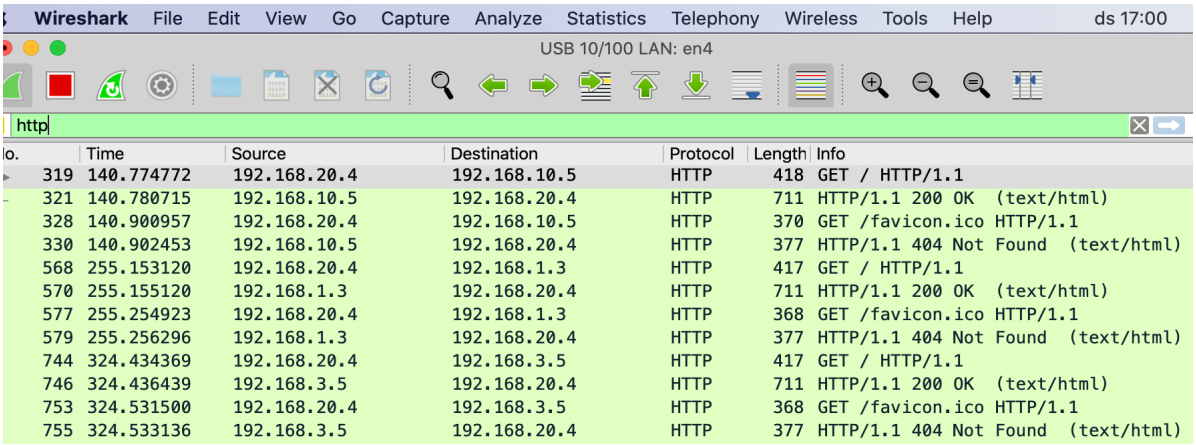


Figura 3-56 Petición desde la vlan 20 de una página web a tres servidores instalados en otras redes (Autoría propia)

3.6.3 Protocolo OSPF

OSPF (*Open Shortest Path Protocol*) estandarizado en el RFC 2328 [47] forma parte de los protocolos de encaminamiento interior IGPs (*Interior Gateway-Routing Protocols*). Se fundamenta en enviar LSAs (*Link State Advertisements*) con los cambios que se producen en la red dentro de zonas llamadas áreas. Se trata de que las bases de datos (*Link State Database*) dentro de un área sean idénticas para todos los rúteres, y a partir de la información de las bases de datos calcular el enrutamiento usando el algoritmo de Dijkstra, consiguiendo que cada rúter genere y mantenga una sola tabla de encaminamiento para todas las redes.

Funcionamiento general del protocolo OSPF:

1- Los rúteres intercambian información (métricas) con todos los otros rúteres de la red a la que pertenecen usando el mecanismo de inundación (*flooding*) enviando LSAs. Un rúter al recibir un LSA lo reenvía por todos sus puertos de salida excepto por el que le ha llegado.

2- Si hay un cambio en la red, será puesto en conocimiento de todos mediante un LSA.

3- A partir de los LSAs y provistos del algoritmo de Dijkstra, los rúteres construyen la base de datos de la topología de la red.

4- Como consecuencia de la información contenida en la base de datos se pasa a rellenar la tabla de encaminamiento

3.6.3.1 Configuración del protocolo OSPF en los rúteres R1 y R3 con el sistema operativo Mikrotik RouterOS.

En el apartado anterior se utilizó encaminamiento estático definido manualmente en R1 y R3. El objetivo es utilizar ahora en su lugar OSPF. Para ello:

- Construimos una interface *bridge* con el nombre *bridgeloopback* asociándole la IP 1.1.1.1/32 para el rúter R1 y lo mismo para el rúter R3 la misma interfaz *bridgeloopback* pero con la IP 3.3.3.3/32. Estas interfaces se comportan como unas interfaces lógicas.

- En la interfaz de configuración web de los rúteres (webfig) entramos en *routing-ospf-instances*, e indicamos *Router ID=1.1.1.1* y *redistribute-connected-routes as type 1*. Esto para el rúter R1, para el R3 exactamente lo mismo, pero cuidando de establecer *Router ID=3.3.3.3.* Podríamos haber puesto en Router ID cualquier IP de la interfaz del rúter correspondiente, pero preferimos ponerle una lógica para asegurarnos de que es una interfaz siempre activa. En cuanto al *redistribute-connect-routes as type 1* el OSPF hará que su direccionamiento se extienda a las redes directamente conectadas a los rúteres.

- En webfig entramos *en routing-ospf-networks* y añadimos la red 10.13.0.0/29 tanto para R1 como para R3.

- En webfig no necesitamos entrar en *routing-ospf-areas* puesto que en nuestro escenario existirá una única área (área por defecto denominada *backbone*).

Podemos ver como el protocolo OSPF está configurado, pero no funcionando en algún caso, porque tiene una distancia administrativa de 110 y nosotros habíamos configurado una ruta estática con distancia 1 que tiene preferencia. La simbología A=active sirve para comprobarlo. (Figuras 3-57 y 3-58).

		Dst. Address	Gateway	Distance
-	DAC	1.1.1.1/32	bridgeloopback reachable	0
-	DAo	3.3.3.3/32	10.13.0.3 reachable ether1	110
-	DAC	10.13.0.0/29	ether1 reachable	0
-	DAC	192.168.1.0/24	bridgelanyael reachable	0
-	Do	192.168.3.0/24	10.13.0.3 reachable ether1	110
- D	AS	192.168.3.0/24	10.13.0.3 reachable ether1	1
-	DAC	192.168.10.0/24	vlan10 reachable	0
-	DAC	192.168.20.0/24	vlan20 reachable	0
-	DAC	192.168.88.0/24	bridge reachable	0

Figura 3-57 Tablas de rutas del rúter R1 (Autoría propia)

		Dst. Address	Gateway	Distance	Routing Mark	Pref. Source
-	DAo	1.1.1.1/32	10.13.0.1 reachable ether1	110		
-	DAC	3.3.3.3/32	bridgeloopback reachable	0		3.3.3.3
-	DAC	10.13.0.0/29	ether1 reachable	0		10.13.0.3
-	Do	192.168.1.0/24	10.13.0.1 reachable ether1	110		
- D	AS	192.168.1.0/24	10.13.0.1 reachable ether1	1		
-	DAC	192.168.3.0/24	bridgelaniael reachable	0		192.168.3.1
-	Do	192.168.10.0/2	10.13.0.1 reachable ether1	110		
- D	AS	192.168.10.0/2	10.13.0.1 reachable ether1	1		
-	Do	192.168.20.0/2	10.13.0.1 reachable ether1	110		
- D	AS	192.168.20.0/2	10.13.0.1 reachable ether1	1		
-	DAC	192.168.88.0/2	bridge reachable	0		192.168.88.1

Figura 3-58 Tablas de rutas del rúter R3 (Autoría propia)

Para conseguir que sea el protocolo OSPF el que esté funcionando y no la configuración estática basta con cambiar la distancia administrativa de ésta por otra mayor que 110, por ejemplo 150. Para ello vamos a *IP>Routes> Distance* en WebFig. Se puede observar que en este caso la ruta estática, de mayor coste, ha dejado de estar activa. (Figura 3-59 y 3-60).

		Dst. Address	Gateway	Distance	Routing Mark	Pref. Source
-	DAC	1.1.1.1/32	bridgeloopback reachable	0		1.1.1.1
-	DAo	3.3.3.3/32	10.13.0.3 reachable ether1	110		
-	DAC	10.13.0.0/29	ether1 reachable	0		10.13.0.1
-	DAC	192.168.1.0/24	bridgelanyael reachable	0		192.168.1.1
-	DAo	192.168.3.0/24	10.13.0.3 reachable ether1	110		
- D	S	192.168.3.0/24	10.13.0.3 reachable ether1	150		
-	DAC	192.168.10.0/24	vlan10 reachable	0		192.168.10.1
-	DAC	192.168.20.0/24	vlan20 reachable	0		192.168.20.1
-	DAC	192.168.88.0/24	bridge reachable	0		192.168.88.1

Figura 3-59 Tabla de rutas OSPF del rúter R1 (Autoría propia)

		Dst. Address	Gateway	Distance	Routing Mark	Pref. Source
-	DAo	1.1.1.1/32	10.13.0.1 reachable ether1	110		
-	DAC	3.3.3.3/32	bridgeloopback reachable	0		3.3.3.3
-	DAC	10.13.0.0/29	ether1 reachable	0		10.13.0.3
-	DAo	192.168.1.0/24	10.13.0.1 reachable ether1	110		
- D	S	192.168.1.0/24	10.13.0.1 reachable ether1	150		
-	DAC	192.168.3.0/24	bridgelaniael reachable	0		192.168.3.1
-	DAo	192.168.10.0/2	10.13.0.1 reachable ether1	110		
- D	S	192.168.10.0/2	10.13.0.1 reachable ether1	150		
-	DAo	192.168.20.0/2	10.13.0.1 reachable ether1	110		
- D	S	192.168.20.0/2	10.13.0.1 reachable ether1	150		
-	DAC	192.168.88.0/2	bridge reachable	0		192.168.88.1

Figura 3-60 Tabla de rutas OSPF del rúter R3 (Autoría propia)

En la Figura 3-61 se puede observar una captura de paquetes OSPF visualizada a través de la herramienta Wireshark:

- El mensaje *hello* sigue el encabezado del paquete IP, con el protocolo IP decimal 89 (hexadecimal 0x59) del protocolo OSPF (RFC 1583). Se envía a la dirección multicast 224.0.0.5 destinada a todos los rúteres que hablan OSPF, estos escuchan los paquetes enviados a la dirección IP multicast 224.0.0.5 para recibir los paquetes *hello* y aprender sobre los nuevos vecinos. En nuestro caso la IP de origen es la interfaz del rúter R3 10.13.0.3 que es la que se utiliza para conectarse a la interfaz del rúter R1 que es la 10.13.0.1.
- En la trama de enlace de datos tenemos la dirección MAC de origen de la interfaz1 del rúter R3 que es cc:d2:e0:79:2e:5d y con una MAC de destino multicast 01:00:5e:00:00:05.
- En el encabezado del paquete OSPF (*OSPF Header*) tenemos el ID del rúter R3 3.3.3.3, ID del área (*backbone*) 0.0.0.0 y el tipo de paquete OSPF, en este caso 0x01 saludo (*Hello packet).*
- Dentro del contenido del paquete del saludo tenemos el intervalo de saludo que en este caso es de 10 segundos y el intervalo muerto OSPF que suele ser por defecto 4 veces el intervalo de saludo (40 segundos).

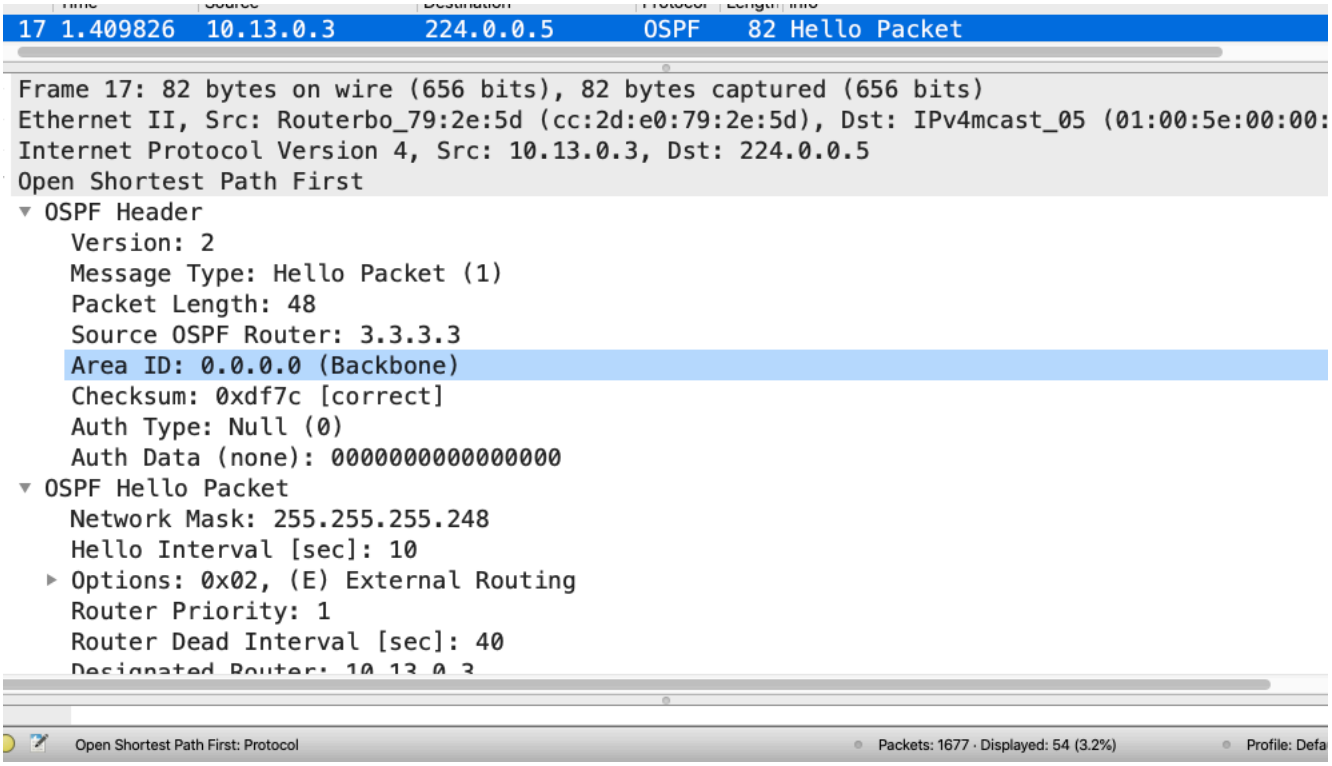


Figura 3-61 Protocolo OSPF captura de un paquete *hello* (Autoría propia)

4 CONCLUSIONES Y LÍNEAS FUTURAS

4.1 Conclusiones

En este trabajo de final de grado, se ha conseguido el objetivo principal de diseñar montajes (maquetas) con equipos reales: rúteres MikroTik, conmutadores (s*witches*) y equipos finales representados por Raspberry Pi (RPi) (alguna como servidor) con el fin de comprobar físicamente en ellas protocolos como ARP, HTTP, DHCP, 802.1q y OSPF. Se ha empleado una metodología ágil, basada en pequeños desarrollos de complejidad incremental permitiendo avanzar a través de una serie de actividades a realizar en cada uno de ellos, con la finalidad de obtener un instrumento con el que complementar el aprendizaje teórico de una asignatura de redes. La finalidad última sería hacer más atractivo el proceso de aprendizaje mediante el uso de metodologías activas.

4.2 Líneas futuras

Entre las posibles líneas futuras de continuación de este trabajo se destacan:

- En la última maqueta debería habilitarse en algún puerto de algún rúter que tengamos libre una conexión WAN, para probar el funcionamiento del NAT y establecer las configuraciones del cortafuegos que se crean pertinentes para de esta forma tener acceso al exterior (Internet pública).
- Con las otras dos LAN de la última maqueta hacer una conexión vía IPsec con otra red accesible a través de Internet comprobando en cada paquete IP los diversos protocolos disponibles en seguridad: integridad, autenticidad y confiabilidad.
- Protocolo OSPF multiarea añadiendo más rúteres a la última maqueta.
- Protocolo BGP (*Border Gateway Protocolo*) sobre más rúteres añadidos a la última maqueta.
- IoT. Se trataría de configurar una de las Raspberry Pi en Python para utilizarla como robot colocándole un motor que pudiera ser controlado desde otras Raspberry Pi en otras redes.

5 BIBLIOGRAFÍA

[1]. Zen, Vittore. *Theory, laboratories and exercises for Mikrotik RouterOS-Routing.* ISBN:978-1686046964.

[2]. Montañana, Rogelio. Historia de Internet Curso de Redes, UPV. [En línea] [Citado el: 20 de enero de 2022]. https://www.youtube.com/watch?v=v_-bHKhDhzA.

[3]. OSPF *multiacceso con MikroTik.* [En línea] [Citado el 22 de enero de 2022]. https://www.youtube.com/watch?v=mRt4raJkHMk.

[4]. OSPF, dibujos y pasos a seguir para configurar. *www.mikrotik.com.* [En línea] https://mum.mikrotik.com/presentations/HN18/presentation_5623_1532719027.pdf.

[5]. Manual: OSPF Case Studies MikroTik documentation. *www.mikrotik.com.* [En línea] https://wiki.mikrotik.com/wiki/Manual:OSPF_Case_Studies#Configuring_OSPF.

[6]. Tanenbaum, Andrew. S. *Redes de Computadoras.* s.l. : Editorial Genios, 2010. ISBN:970-26-0162-2

[7]. Configuración OSPF MikroTik. [En línea] https://www.youtube.com/watch?v=h9vTEhPQYAg.

[8]. GNS3. *www.gns3.com.* [En línea]

[9]. Instalación y configuración de GNS3 en MacOs. *www.youtube.com.* [En línea] [Citado el: 20 de Enero de 2022.]

[10]. GNS3 Mac OS- Instalación y ejercicio de routers. [En línea] https://www.youtube.com/watch?v=oGBnq0nAYiY[Citado el: 20 de Enero de 2022.]

[11]. VMware. *www.vmware.com.* [En línea] [Citado el: 20 de Enero de 2022.]

[12]. Wireshark simulador de paquetes. *www.wireshark.org.* [En línea] [Citado el: 20 de Enero de 2022.]

[13]. MikroTik. *www.mikrotik.com.* [En línea] [Citado el: 20 de Enero de 2022.]

[14]. Tipos de licencias MikroTik. *www.soporte.syscom.mx.* [En línea] [Citado el: 20 de Enero de 2022.]

[15]. Introducció a les xarxes telemàtiques, apunts de Grau de la ETSETB. *www.wuolah.com.* [En línea] https://www.wuolah.com/es/universidad-politecnica-de-catalunya/upc-escuela-tecnica-superior-de-ingenieria-de-telecomunicacion-etsetb.

[16]. AS y IXP. *https://tutorialesenlinea.es.* [En línea] 19 de enero de 2022. [Citado el: 20 de Enero de 2022.] https://tutorialesenlinea.es/-internet-_peering-bgp-y-sistemas-autonomos_as.html.

[17]. CDIO ETSETB. *www.upc.edu.* [En línea] [Citado el: 21 de Enero de 2022.] https://www.upc.edu/gestioestudis/files/files_graus/diptic/369_cat.pdf.

[18]. Zen, Vittore. *Theory, laboratories, MikroTik RouterOS-started.*

[19]. OSPF multiarea. *www.youtube.com.* [En línea] [Citado el: 20 de Enero de 2022.] https://www.youtube.com/watch?v=ksVAvvNdksU.

[20]. Tipos de simuladores de redes. [En línea] [Citado el: 21 de Enero de 2022.] https://jennifervacaicas.blogspot.com/2014/09/simuladores-de-redes.html.

[21]. Cisco Virtual Internet Routing Lab. *www.cisco.com.* [En línea] [Citado el: 22 de Enero de 2022.] https://learningnetwork.cisco.com/s/virl.

[22]. Cisco Packet Tracer. *www.netacad.com.* [En línea] [Citado el: 20 de Enero de 2022.] https://www.netacad.com/es/courses/packet-tracer.

[23]. CISCO formación eventos y seminarios web. [En línea] [Citado el: 22 de enero de 2022.] https://www.cisco.com/c/es_es/training-events.html.

[24]. Programa certificación CISCO CCNA. [En línea] https://www.topformacion.es/curso-cisco-ccna-p30154.html. [Citado el: 20 de Enero de 2022.]

[25]. Montañana, Rogelio. UPV CURSO REDES. [En línea] [Citado el: 30 de enero de 2022.] https://www.youtube.com/watch?v=98Igov-JmVI&list=PLomN84AdULIBcoI8Rb98dnompliIktJk9&index=124.

[26]. Banco mundial. [En línea] [Citado el: 9 de Febrero de 2022.] https://datos.bancomundial.org/indicator/IT.NET.USER.ZS?end=2020&start=1969.

[27]. Cisco Internet of Things. [En línea] [Citado el: 9 de Febrero de 2022.] https://www.cisco.com/c/dam/global/es_es/assets/executives/pdf/Internet_of_Things_IoT_IBSG_0411FINAL.pdf.

[28]. Imagen raspberrypiOS. [En línea] [Citado el: 2022 de ferebro de 12.] https://www.raspberrypi.com/software/.

[29]. Modelo OSI y TCP/IP. [En línea] [Citado el: 20 de febrero de 2022.] http://juanp-internet.blogspot.com/2010/07/tcpip.html.

[30]. Riverbed Modeler. *https://www.riverbed.com/en-gb/products/npm/riverbed-modeler.html.* [En línea] [Citado el: 20 de febrero de 2022.]

[31]. MIT. [En línea] [Citado el: 20 de febrero de 2022.] https://www.mit.edu.

[32]. NETSIM. [En línea] [Citado el: 10 de marzo de 2022.] https://www.boson.com/netsim-cisco-network-simulator.

[33]. NS-2 Y NS-3. [En línea] [Citado el: 10 de Marzo de 2022.] https://www.nsnam.org/support/faq/ns2-ns3/.

[34]. JIMSIM SIMULATOR. [En línea] [Citado el: 10 de Marzo de 2022.] http://freshmeat.sourceforge.net/projects/jimsim.

[35]. CORE. [En línea] [Citado el: 10 de marzo de 2022.] https://www.nrl.navy.mil.

[36]. KIVANS. [En línea] [Citado el: 22 de marzo de 2022.] http://www.aurova.ua.es/kiva/.

[37]. CLOONIX. [En línea] [Citado el: 10 de marzo de 2022.] https://clownix.net.

[38]. MININET. [En línea] [Citado el: 12 de Marzo de 2022.] http://mininet.org/download/.

[39]. OMNET. [En línea] [Citado el: 11 de Marzo de 2022.] https://omnetpp.org.

[40]. MARIONNET. [Citado el: 11 de Marzo de 2022.] https://www.marionnet.org/site/index.php/en/.

[41]. Microsoft Message Analyzer. [En línea] [Citado el: 11 de marzo de 2022.] https://docs.microsoft.com/en-us/message-analyzer/microsoft-message-analyzer-operating-guide.

[42]. Cursos MikroTik. [Citado el: 11 de Marzo de 2022.] https://mikrotik.com/training/about.

[43]. Referencia estandarizada de ARP. [En línea] [Citado el: 11 de marzo de 2022.] https://datatracker.ietf.org/doc/html/rfc826.

[44]. Referencia estadarizada DHCP. [En línea] [Citado el: 11 de Marzo de 2022.] https://datatracker.ietf.org/doc/html/rfc2131.

[45]. Referencia estandarizada HTTP. [En línea] [Citado el: 10 de marzo de 2022.] https://datatracker.ietf.org/doc/html/rfc2616.

[46]. Referencia estandarizada 802.1Q. [En línea] [Citado el: 10 de Marzo de 2022.] https://www.ietf.org/rfc/rfc2674.txt.

[47]. Referencia estandarizada OSPF. [En línea] [Citado el: 10 de Marzo de 2022.] https://datatracker.ietf.org/doc/html/rfc2328.

[48]. Protocolo DHCP. [En línea] [Citado el: 12 de Marzo de 2022.] https://es.m.wikipedia.org/wiki/Archivo:Sesión_DHCP.svg.