

Trabajo Fin de Grado en Ingeniería de las Tecnologías de Telecomunicación

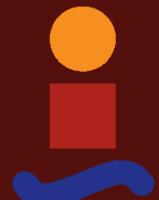
Estudio y Comparación de Métodos para la Implementación de Conectividad IPv6 en Redes sin Soporte Nativo

Autor: Adrián Garrido Real

Tutor: Juan Antonio Ternero Muñiz

Dpto. de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2025



Trabajo Fin de Grado
en Ingeniería de las Tecnologías de Telecomunicación

Estudio y Comparación de Métodos para la Implementación de Conectividad IPv6 en Redes sin Soporte Nativo

Autor:
Adrián Garrido Real

Tutor:
Juan Antonio Ternero Muñiz
Profesor colaborador

Dpto. de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2025

Trabajo Fin de Grado: Estudio y Comparación de Métodos para la Implementación de Conectividad IPv6 en Redes sin Soporte Nativo

Autor: Adrián Garrido Real

Tutor: Juan Antonio Ternero Muñiz

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2025

El Secretario del Tribunal

Agradecimientos

En primer lugar, me gustaría agradecer a mis padres por haberme apoyado durante todos estos años, por confiar siempre en mí y por ayudarme a no perder nunca las fuerzas ni la motivación para seguir adelante. También, agradecer a mi hermana, mi familia y mis amigos por estar siempre presentes, especialmente en los momentos más difíciles, brindándome su apoyo y su cariño.

Quisiera dedicar una mención muy especial a mi abuela. Ella siempre tuvo la ilusión de verme graduado y cumplir mis sueños. Aunque, por desgracia, no haya podido acompañarme en este momento, sé que desde el cielo nunca ha dejado de apoyarme. Ella ha sido mi motor y mi fuerza en los momentos en los que estuve a punto de rendirme.

En general, quiero agradecer a todas las personas que siempre han estado a mi lado, tanto en los buenos como en los malos momentos. Aunque algunas ya no están a mi lado, siempre guardaré cariño especial hacia cada una de ellas.

Por último, agradecer a todos los profesores que he tenido a lo largo de la carrera, por compartir todo su conocimiento y contribuir a mi formación. En particular, quiero agradecer a mi profesor Juan Antonio Ternerero por acompañarme durante el desarrollo de este proyecto y por estar dispuesto a ayudarme en cualquier necesidad.

Resumen

El uso de IPv6 ha dejado de ser algo futuro para convertirse en una necesidad actual, debido a la escasez de direcciones IPv4 que está provocando la migración hacia la nueva versión del protocolo IP. Este Trabajo Fin de Grado tiene como objetivo analizar y estudiar diferentes servicios que proporcionen conectividad IPv6 en redes donde la operadora no ofrece este direccionamiento de manera nativa.

Para ello, el proyecto se inicia con un análisis detallado del protocolo IPv6, destacando sus características más relevantes. Seguidamente, analizamos distintos servicios que proporcionen conectividad IPv6 a la red, como TunnelBroker de Hurricane Electric, una VPN con soporte IPv6 y la creación de una VPN Wireguard configurando como servidor una VPS contratada, la cual posee una dirección pública IPv4 y direcciones globales IPv6.

Posteriormente, estudiamos distintos parámetros de red claves para el correcto funcionamiento de la red, tales como latencia, RTT, pérdida de paquetes, ancho de banda, jitter, MTU y tiempo de resolución DNS. Para su evaluación, se emplean herramientas profesionales como PingPlotter, NetScanTools, Wireshark, Mtr y SpeedTest para obtener mediciones precisas de cada servicio implementado.

Para el proyecto, llevamos a cabo un conjunto de pruebas prácticas donde implementamos y configuramos los distintos servicios, midiendo y analizando cada parámetro estudiado previamente, comparando los resultados obtenidos para extraer una conclusión sobre cada servicio. Además, podremos observar las ventajas y desventajas de cada uno de ellos. Así, concluimos que mientras la conectividad Ipv6 nativa ofrece el mejor rendimiento de manera general, soluciones como los túneles o VPNs son alternativas fiables y que ofrecen un buen rendimiento para redes que no tengan conectividad Ipv6 de manera nativa.

Finalmente, el proyecto cuenta con un apartado donde se incluyen propuestas para poder mejorarlo con nuevas extensiones futuras, convirtiéndose como una guía para implementar IPv6 en redes donde no se ofrece Ipv6 de manera nativa.

Abstract

The use of IPv6 is no longer a matter of the future but a current necessity, driven by the depletion of IPv4 addresses that is accelerating the migration to the new version of the IP protocol. This Final Degree Project aims to analyze and study different services that provide IPv6 connectivity in networks where the Internet Service Provider does not offer this addressing natively.

To this end, the project begins with a detailed analysis of the IPv6 protocol, highlighting its most relevant features. It then examines various services that enable IPv6 in the network, such as Hurricane Electric's TunnelBroker, a VPN with IPv6 support, and the deployment of a WireGuard VPN configured on a rented VPS, which has both public IPv4 and global IPv6 addresses.

Subsequently, key network parameters essential for proper performance are studied, including latency, RTT, packet loss, bandwidth, jitter, MTU, and DNS resolution time. For their evaluation, professional tools such as PingPlotter, NetScanTools, Wireshark, MTR, and SpeedTest are employed to obtain precise measurements for each implemented service.

The project incorporates a set of practical tests in which these services are configured and analyzed, measuring each parameter and comparing the results to draw conclusions about their efficiency. Thus, it is observed that while native IPv6 connectivity generally delivers the best performance, solutions like tunnels or VPNs are reliable alternatives that provide good results in networks without native IPv6 support.

Finally, the work includes proposals for future improvements and extensions, serving as a practical guide for implementing IPv6 in environments where native support is not available.

Índice

Agradecimientos	vii
Resumen	ix
Abstract	xi
Índice	xii
Índice de Tablas	xiv
Índice de Figuras	xvi
Notación	xx
1 Introducción y Objetivo	1
1.1 <i>Objetivos</i>	1
1.2 <i>Metodología de trabajo</i>	2
1.3 <i>Estructura del Proyecto</i>	2
2 Internet Protocol Version 6 (IPv6)	3
2.1. <i>Introducción</i>	3
2.1.1 Cabecera IPv6	3
2.1.2 Direccionamiento IPv6	4
2.1.3 Subredes	6
2.2 <i>ICMPv6</i>	6
2.2.1 Neighbor Discovery	7
2.3 <i>Seguridad Ipv6</i>	9
2.4 <i>Diferencias Ipv6 vs Ipv4</i>	9
2.5 <i>Visión futura</i>	10
3 Métodos de acceso a IPv6	11
3.1 <i>Conectividad Ipv6 Nativa</i>	11
3.1.1 Funcionamiento	11
3.1.2 Configuración Servicio Nativo	12
3.2 <i>Túneles Ipv6 sobre Ipv4</i>	13
3.2.1 TunnelBroker (Hurricane Electric)	13
3.2.2 VPN con soporte Ipv6	17
3.2.3 VPN con Wireguard con servidor una VPS	19
3.3 <i>Comparación de Servicios</i>	23
4 Parámetros de evaluación y herramientas empleadas	25

4.1 Parámetros de red	25
4.1.1 Latencia	25
4.1.2 Pérdida de paquetes	26
4.1.3 Ancho de banda	27
4.1.4 Jitter	28
4.1.5 MTU	29
4.1.6 Tiempo de resolución DNS	30
4.1.7 Rutas	31
4.2 Herramientas utilizadas	32
4.2.1 NetScanTools	32
4.2.2 PingPlotter	32
4.2.3 Mtr (My traceroute)	33
4.2.4 Wireshark	34
4.2.5 Speedtest Cloudflare	35
4.2.6 Netsh/ifconfig	36
4.2.7 Dig/nslookup	37
4.3 Consideraciones del entorno de prueba	38
5 Resultados y análisis de las mediciones	39
5.1 DNS Optimo	39
5.1.1 Nativo	39
5.1.2 TunnelBroker de Hurricane Electric implementado en Windows	43
5.1.3 TunnelBroker de Hurricane Electric implementado en Ubuntu	47
5.1.4 VPN de Hide.me	51
5.1.5 Wireguard+VPS	51
5.2 Comparación de servicios	54
5.2.1 RTT	54
5.2.2 Pérdida de paquetes y ruta utilizada para cada destino	57
5.2.3 Tiempo de resolución DNS	69
5.2.4 Prueba en descargas de gran tamaño	70
5.2.5 Ancho de banda	71
5.2.6 Jitter	71
5.2.7 Número de saltos	72
5.2.8 MTU	72
6 Conclusiones y mejoras futuras	74
Anexo A. Equipos utilizados	76
A.1 MSI GF63 Thin 9SC	76
A.2 Laptop HP	76
A.3 Proveedor de red contratado y tipo de conexión empleado	77
Anexo B. VPS contratada	78
Anexo C. Script para DNS óptimo	80
Referencias	85
Índice de Conceptos	88

ÍNDICE DE TABLAS

Tabla 2-1 Diferencias IPv6 vs IPv4	10
Tabla 3-1 Similitudes y Diferencias Servicios	26
Tabla 4-1 Ancho de banda recomendado	30
Tabla 5-1 Paquetes Pérdidos IPv6 Nativo	43
Tabla 5-2 Paquetes Pérdidos Ipv6 Hurricane Electric Windows	47
Tabla 5-3 Paquetes Pérdidos Ipv6 Hurricane Electric Ubuntu	50
Tabla 5-4 Paquetes Pérdidos VPN Wireguard	54
Tabla 5-5 Paquetes Pérdidos Servicios	59
Tabla 5-6 Rutas IPv6 Nativo	60
Tabla 5-7 Rutas IPv6 Hurricane Electric Windows	61
Tabla 5-8 Rutas IPv6 Hurricane Electric Ubuntu	63
Tabla 5-9 Rutas IPv6 VPN Hide.me	65
Tabla 5-10 Rutas IPv6 VPN Wireguard	67
Tabla 5-11 Ancho de banda Servicios	71
Tabla 5-12 Jitter Servicios	73
Tabla 5-13 Número Saltos Servicios	74
Tabla 5-14 MTU Servicios	74
Tabla 6-1 Valoración Servicios	76

ÍNDICE DE FIGURAS

Figura 1-1 Impacto IPv6 global	1
Figura 2-1 Cabecera IPv6	4
Figura 2-2 Cabecera ICMPv6	7
Figura 3-1 Paquete IPv6 Nativo	13
Figura 3-2 Interfaz Servicio Nativo	13
Figura 3-3 Conectividad IPv6 Nativo	14
Figura 3-4 Paquete IPv4 Hurricane Electric	15
Figura 3-5 Paquete Ipv6 Hurricane Electric	15
Figura 3-6 Interfaz Hurricane Electric	16
Figura 3-7 Configuraciones Hurricane Electric	17
Figura 3-8 Conectividad IPv6 Hurricane Electric Windows	17
Figura 3-9 Conectividad IPv6 Hurricane Ubuntu	18
Figura 3-10 Paquete UDP VPN Hide.me	19
Figura 3-11 Configuración VPN Hide.me	19
Figura 3-12 Interfaz VPN Hide.me	20
Figura 3-13 Paquete IPv4 VPN Wireguard	21
Figura 3-14 Paquete IPv6 VPN Wireguard	21
Figura 3-15 Configuración Servidor Wireguard	22
Figura 3-16 Configuración Cliente Wireguard	23
Figura 3-17 Conectividad IPv6 VPN Wireguard	24
Figura 4-1 RTT	26
Figura 4-2 Pérdida de paquetes	27
Figura 4-3 Ancho de banda	28
Figura 4-4 Jitter	29
Figura 4-5 MTU	30
Figura 4-6 Tiempo de resolución DNS	30
Figura 4-7 Rutas	31
Figura 4-8 Herramienta NetScanTools	32
Figura 4-9 Herramienta PingPlotter	33
Figura 4-10 Herramienta Mtr	34
Figura 4-11 Herramienta Wireshark	35

Figura 4-12 Web Speedtest	36
Figura 4-13 Comando Netsh Interface IPv6 Show Subinterfaces	36
Figura 4-14 Comando ifconfig	37
Figura 4-15 Herramienta nslookup	37
Figura 4-16 Herramienta dig	38
Figura 5-1 Resultado PingPlotter DNS IPv6 Nativo	40
Figura 5-2 RTT Mínima DNS IPv6 Nativo	40
Figura 5-3 RTT Máxima DNS IPv6 Nativo	41
Figura 5-4 RTT Media DNS IPv6 Nativo	41
Figura 5-5 RTT de resolución DNS IPv6 Nativo	42
Figura 5-6 RTT Minima Dominios IPv6 Nativo	42
Figura 5-7 RTT Máxima Dominios IPv6 Nativo	42
Figura 5-8 RTT Media Doninios IPv6 Nativo	43
Figura 5-9 Resultado PingPlotter IPv6 Hurricane Electric	44
Figura 5-10 RTT Mínima DNS Hurricane Electric Windows	44
Figura 5-11 RTT Maxima DNS Hurricane Electric Windows	45
Figura 5-12 RTT Media DNS Hurricane Electric Windows	45
Figura 5-13 Tiempo de resolución DNS Hurricane Electric Windows	46
Figura 5-14 RTT Mínima Dominios Hurricane Electric Windows	46
Figura 5-15 RTT Máxima Dominios Hurricane Electric Windows	47
Figura 5-16 RTT Media Dominios Hurricane Electric Windows	47
Figura 5-17 RTT Mínima DNS Hurricane Electric Ubuntu	48
Figura 5-18 RTT Máxima DNS Hurricane Electric Ubuntu	48
Figura 5-19 RTT Media DNS Hurricane Electric Ubuntu	49
Figura 5-20 Tiempos de resolución DNS Hurricane Electric Ubuntu	49
Figura 5-21 RTT Mínima Dominios Hurricane Electric Ubuntu	50
Figura 5-22 RTT Máxima Dominios Hurricane Electric Ubuntu	50
Figura 5-23 RTT Media Domimios Hurricane Electric Ubuntu	51
Figura 5-24 Resultado PingPlotter DNS VPN Wireguard	52
Figura 5-25 RTT Mínima DNS VPN Wireguard	52
Figura 5-26 RTT Máxima DNS VPN Wireguard	53
Figura 5-27 RTT Media DNS VPN Wireguard	53
Figura 5-28 Tiempo de resolución DNS VPN Wireguard	54
Figura 5-29 RTT Mínima Dominios VPN Wireguard	54
Figura 5-30 RTT Máxima Dominios VPN Wireguard	55
Figura 5-31 RTT Media Dominios VPN Wireguard	55
Figura 5-32 RTT Mínima Servicios NetScanTools/Ping	56
Figura 5-33 RTT Máxima Servicios NetScanTools/Ping	56
Figura 5-34 RTT Media Servicios NetScanTools/Ping	57

Figura 5-35 RTT Mínima Servicios PingPlotter/Mtr	57
Figura 5-36 RTT Máxima Servicios PingPlotter/Mtr	58
Figura 5-37 RTT Media Servicios PingPlotter/Mtr	58
Figura 5-38 Tiempos de resolución DNS Servicios	71
Figura 5-39 Tiempos Descargas Gran Volumen Servicios	71
Figura B-1 Servidor VPS	79
Figura B-2 Configuraciones VPS	79
Figura B-3 Ubicación VPS	80

Notación

IPv6	Internet Protocol versión 6
IPv4	Internet Protocol versión 6
RFC	Request For Comments
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
RIR	Regional Internet Registry
IANA	Internet Assigned Numbers Authority
ICMPv6	Internet Control Message Protocol version 6
VPN	Virtual Private Network
GRP	Global Route Prefix
SLAAC	StateLess Address Auto Configuration
ISP	Internet Service Provider
IGMP	Internet Group Management Protocol
ARP	Address Resolution Protocol
MTU	Maximum Transmission Unit
MAC	Media Access Control
NS	Neighbor Solicitation
NA	Neighbor Advertisement
RS	Router Solicitation
RA	Router Advertisement
DHCP	Dynamic Host Configuration Protocol
DAD	Duplicate Address Detection
MLD	Multicast Listener Discovery
DNS	Domain Name System
TTL	Time To Live
RTT	Round-trip time
VPS	Virtual Private Server
ISO	International Organization for Standardization

1 INTRODUCCIÓN Y OBJETIVOS

La evolución de Internet y su creciente demanda han provocado una escasez de direcciones IPv4.

Este protocolo, es utilizado desde los inicios de Internet, ofreciendo un espacio de direcciones de 32 bits que, en un primer momento, se consideró suficiente, ya que nadie imaginaba la inmensidad y la rápida extensión que alcanzaría este protocolo.

Con el paso del tiempo, la cantidad total de direcciones IPv4 disponibles fue disminuyendo considerablemente, temiendo una posible escasez de direcciones en un periodo corto de tiempo. Por esta razón, se planteó y desarrolló la idea de migrar hacia un sistema más escalable y eficiente, con una nueva versión del protocolo IP, conocida como IPv6.

El protocolo IPv6, definido en el RFC 2460, incorporó numerosas mejoras y resolvió el problema principal que estaba sometiendo al mundo moderno: la escasez de direcciones IPv4. No obstante, su adopción al mundo está siendo más lento de lo esperado, ya sea por el temor de interrumpir servicios existentes o simplemente por el acostumbramiento del mundo a una vida con IPv4. Ante esta situación, los profesionales se vieron obligados a desarrollar soluciones de transición para el protocolo IPv4 para combatir esa escasez de direcciones mientras la adopción de IPv6 se hiciera realidad.

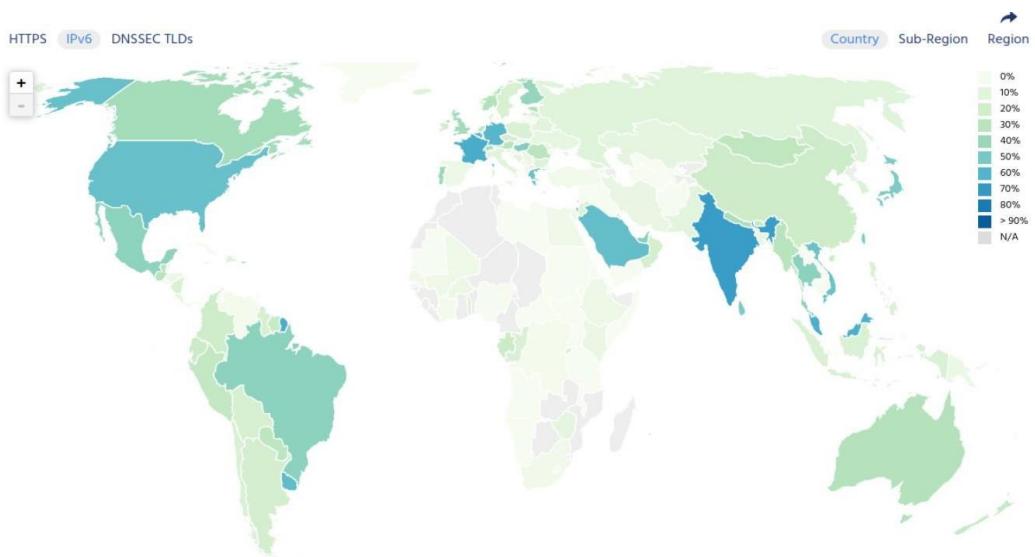


Figura 1-1 Impacto IPv6 global [1]

1.1 Objetivos

El objetivo principal de este proyecto es analizar el comportamiento de distintos servicios implementados para poder obtener conectividad IPv6 en redes que no disponen inicialmente de este protocolo de forma nativa por parte del proveedor de servicios. Los objetivos a tener en cuenta son los siguientes:

- Analizar el protocolo IPv6, sus características, novedades y su aplicación al mundo real.
- Analizar y estudiar las diferentes alternativas de acceso a IPv6, tales como túneles y VPN.
- Realizar pruebas de conectividad a los diferentes servicios de red utilizando herramientas

prácticas como Ping, dig, Wireshark, etc.

- Comparar los resultados obtenidos entre los distintos servicios, extrayendo conclusiones sobre su eficiencia, estabilidad, implementación y rendimiento

1.2 Metodología de trabajo

Para una organización adecuada, el proyecto lo hemos estructurado con una metodología basada en las siguientes fases:

- **Investigación:** Llevamos a cabo una investigación previa sobre los conceptos básicos y fundamentales necesarios para poder desarrollar el proyecto. Realizamos una investigación detallada del protocolo IPv6 y de los diferentes servicios para poder implementar a una red sin IPv6 nativa.

Asimismo, realizamos un estudio profundo y detallado de los distintos parámetros que afectan al rendimiento de la red, así como las herramientas profesionales utilizadas para poder realizar las mediciones correspondientes.

- **Implementación práctica:** Configuración e implementación de los distintos tipos de servicios para la obtención de IPv6.
- **Pruebas:** Realización de las pruebas para medir los distintos parámetros de red que afectan a cada servicio. Utilizamos software profesional para llevar a cabo las pruebas y obtener un análisis detallado.
- **Comparación:** Comparación de los resultados obtenidos para cada servicio implementado, evaluando su comportamiento para obtener el que ofrece mayor rendimiento.
- **Documentación:** Elaboración de un documento técnico que recoja todo el proceso seguido en el desarrollo del proyecto.

1.3 Estructura del Proyecto

Vamos a dividir el proyecto en los siguientes apartados:

- **Introducción:** Conceptos fundamentales que abordarán al proyecto, así como un estudio previo de la estructura y el desarrollo del proyecto.
- **IPv6:** Análisis en profundidad del protocolo IPv6, incluyendo los fundamentos teóricos, como su cabecera, direccionamiento, mejoras frente a IPv4, nuevos procedimientos, entre otros.
- **Servicios y herramientas:** Descripción y estudio previo de los servicios implementados para poder obtener conectividad IPv6, así como las herramientas empleadas para llevar a cabo el análisis.
- **Pruebas:** Análisis detallado de los resultados obtenidos en cada servicio implementado.
- **Comparaciones y futuro:** Conclusiones sobre el proyecto, realizando una valoración final y posibles mejoras para una extensión futura

2 INTERNET PROTOCOL VERSION 6 (IPv6)

En este apartado vamos a introducir el concepto del protocolo IPv6, clave para el desarrollo del proyecto. Explicaremos todo lo esencial de este protocolo, las principales diferencias respecto a su versión anterior, y ofreceremos una visión a futuro con una reflexión subjetiva del comportamiento que adoptará este protocolo en los próximos años.

2.1 Introducción

El protocolo IPv6 (Internet Protocol versión 6) es una nueva versión del protocolo IP (Interrnet Protocol), definida en el RFC 2460, diseñada para sustituir a IPv4 (Internet Protocol versión 4).

2.1.1 Cabecera IPv6

La cabecera del protocolo IPv6 tiene una extensión fija de 40 bytes, lo que beneficia al proceso de encaminamiento en los equipos intermedios. A diferencia de IPv4, cuya cabecera tiene una longitud variable, IPv6 elimina y modifica varios campos de IPv4, reduciendo la complejidad en el procesamiento de los paquetes y mejorando el rendimiento de la red.

Los principales cambios de la cabecera IPv6 con respecto a la cabecera IPv4 son los siguientes [2]:

- Eliminación de las opciones IP, sustituidas por las cabeceras de extensión. Este mecanismo nos permite añadir información adicional únicamente cuando sea necesario. Además, son procesadas por el destino, mejorando considerablemente el rendimiento de los nodos intermedios.
- Eliminación del tratamiento de la fragmentación, eliminándose los campos de Identificador, Flags y Offset presentes en IPv4. La fragmentación ahora es gestionada exclusivamente por el emisor, indicada a través de una cabecera de extensión.
- Eliminación de la suma de verificación (Checksum), asumiendo que las capas superiores ya realizan estas verificaciones de integridad.
- Eliminación del campo de longitud de la cabecera, pasando a ser innecesario por el tamaño fijo de 40 bytes que presenta la cabecera de IPv6.
- Uso de un alineado a 64 bits, ayudando a reducir la carga de procesamiento en los routers al reenviar paquetes.

La cabecera IPv6 se compone de la siguiente estructura [2]

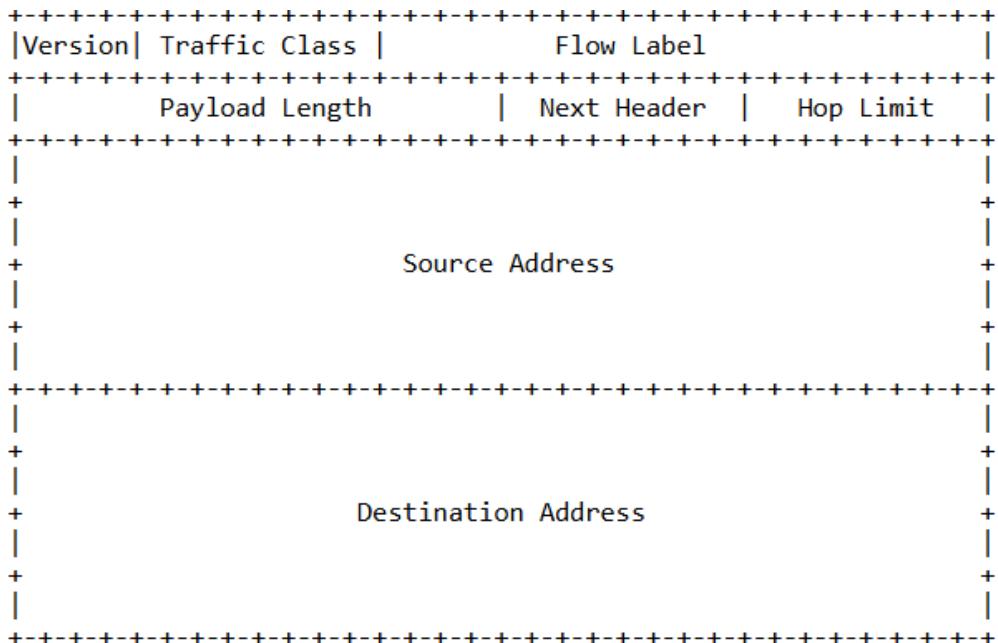


Figura 2-1 Cabecera IPv6 [2]

La función de estos campos es la siguiente:

- **Version:** Es un campo de 4 bits que nos indica la versión del Protocolo, en este caso la versión 6.
 - **Traffic Class:** Campo de 8 bits utilizado por los nodos de origen y/o por los routers para clasificar los paquetes IP según clases de tráfico o prioridades.
 - **Flow Label:** Es un campo de 20 bits que nos permite identificar los flujos de paquetes que requieren un tratamiento especial. Entendemos por flujo a un conjunto de paquetes que son tratados de la misma manera.
 - **Next Header:** Campo de 8 bits utilizado para indicar el siguiente protocolo que le sigue a la cabecera IPv6. A través de este campo, podemos indicar protocolos como TCP o UDP, de la capa de transporte, o bien podemos indicar una funcionalidad exclusiva de IPv6 llamadas cabeceras de extensión.
 - **Hop Limit:** Campo de 8 bits que representa el número máximo de saltos que un paquete puede realizar por la red antes de pasar a ser descartado. Empieza con un valor determinado que se decrementará con cada salto que realice por la red. Si su valor llega a 0, procederá a ser descartado, evitando así la aparición de bucles en la red.
 - **Source Address:** Campo de 128 bits que nos indica la dirección IPv6 del equipo.
 - **Destination Address:** Campo de 128 bits que nos indica la dirección IPv6 del equipo destino al que queremos enviar el paquete.

2.1.2 Direccionamiento IPv6

El direccionamiento IPv6 constituye una versión mejorada respecto al protocolo IPv4, ampliando el espacio de direcciones disponible. Sus direcciones pasan a tener una longitud de 128 bits, lo que permite una cantidad de direcciones prácticamente ilimitada [3].

2.1.2.1 Formato General

Estas direcciones están divididas en tres campos de longitud variable. Estos campos son [4]:

- **Global Routing Prefix (GRP):** Es el prefijo de red asignado a una organización o proveedor. Identifica la red de manera global.

- **ID Subred:** Es el rango de dirección utilizado para la creación de subredes.
- **ID Interfaz:** Identifica de manera única una interfaz en el enlace. Es el rango de direcciones que puede tomar un equipo dentro de la subred establecida.

Una dirección IPv6 se compone de 8 bloques de 16 bits expresados en formato hexadecimal y separados por el carácter “：“. Un ejemplo de dirección IPv6 sería **2001:0000:130F:0000:0000:09AB:234a:1231**.

La particularidad de estas direcciones es que se pueden simplificar en ciertos momentos. Existen varias reglas para aplicar la simplificación de direcciones. Estas son:

- **Simplificación de ceros a la izquierda:** Los ceros iniciales de cada bloque son opcionales y pueden ser simplificados.
- **Omisión de bloques consecutivos de ceros:** Varios bloques sucesivos a 0 pueden ser sustituidos por “::”, aunque este procedimiento solo puede realizarse una única vez por dirección.

Siguiendo las reglas descritas anteriormente, la dirección anterior se puede simplificar de la siguiente manera:

2001:0:130F::9AB:234a:1231.

2.1.2.2 Tipos de dirección

A diferencia de IPv4, una misma interfaz en IPv6 podemos encontrar varios tipos de direcciones asignadas. Podemos clasificar los tipos de direcciones en tres grandes grupos. Estos son [4]:

- **Unicast:** Son identificadores para una interfaz en particular. Designa una interfaz de un nodo.
- **Anycast:** Es una dirección o un identificador asignado a varias interfaces de equipos distintos. Un paquete enviado a esta dirección será recibido por el nodo más cercano. Son direcciones que no deben usarse como dirección origen ni por un sistema final. Su uso es en parte experimental.
- **Multicast:** Son direcciones asignadas a varias interfaces de equipos distintos. El paquete con destino esta dirección será entregado a todas las interfaces que posean esta dirección.

Dentro de las direcciones Unicast nos encontramos con los siguientes tipos de direcciones:

- **Globales:** Son el tipo de dirección utilizada para el encaminamiento a través de Internet. Esta dirección está dividida en tres campos:
 - **GRP (Global Route Prefix):** El prefijo identifica el tipo de dirección.
 - **SLA (Site-Level Aggregator):** Usada por el poseedor del GRP para organizar su red jerárquicamente en distintas subredes.
 - **Interface-ID:** Identifica interfaces en un enlace de manera única.
- **Local a la ubicación:** Son unas direcciones en el rango **FEC0::/10**. Son similares a las direcciones privadas de IPv4 (192.168.x.x). Además, no deben ser enrutasadas en Internet. Sin embargo, se han convertido en direcciones obsoletas y se han reemplazado por las direcciones localmente únicas.
- **Local al enlace:** Son direcciones en el rango **FE80::/10**, las cuales son obligatorias y no deberían aparecer en Internet. Son utilizadas para funciones como la autoconfiguración o para el descubrimiento de vecinos, entre otras funciones.
- **Localmente única:** Estas direcciones están en el rango **FC00::/8** o **FD00::/8**. Son direcciones privadas no enruteables en Internet, pero válidas en áreas reducidas. Además, son direcciones independientes de la operadora.

Las direcciones Multicast en IPv6 se encuentran dentro del rango **FF00::/8**. A diferencia de IPv4, donde se utilizaba la difusión para comunicarse con todos los equipos que estuvieran conectados a la red en ese

momento, IPv6 sustituye este mecanismo por el uso de direcciones multicast específicas, haciendo que los paquetes solo sean recibidos por un grupo de equipos y no por la red completa como ocurre en IPv4 a través del protocolo ARP.

Ciertas funciones específicas realizadas por el protocolo IPv6 se realizan mediante direcciones multicast reservadas, siendo de gran utilidad para distintos procedimientos. Las más importantes son:

- **FF01::1**: Dirigida a todas las interfaces del nodo.
- **FF02::1**: Dirigida a todos los nodos del enlace.
- **FF02::2**: Enviada a todos los routers del enlace.
- **FF02::FFXX:XXXX**: Dirección que solo escucha un nodo en concreto.

Además, tenemos direcciones especiales que van a sernos útiles en distintos momentos:

- **::/0**: La dirección **::/0** es utilizada cuando el equipo no se sabe cual es su dirección propia. No debe asignarse a ninguna interfaz ni debe ser encaminada por routers.
- **::1**: La dirección **::1** es la dirección de loopback o autodirección, equivalente a la **127.0.0.1** en IPv4. Es utilizada por los equipos para referirse a ellos mismos.

2.1.3 Subredes

La autoridad principal responsable de la asignación de direcciones IP es la IANA (Internet Assigned Numbers Authority), la cual gestiona el rango de direcciones globales enrutables en Internet definido por el prefijo **2000::/3**. Este organismo se encarga de asignar prefijos del tipo /23 a los diferentes RIR (Regional Internet Registries), que son organismos regionales responsables de asignar bloques de direcciones IPv6 a ISP o clientes grandes en su proporción [4].

Cada RIR asigna bloques de /32 a cada ISP o clientes grandes en concreto. Estos a su vez distribuyen estos bloques a todos sus clientes, recibiendo como mínimo un prefijo /48. Cada cliente con este bloque asignado, puede dividir este espacio asignado en bloques de /64, formando así múltiples subredes, permitiendo hasta un total de 2^{16} subredes (65536 subredes).

De esta forma, cada subred tiene un prefijo /64. Aunque técnicamente es posible generar un prefijo mayor a /64, se recomienda usar /64 para todas las redes que contengan hosts. Esto permite utilizar la autoconfiguración sin estado (SLAAC) y muchas otras funcionalidades de IPv6 diseñadas para trabajar sobre este tamaño.

2.2 ICMPv6

El protocolo ICMPv6 es una versión mejorada del protocolo ICMP del protocolo IPv4, incorporando múltiples funciones adicionales, reemplazando a protocolos necesarios en IPv4 como puede ser el protocolo ARP para temas de difusión o IGMP para gestionar direcciones multicast.

ICMPv6 permite informar tanto de errores que hayan sucedido durante la transmisión de datos a través de la red como mensajes con un carácter informativo.

El formato del mensaje ICMP tiene la siguiente estructura [5]:

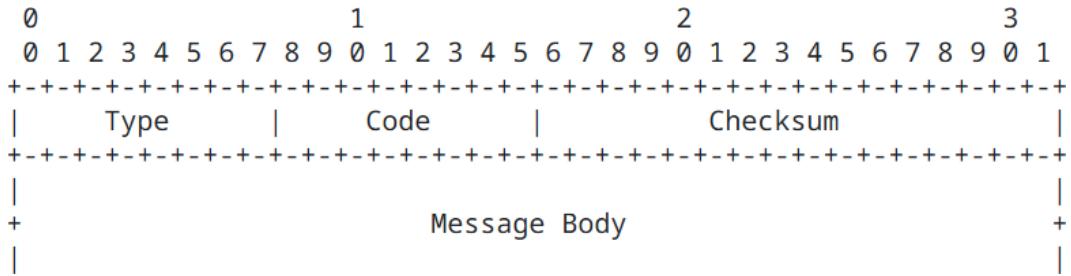


Figura 2-2 Cabecera ICMPv6 [5]

Cada campo contiene lo siguiente:

- **Type**: Es un campo de 8 bits utilizado para indicar el tipo de mensaje ICMP que se va a transmitir. Por ejemplo, el campo type con valor 128 indica un mensaje Echo Request.
- **Code**: Campo de 8 bits que contiene información adicional específica para el tipo de mensaje que se está transmitiendo.
- **Checksum**: Campo de 16 bits encargado de la detección de errores.
- **Message Body**: Campo de longitud variable que contiene el contenido del mensaje.

Para los mensajes que indican un error ocurrido tenemos la siguiente clasificación:

- **Destination Unreachable**: Indica que el destino es inalcanzable por alguna razón.
- **Packet Too Big**: Indica que el paquete es demasiado grande como para ser transmitido por el enlace.
- **Time exceed**: El paquete ha excedido el número máximo de saltos permitido.
- **Parameter Problem**: Algún parámetro erróneo en el encabezado del paquete IPv6 o cabecera de extensión.

Para los mensajes con carácter informativo tendremos la siguiente clasificación de los mensajes más importantes:

- **Echo Request**: Mensaje de petición enviado por el emisor para pruebas de conectividad.
- **Echo Reply**: Respuesta al mensaje Echo Request por parte del receptor, verificando la conectividad entre dos equipos.
- **Multicast Listener Discovery**: Gestiona el registro de oyentes multicast.
- **Router Solicitation / Router Advertisement**: Permiten a los equipos descubrir los routers que están disponibles en la red.
- **Neighbor Solicitation / Neighbor Advertisement**: Mensajes para descubrir vecinos y resolver sus direcciones MAC.
- **Redirect**: Mensaje para especificar a un host un camino alternativo hacia el destino más eficiente.

2.2.1 Neighbor Discovery

Utilizamos los siguientes mensajes ICMPv6 [6]:

- **Router Solicitation**: Mensaje enviado por un host solicitando que un router vecino se anuncie en la red.

- **Router Advertisement:** Mensaje de respuesta del router, confirmando su presencia y proporcionando distintos parámetros necesarios para la configuración, como el prefijo de red y la MTU.
- **Neighbor Solicitation:** Mensaje enviado por un host preguntando por la dirección MAC del equipo destino. Sustituye al protocolo ARP de IPv4. También se puede utilizar por el host para comprobar si su dirección configurada está duplicada o si un nodo vecino sigue siendo accesible.
- **Neighbor Advertisement:** Respuesta del equipo destino al mensaje NS, proporcionando su dirección MAC o respondiendo, por ejemplo, a un mensaje NS de comprobación de dirección duplicada.
- **Redirect:** Mensaje enviado por un router proporcionando a un host un camino alternativo óptimo para un destino en concreto.

2.2.2 SLAAC

El procedimiento SLAAC es empleado por los hosts para configurar sus direcciones de manera automática, sin la necesidad de un servidor DHCP [7].

El equipo arranca sin dirección IPv6, por lo que utiliza la dirección especial “::” para enviar un mensaje RS solicitando la presencia de algún router vecino en la red. Si existe alguno, este responderá con un mensaje RA, indicando varios parámetros claves para el proceso, como el prefijo de red y la MTU.

A partir del prefijo de red, el host puede realizar este procedimiento utilizando varios mecanismos. El más común es el mecanismo EUI-64 utilizado para configurar una dirección IP a través de su dirección MAC. El procedimiento es el siguiente [4]:

- El equipo visualiza su dirección MAC y le añade el valor FFFE en el centro.
- Complementa el bit u/g.
- El resultado es una dirección de 64 bits que es justamente la necesitada para formar el identificador de interfaz.

De esta forma, el equipo configura una dirección IPv6 de manera automática, sin tener que depender de un servidor DHCP externo u otros procedimientos. Simplemente con el prefijo de red anunciado por el router vecino y aplicando el método EUI-64 se obtiene una dirección IPv6.

2.2.3 DAD

Al aplicar el mecanismo SLAAC para autoconfigurar sus direcciones, puede ocurrir de que pueda existir un equipo con sus mismas direcciones IP, ya sea por una configuración manual o por una dirección proporcionada por un servidor DHCP [4].

Para evitar esta duplicidad, el equipo al terminar de configurar su dirección aplica el procedimiento DAD, enviando un mensaje NS con su propia dirección como destino y la dirección especial por defecto como origen.

Si no recibe respuesta, se considera que la dirección es única en la red y por tanto la autoconfiguración se válida. En cambio, si recibe como respuesta un mensaje NA, el equipo origen deberá descartar esa dirección ya que estaría siendo duplicada.

2.2.4 Path MTU Discovery

El descubrimiento de MTU es un mecanismo que permite que los paquetes se ajusten a la MTU mínima a lo largo de todo el camino hacia el destino. Cuando el paquete es enviado por el origen y el router que lo recibe no puede reenviarlo debido a que su tamaño excede la MTU, responde con un mensaje ICMPv6 indicando el error Packet Too Big. Este error incluye un campo adicional en el que se especifica la MTU máxima del enlace. El origen al recibir el mensaje recibe también la nueva MTU, por lo que puede ajustar el tamaño de sus paquetes, evitando así la fragmentación [8].

2.2.5 Multicast Listener Discovery (MLD)

Mecanismo equivalente al protocolo IGMP de IPv4, utilizado para permitir que los routers descubran los hosts que están escuchando en un grupo multicast determinado.

MLD actúa como intermediario entre el host y el router, permitiendo la inscripción de los diferentes hosts a grupos multicast. De esta forma, los routers pueden enviar el tráfico multicast únicamente a los hosts que lo han solicitado y, por tanto, interesados en recibir este tráfico multicast [9].

2.3 Seguridad IPv6

Una de las principales mejoras introducidas por IPv6 es la inclusión obligatoria de IPSec (Internet Protocol Security). Es una familia de protocolos utilizado para encriptar las comunicaciones a través de una red. Estos protocolos actúan en la capa de red, permitiendo dar soporte a protocolos de capa 4, como TCP y UDP [10].

Las principales características de esta familia de protocolos son las siguientes:

- **Protocolos de seguridad:** Con la ayuda de estos protocolos, IPSec puede garantizar la autenticidad e integridad de los datos. Para ello, IPSec se basa en dos protocolos, que son:
 - AH (Authentication Header): Se encarga de la autenticación e integridad de los datos, pero no maneja el cifrado. Garantiza que los paquetes no hayan sido modificados por los nodos intermedios [11].
 - ESP (Encapsulating Security Payload): Se encarga de proporcionar el cifrado adecuado de los datos y, opcionalmente, autenticación y protección de integridad [12].
- **Administración de claves:** La gestión de claves en IPSec se realiza principalmente a través del protocolo IKE (Internet Key Exchange). Este protocolo es utilizado principalmente para la gestión del cifrado ESP. Ambos extremos de la comunicación negocian de manera segura todos los parámetros y algoritmo de cifrado para garantizar la seguridad de la transmisión [13].
- **Modos de operación:** Esta familia de protocolos ofrece soporte para dos modos de operación:
 - Modo transporte: El modo transporte es el modo en el cual dos puntos finales están conectados directamente. Es utilizado para cifrar la carga útil del paquete IP.
 - Modo túnel: El modo túnel es el modo en el cual se crea una conexión entre dos redes IP. En este modo se encapsula todo el paquete IP original dentro de un nuevo paquete IP. Es muy utilizado en servicios como VPNs.
- **Algoritmos de Encriptación:** Se recomiendan los algoritmos de protección más recomendados para los siguientes protocolos [14]:
 - Para ESP se recomiendan algoritmos de cifrado como AES-GCM, AES-CBC, AES-CTR, TripleDES-CBC, HMAC-SHA1-96, AES-GMAC con AES-128 o AES-XCBC-MAC-96. Se

recomienda evitar el uso del algoritmo de cifrado DES-CBC.

- Para AH se recomienda usar algoritmos de cifrado como AES-GCM, AES-GMAC, TripleDES-CBC, AES-XCBC-MAC-96 o AES-CTR. En cambio, se recomienda evitar el algoritmo DES-CBC.

2.4 Diferencias IPv6 vs IPv4

Para terminar el estudio de IPv6, vamos a comentar las diferencias más notorias entre el protocolo IPv6 y el protocolo IPv4, destacando todas sus ventajas y desventajas. Para ello, empleamos una tabla donde se recoge toda la información necesaria para este estudio:

Característica	Ipv4	IPv6
Longitud Dirección	Direcciones de 32 bits	Direcciones de 128 bits
Seguridad	IPSec opcional	Uso de IPSec obligatorio
Cabecera	Longitud variable	Longitud fija de 40 bytes
Resolución Direcciones	Utiliza protocolo ARP	Utiliza Neighbor Discovery Protocol
Fragmentación	No aconsejable. Se realiza tanto en routers como en hosts	No aconsejable. Host de origen encargado de realizarlo
NAT	Si, debido a escasez de direcciones	No necesario
MTU	Mínimo 576 bytes. Si se supera MTU, se recurre a fragmentación	Mínimo 1280 bytes. Si se supera MTU, se aplica mecanismo Path MTU Discovery.
Gestión grupos Multicast	A través de protocolo IGMP	A través de MLD
DNS	Utiliza registros de tipo A y PTR para búsquedas inversas	Utiliza registros de tipo AAAA y PTR para búsquedas inversas
Configuración direcciones	Manual o a través de diferentes protocolos como DHCP	Manual, protocolos como DHCP o autoconfiguración (SLAAC)
Anycast	No definido explícitamente, pero puede implementarse	Soportado de manera nativa

Tabla 2-1 Diferencias IPv6 vs IPv4

Como podemos apreciar en la Tabla 2-1, IPv6 mejora bastantes aspectos de IPv4, destacando la seguridad, escalabilidad y eficiencia en la red. Sin embargo, su adopción al mundo está todavía en progreso, aunque es algo que llegará en un futuro no tan lejano.

2.5 Visión Futura

En proporción, la creación de IPv6 surgió como respuesta a la creciente demanda de direcciones IPv4. Al ver esta situación tomaron la decisión de crear una nueva versión del protocolo IP, asegurándose así

de combatir esta escasez de direcciones que ya estaba sometiendo al mundo entero.

Este protocolo fue bautizado como IPv6, el cual nació como una versión para reemplazar a IPv4 trayendo consigo nuevas funcionalidades y combatiendo el problema principal que era la escasez de direcciones. Miles de proveedores de Internet, acordaron que IPv6 fuese el nuevo estándar sobre el 2012, aunque es un proceso que no llegó a terminarse. Sin embargo, la demanda de IPv4 sigue creciendo hoy en día y su capacidad cada vez está más cerca de llegar al límite.

Para retardar este límite, se han aplicado numerosas soluciones. Una de ellas es el famoso procedimiento NAT, el cual hace que los equipos tengan una dirección privada para circular por la red LAN, pero al salir a Internet se realice una traducción de direcciones haciendo que su dirección IP privada se reemplace por la dirección IP pública de su router.

Estas medidas no serían necesarias con IPv6, ya que ofrece una cantidad de direcciones que es prácticamente ilimitada, por lo que muchas personas se cuestionan y debaten el trasladarse definitivamente a IPv6.

3 MÉTODOS DE ACCESO A IPv6

Hoy en día, muchas compañías de red todavía no ofrecen direcciones IPv6 globales, ofreciendo únicamente una dirección IPv4 pública. Esta limitación provoca que, si necesitamos IPv6, tengamos que utilizar algunos servicios que nos proporcionen un direccionamiento IPv6, o bien contactar con el proveedor de red para que nos proporcione un rango de direcciones.

El direccionamiento IPv6 proporcionado por la operadora recibe el nombre de IPv6 nativo, el cual es un tipo de conectividad directa y normalmente su uso tiene un mejor rendimiento que usando otro tipo de servicios.

En cuanto a servicios que nos ofrecen direccionamiento IPv6, existen los túneles, los cuales desempeñan un papel muy importante al permitir la comunicación entre redes IPv4 e IPv6. Su procedimiento consiste en encapsular paquetes IPv6 dentro de paquetes IPv4 para que este pueda viajar a través de una infraestructura que sea solo de IPv4. Su principal inconveniente, como se puede deducir, es la bajada de rendimiento comparada con el direccionamiento IPv6 ofrecido por la operadora. Sin embargo, es una solución bastante buena y asequible hoy en día.

3.1 Conectividad IPv6 Nativa

Como hemos comentado, la conectividad IPv6 nativa surge del ofrecimiento por parte de la operadora de red de un rango de direcciones IPv6 globales, permitiendo que la red tenga la posibilidad de poder acceder a Internet a través de estas direcciones IPv6 globales, teniendo una conectividad directa y bastante estable.

3.1.1 Funcionamiento

Su funcionamiento por la red se basa en el siguiente procedimiento:

- En primer lugar, el equipo origen obtiene una dirección IPv6 a través de procedimientos como SLAAC (Stateless Address Autconfiguration), configuración manual o a través de servidores DHCPv6.
- Una vez establecida la dirección, para acceder a dominios como Google.com, el equipo envía una solicitud a su servidor DNS configurado, el cual la resuelve, devolviendo su dirección IPv6 asociada.
- El equipo origen construye el paquete con las direcciones IPv6 correspondientes y lo envía por la red.
- El paquete IPv6 viaja por la red de la operadora y posteriormente por la red global, atravesando distintos routers que entienden IPv6.
- Una vez llega al destino, el equipo final procesa el paquete y genera un paquete IPv6 de respuesta.
- Este paquete IPv6 es recibido por el equipo origen, que lo procesa y muestra la respuesta correspondiente.

La ventaja de este plan de direccionamiento es la posibilidad de no usar túneles ni encapsulación, por lo que todo el tráfico viajará directamente por la red IPv6, obteniendo así una latencia mucho menor, además de una gran eficiencia y estabilidad.

A través de la herramienta Wireshark, podemos observar los paquetes de red para su posterior análisis:

```
> Frame 30: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{1D3C1773-802F-4945-B8A7-7998D5CA532A}, id 0
> Ethernet II, Src: MicroStarINT_89:ed:23 (00:08:61:89:ed:23), Dst: zte_97:0d:14 (c0:51:5c:97:0d:14)
> Internet Protocol Version 6, Src: 2a0c:5a84:7807:bc00:2dbf:a8eb:305e:4baa (2a0c:5a84:7807:bc00:2dbf:a8eb:305e:4baa), Dst: releases.ubuntu.com (2620:2d:4000:1::1a)
    0110 .... = Version: 6
> .... 0000 0000 .... .... .... .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
.... 1011 0010 1101 1101 1101 = Flow Label: 0xb2ddd
Payload Length: 20
Next Header: TCP (6)
Hop Limit: 64
> Source Address: 2a0c:5a84:7807:bc00:2dbf:a8eb:305e:4baa (2a0c:5a84:7807:bc00:2dbf:a8eb:305e:4baa)
> Destination Address: releases.ubuntu.com (2620:2d:4000:1::1a)
[Stream index: 2]
> Transmission Control Protocol, Src Port: 50577, Dst Port: 443, Seq: 1, Ack: 1, Len: 0
```

Figura 3-1 Paquete IPv6 Nativo

3.1.2 Configuración Servicio Nativo

En este proyecto para realizar el estudio sobre los distintos servicios que ofrecen IPv6, utilizaremos una red doméstica la cual contiene IPv6 nativo ofrecido por parte de la operadora de red. Como dirección IP tendremos la dirección **2a0c:5a84:7201:ce00:bb30:7323:aedb:13e3**, obtenida a través de un servidor DHCP configurado en el router. Además, realizaremos el estudio de esta red probando diferentes tipos de servidores DNS para obtener la mayor velocidad y el mejor tiempo de respuesta posible.

Utilizando el comando **ipconfig**, podemos ver la configuración de la interfaz de red a través de la siguiente imagen:

Figura 3-2 Interfaz Servicio Nativo

Una vez configurado y establecido el servicio, podemos corroborar que la red contiene conectividad IPv6 a través de distintas webs. Podemos ver su conectividad a través de la siguiente imagen:



¿Tu conexión está preparada para el Internet del futuro?

✓ Sí, parece que ya utilizas IPv6.

¡Te damos la bienvenida al Internet del futuro!

Obtén más información sobre el [protocolo IPv6](#) o sobre el [Día mundial del IPv6](#).

Google

Figura 3-3 Conectividad IPv6 Nativo

3.2 Túneles IPv6 sobre IPv4

Cuando la operadora de red no nos proporciona un plan de direccionamiento IPv6 de manera nativa, una gran alternativa a la que podemos recurrir es la de utilizar túneles, cuya función es encapsular paquetes IPv6 dentro de paquetes IPv4. Esta solución permite un rendimiento más que aceptable, proporcionando una buena experiencia al usuario.

3.2.1 TunnelBroker (Hurricane Electric)

TunnelBroker de Hurricane Electric [16] es uno de los servicios más populares y utilizados hoy en día, ya que se trata de un servicio gratuito que permite a cualquier usuario de una red que disponga de una dirección IPv4 pública libre de CG-NAT, poder obtener una dirección IPv6 global, aunque su operadora no se la proporcione de manera nativa.

3.2.1.1 Funcionamiento

El servicio consta de la creación de un túnel IPv6 sobre IPv4, utilizando el protocolo 41, facilitando el acceso a la red IPv6 de manera global. Su funcionamiento una vez realizado la configuración correspondiente es el siguiente

[16] [17]:

- Una vez realizada la configuración del túnel en el equipo origen, el usuario recibirá un prefijo /64 junto con una dirección IPv6 global, teniendo la posibilidad de poder acceder a Internet.
- Cuando el usuario quiere acceder a un dominio como Google.com, el equipo genera un paquete IPv6 para resolver el dominio mediante la consulta a un servidor DNS, el cual le proporciona la dirección IPv6 destino correspondiente.
- El paquete IPv6 generado para acceder a Google.com por medio de IPv6, se encapsula dentro de un paquete IPv4 usando el protocolo 41. La dirección IPv4 origen es la del equipo origen y la dirección destino la IPv4 del otro extremo del túnel, el cual será el servidor de Hurricane Electric.
- El paquete encapsulado es enviado por el túnel, el cual viaja como un paquete IPv4 normal con la diferencia de que dentro lleva encapsulado un paquete IPv6.

- Una vez recibido por el extremo del túnel, el servidor desencapsula el paquete, quedándose con el paquete IPv6 original.
- Este paquete IPv6 es enviado por la red IPv6 global hasta alcanzar su destino.
- Una vez el destino recibe el paquete, envía su respuesta a través de un paquete IPv6 dirigido al servidor de Hurricane Electric.
- El servidor vuelve a encapsular el paquete IPv6 de respuesta en un paquete IPv4, reenviándolo de vuelta al usuario por el túnel.
- Una vez llega al equipo, desencapsula el paquete IPv4 recibido, recuperando el paquete IPv6 de respuesta para su posterior procesamiento.

Como podemos intuir, es un proceso más complejo que el plan de direccionamiento nativo, ya que se deben de utilizar procedimientos como la encapsulación del paquete Ipv6 en un nuevo paquete IPv4, además de un mayor número de saltos intermedios para poder llegar al destino. Esto se traduce en una subida en parámetros como la latencia y la pérdida de paquetes, pero sigue siendo una solución bastante precisa y eficiente.

A través de la herramienta Wireshark podemos analizar los paquetes enviados por la red:

```
> Frame 18: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface \Device\NPF_{49A0D2ED-FD5D-4A8B-81FF-7AF14D919879}, id 0
> Ethernet II, Src: HuaweiTechno_ca:01:be (90:17:3f:ca:01:be), Dst: Intel_74:5f:6d (4c:1d:96:74:5f:6d)
  Internet Protocol Version 4, Src: tserv1.lon1.he.net (216.66.80.26), Dst: 192.168.18.5 (192.168.18.5)
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 80
      Identification: 0xf168 (61800)
    > 010. .... = Flags: 0x2, Don't fragment
      ...0 0000 0000 0000 = Fragment Offset: 0
      Time to Live: 246
      Protocol: IPv6 (41)
      Header Checksum: 0x9811 [validation disabled]
        [Header checksum status: Unverified]
      Source Address: tserv1.lon1.he.net (216.66.80.26)
      Destination Address: 192.168.18.5 (192.168.18.5)
        [Stream index: 0]
    > Internet Protocol Version 6, Src: releases.ubuntu.com (2620:2d:4000:1::1a), Dst: tunnel955780-pt.tunnel.tserv5.lon1.ipv6.he.net (2001:470:1f08:3e9::2)
    > Transmission Control Protocol, Src Port: 443, Dst Port: 64102, Seq: 1, Ack: 301, Len: 0
```

Figura 3-4 Paquete IPv4 Hurricane Electric

```
> Frame 18: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface \Device\NPF_{49A0D2ED-FD5D-4A8B-81FF-7AF14D919879}, id 0
> Ethernet II, Src: HuaweiTechno_ca:01:be (90:17:3f:ca:01:be), Dst: Intel_74:5f:6d (4c:1d:96:74:5f:6d)
> Internet Protocol Version 4, Src: tserv1.lon1.he.net (216.66.80.26), Dst: 192.168.18.5 (192.168.18.5)
  Internet Protocol Version 6, Src: releases.ubuntu.com (2620:2d:4000:1::1a), Dst: tunnel955780-pt.tunnel.tserv5.lon1.ipv6.he.net (2001:470:1f08:3e9::2)
    0110 .... = Version: 6
    .... 0000 0000 .... .... .... .... .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
      .... 1011 1001 0011 0110 0011 = Flow Label: 0xb9363
    Payload Length: 20
    Next Header: TCP (6)
    Hop Limit: 59
    > Source Address: releases.ubuntu.com (2620:2d:4000:1::1a)
    > Destination Address: tunnel955780-pt.tunnel.tserv5.lon1.ipv6.he.net (2001:470:1f08:3e9::2)
      [Stream index: 2]
    > Transmission Control Protocol, Src Port: 443, Dst Port: 64102, Seq: 1, Ack: 301, Len: 0
```

Figura 3-5 Paquete Ipv6 Hurricane Electric

3.2.1.2 Configuración Servicio Hurricane Electric

Para el servicio TunnelBroker de Hurricane Electric, la red debe estar libre de la restricción impuesta por muchas compañías de red, llamada CG-NAT. Esta restricción hace que tu red no disponga de una dirección IPv4 pública que sea exclusivamente de tu propiedad, por lo que será compartida por miles de usuario realizando NAT en el router del operador. Al realizar este procedimiento, los servicios externos no pueden ver tu dirección desde fuera, por lo que servicios como este de Hurricane Electric resulta imposible instalarlos teniendo esta restricción.

Sin embargo, para librarse de esta restricción, podemos contactar con nuestra operadora de red para que a

través de una subida de precio te liberen de esta restricción y te proporcionen una dirección IPv4 pública y exclusiva.

Para este proyecto, la red sin IPv6 tenía impuesta esta restricción, por lo que resultaba imposible la instalación del servicio. Afortunadamente, comentando la situación con el operador de red, a través de un pago extra la red fue liberada de la restricción impuesta.

Para la instalación y configuración del servicio, seguimos los siguientes pasos:

- Para la instalación del servicio Hurricane Electric, en primer lugar, debemos acceder a la web oficial de Hurricane Electric. Una vez dentro elegimos la opción **Free IPv6 Tunnel Broker** donde tendremos que registrarnos.
- Posteriormente, una vez realizado el registro, habrá que acceder a la pestaña **Create Regular Tunnel** donde habrá que especificar tu dirección IPv4 pública y la zona geográfica donde quieras que se aloje.
- Finalmente, una vez introducido estos valores, se crea el túnel. Una vez realizado la creación, el servicio nos proporciona varios parámetros muy interesantes para poder configurar el túnel en tu equipo.
- Para la configuración del servicio, el usuario tiene que seleccionar la pestaña **Example Configurations** y seleccionar el sistema operativo donde desee configurar el servicio.
- Una vez seleccionado, mostrará una serie de comandos para que el usuario ejecute en su equipo y poder así completar la instalación de este servicio.

Tras el registro y la creación del túnel, obtuvimos varios parámetros de configuración claves para poder obtener conectividad IPv6. Como dirección IP, el túnel nos ofrece la dirección IPv6 **2001:470:1f08:3e9::2/64** [16].

El servidor de Hurricane Electric o el extremo remoto del túnel, utiliza la dirección IPv6 **2001:470:1f08:3e9::1/64**

y como dirección IPv4 la **216.66.80.26**.

Además, nos proporciona un prefijo de red el cual nos permite la posibilidad de asignar direcciones IPv6 a otros equipos interesados en obtener un plan de direccionamiento IPv6.

Para la resolución de dominios, nos ofrece varios servidores DNS, uno de ellos utilizando IPv6 y el otro IPv4. El servidor DNS que utiliza IPv6 utiliza la dirección **2001:470:20::2**. Para el caso de IPv4, el servidor utiliza la dirección **74.82.42.42**.

Por tanto, el conjunto de todos los parámetros nos permite implementar una red IPv6. Esta red nos permite gozar de un plan de direccionamiento IPv6, pudiendo acceder de manera completa a la red global IPv6 con un excelente rendimiento.

Para poder ver la configuración de la interfaz de red, utilizamos el comando **ipconfig**:

```
Adaptador de túnel IP6Tunnel:  
  
Sufijo DNS específico para la conexión. . . :  
Dirección IPv6 . . . . . : 2001:470:1f08:3e9::2  
Vínculo: dirección IPv6 local. . . : fe80::af0a:40e:a751:26a7%63  
Puerta de enlace predeterminada . . . . : 2001:470:1f08:3e9::1
```

Figura 3-6 Interfaz Hurricane Electric



Tunnel Details

IPv6 Tunnel		Example Configurations	Advanced
<input type="checkbox"/> Tunnel ID:	955780	<input type="button" value="Delete Tunnel"/>	
<input type="checkbox"/> Creation Date:	Mar 22, 2025		
<input type="checkbox"/> Description:			
IPv6 Tunnel Endpoints			
<input type="checkbox"/> Server IPv4 Address:	216.66.80.26		
<input type="checkbox"/> Server IPv6 Address:	2001:470:1f08:3e9::1/64		
<input type="checkbox"/> Client IPv4 Address:	185.193.172.15		
<input type="checkbox"/> Client IPv6 Address:	2001:470:1f08:3e9::2/64		
Routed IPv6 Prefixes			
<input type="checkbox"/> Routed /64:	2001:470:1f09:3ea::/64		
<input type="checkbox"/> Routed /48:	Assign /48		
DNS Resolvers			
<input type="checkbox"/> Anycast IPv6 Caching Nameserver:	2001:470:20::2		
Anycast IPv4 Caching Nameserver:	74.82.42.42		
DNS over HTTPS / DNS over TLS:	ordns.he.net		
rDNS Delegations			
<input type="checkbox"/> rDNS Delegated NS1:	Edit		
rDNS Delegated NS2:			
rDNS Delegated NS3:			
rDNS Delegated NS4:			
rDNS Delegated NS5:			

Figura 3-7 Configuraciones Hurricane Electric [16]

Para el servidor de Hurricane Electric implementado en Windows, podemos afirmar la conectividad a través de la siguiente imagen:

[Prueba IPv6](#) [FAQ](#) [Mirrors](#) [estadísticas](#)

Probar tu conectividad IPv6. A 5

Sumario	Pruebas ejecutadas	Compartir Resultados / Contactar	Otros Sitios IPv6	Para el Servicio de Asistencia
<ul style="list-style-type: none"> i Su dirección IPv4 en la Internet parece ser 185.193.172.15 (FIBERPLUS2AS) i Su dirección IPv6 en la Internet parece ser 2001:470:1f08:3e9::2 (HURRICANE) i Puesto que tienes IPv6, estamos incluyendo una ficha que muestra otros sitios IPv6 y cuán bien puede alcanzarlos. [más información] ✓ Tu servidor DNS (posiblemente controlado por tu ISP) parece tener acceso a Internet IPv6. <p style="text-align: center;">Tu puntuación de preparación</p> <p style="text-align: center;">10/10</p> <p style="text-align: center;">para su estabilidad y preparación de IPv6, cuando editores estén obligados a usar sólo IPv6</p> <p>Click para ver Datos de prueba</p> <p>(Actualizando estadísticas de la preparación IPv6 del lado del servidor)</p> <p>This instance (main.test-ipv6.com) is hosted at Linode.</p> <p>Copyright (C) 2010-2024 Jason Fesler. Todos los derechos reservados. Versión 1.1.1012 (8:de716) Mirrors Fuente Email Atribuciones Deja un comentario Res. ES 95.94% Este es un espejo de test-ipv6.com. Las opiniones expresadas aquí pueden no reflejar la opinión del dueño del espejo.</p>				

Figura 3-8 Conectividad IPv6 Hurricane Electric Windows [15]

Como vemos en la Figura 3-8, podemos afirmar que efectivamente se establece la conectividad en la red, mostrándonos la dirección IPv6 asignada y su ubicación, la cual nos indica que es de Hurricane.

Para el mismo servicio de Hurricane pero implementado en el sistema operativo Ubuntu, podemos confirmar su conectividad a través de la siguiente imagen:

Prueba IPv6 | FAQ | Mirrors | estadísticas

Probar tu conectividad IPv6.

A S

Sumario	Pruebas ejecutadas	Compartir Resultados / Contactar	Otros Sitios IPv6	Para el Servicio de Asistencia
Su dirección IPv4 en la Internet parece ser 185.193.172.15 (FIBERPLUS2AS)				
Su dirección IPv6 en la Internet parece ser 2001:470:1f08:3e9::2 (HURRICANE)				
Puesto que tienes IPv6, estamos incluyendo una ficha que muestra otros sitios IPv6 y cuán bien puede alcanzarlos. [más información]				
Tu servidor DNS (posiblemente controlado por tu ISP) parece tener acceso a Internet IPv6.				
Tu puntuación de preparación				
10/10				
para su estabilidad y preparación de IPv6, cuando editores estén obligados a usar sólo IPv6				
Click para ver Datos de prueba <small>(Actualizando estadísticas de la preparación IPv6 del lado del servidor)</small> <small>This instance (amsterdam.test-ipv6.com) is hosted at Linode.</small>				

Copyright (C) 2010, 2024 Jason Fesler. Todos los derechos reservados. Versión 1.1.1016 (409dbc3)
[Mirrors](#) | [Fuente](#) | [Email](#) | [Atribuciones](#) | [Debut](#) |

Este es un espejo de test-ipv6.com. Las opiniones expresadas aquí pueden o no reflejar la opinión del dueño del espejo.

Figura 3-9 Conectividad IPv6 Hurricane Ubuntu [15]

3.2.2 VPN con soporte IPv6

Una VPN es una red virtual privada. Su objetivo es mejorar la privacidad, la seguridad y mantener el anonimato mientras navegas, permitiendo acceder a servicios de otros lugares que no son accesibles desde tu región [18].

Este software permite ocultar tu dirección IP real, proporcionando una dirección IP ubicada en otra zona geográfica. Además, las comunicaciones entre el equipo y el servidor están encriptadas, garantizando al usuario una mayor privacidad.

Muchas de estas VPNs ofrecen un plan de direccionamiento IPv6, haciendo que redes que no dispongan de IPv6 de manera nativa, puedan gozar de este protocolo.

3.2.2.1 Funcionamiento

El funcionamiento de una VPN con soporte para direccionamiento IPv6 es el siguiente [18]:

- Una vez completada la configuración correspondiente, el cliente obtiene una dirección IPv6 global para ser usada a través de Internet. Al no disponer de IPv6 en la red local, el paquete IPv6 no puede ser enviado directamente, por lo que es necesario encapsular el paquete dentro de un paquete IPv4.
- El cliente al querer acceder a algún destino IPv6 en Internet, por ejemplo Google.com, la resolución DNS se realiza a través del túnel establecido, obteniendo la dirección IPv6 del dominio asociado.
- El equipo origen genera un paquete IPv6 con la IP que le ha entregado el servidor DNS, y lo encapsula en un paquete UDP dentro de IPv4.
- El paquete UDP atraviesa el túnel hasta el servidor, el cual lo descifra y desencapsula recuperando el paquete IPv6 original, reenviándolo por la red IPv6 hasta el destino.
- Una vez llega al destino, este genera un paquete IPv6 de respuesta enviándolo de vuelta al cliente.
- El servidor, lo vuelve a cifrar y encapsular en un paquete UDP con IPv4 y lo envía de vuelta por el túnel hasta ser recibido por el cliente.

- El cliente desencapsula el paquete, obteniendo el paquete IPv6 original para poder procesarlo posteriormente.

Las VPNs con soporte para direccionamiento IPv6 son una opción a tener en cuenta a la hora de querer obtener conectividad IPv6. El inconveniente sigue siendo el tema de la velocidad que, en aspectos como la descarga de archivos grandes, puede notarse una bajada de rendimiento. Aun así, es una gran alternativa para tener en cuenta a la hora de querer obtener IPv6.

A través de la herramienta Wireshark, podemos observar los paquetes que se tramanan por la red para cualquier servicio implementado:

```
> Frame 139: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{E3CF15FC-04D7-428B-B60D-F241E084ED86}, id 0
> Ethernet II, Src: Xerox_00:00:00 (00:00:02:00:00:00), Dst: c6:8e:20:00:01:01 (c6:8e:20:00:01:01)
> Internet Protocol Version 6, Src: fd00:6968:6564:67b::a39:ef67 (fd00:6968:6564:67b::a39:ef67), Dst: releases.ubuntu.com (2620:2d:4002:1::109)
    0110 .... = Version: 6
    .... 0000 0000 .... .... .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    .... 0001 1111 0001 0110 0011 = Flow Label: 0x1f163
    Payload Length: 20
    Next Header: TCP (6)
    Hop Limit: 128
> Source Address: fd00:6968:6564:67b::a39:ef67 (fd00:6968:6564:67b::a39:ef67)
> Destination Address: releases.ubuntu.com (2620:2d:4002:1::109)
    [Stream index: 3]
> Transmission Control Protocol, Src Port: 58566, Dst Port: 443, Seq: 522, Ack: 79460, Len: 0
```

Figura 3-10 Paquete UDP VPN Hide.me

3.2.2.2 Configuración Servicio VPN Hide.me

Para nuestro proyecto, utilizaremos el software Hide.me [19], que nos ofrece un plan gratuito de un mes proporcionándonos un plan de direccionamiento IPv6, ideal para el estudio que estamos realizando.

El proceso de configuración es el siguiente:

- A través de página oficial de Hide.me, podremos elegir el plan gratuito seleccionando el ícono del sistema operativo donde queremos instalarlo.
- Al seleccionar el ícono te descarga un ejecutable, el cual al iniciarse aparece una interfaz gráfica. A través de esta gráfica puedes elegir utilizar el plan gratuito.
- Una vez dentro, simplemente basta con iniciar la VPN, seleccionar la región geográfica que nos interese y darle a comenzar.

Una vez establecido la conexión, el sistema operativo crea una nueva interfaz de red, donde se asignan una dirección IPv4 pública y direccionamiento IPv6, el cual será de gran utilidad para realizar el estudio correspondiente en nuestro proyecto.

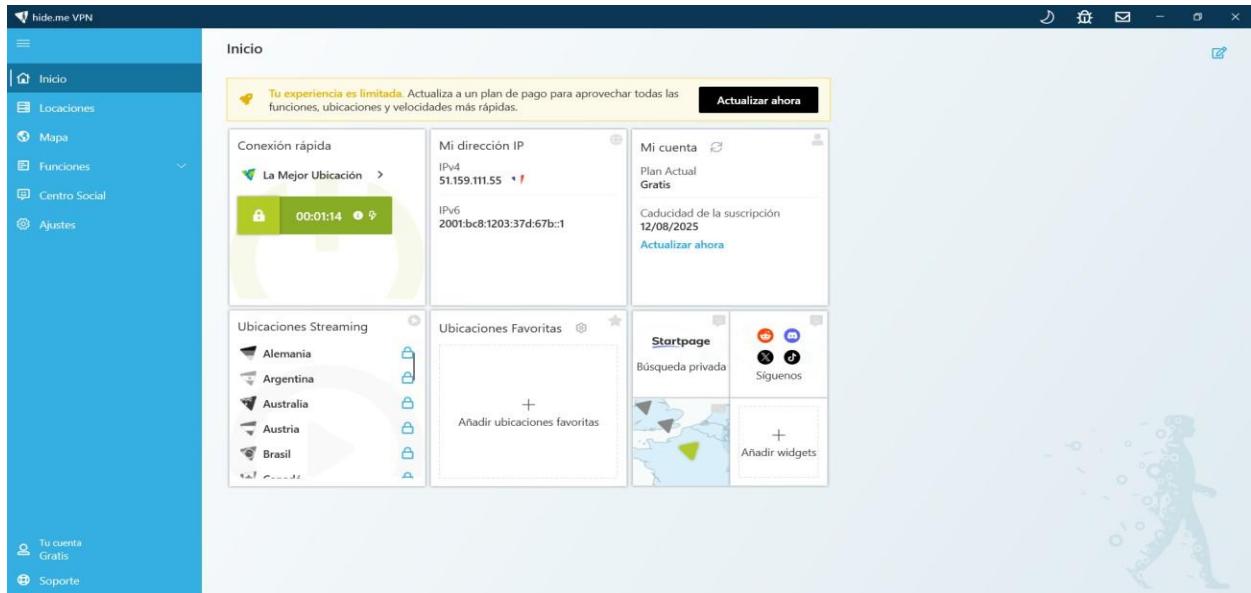


Figura 3-11 Configuración VPN Hide.me

A través del comando **ipconfig**, podemos observar la configuración de la interfaz de red:

Figura 3-12 Interfaz VPN Hide.me

3.1.2 VPN con Wireguard con servidor una VPS

Wireguard es un protocolo VPN que se caracteriza por ser extremadamente simple, ya que tiene un código base bastante pequeño en comparación a otras soluciones como OpenVPN. Esta reducción de código hace que sea una solución bastante rápida y para nada complejo [20].

También, ofrece una criptografía moderna, implementando últimas tecnologías criptográficas para asegurar las conexiones. Además, se integra bastante bien con muchos sistemas operativos, haciendo que sea bastante versátil.

Para este proyecto, hemos realizado el estudio de Wireguard implementando como servidor y alojador del direccionamiento IPv6 una VPS que hemos contratado previamente.

Un VPS es un entorno virtualizado dentro de un servidor físico. Dentro del servidor físico, se extienden varios servidores virtuales independientes, con recursos dedicados, a pesar de compartir el mismo hardware con otros VPS. Cada VPS funciona como una máquina virtual aislada con sus propias características. Además, posee bastante flexibilidad, ya que permite al usuario realizar los cambios que desee sin provocar interrupciones.

3.2.2.3 Funcionamiento

Una vez realizada la configuración en ambos extremos, el funcionamiento es el siguiente [20]:

- En primer lugar, una vez configurado ambos extremos, se establece un túnel a partir de claves públicas y privadas y se establece una conexión segura a través del protocolo de transporte UDP.
- El equipo cliente quiere enviar tráfico IPv6, enviando por ejemplo una consulta al dominio Google.com
- El equipo realiza una consulta al servidor DNS que haya sido configurado para obtener la dirección IPv6 del dominio.
- Una vez obtenida, el cliente crea un paquete IPv6 que Wireguard cifra y encapsula dentro de un paquete UDP con dirección IPv4 destino la IP pública de la VPS.
- Este paquete es enviado por la red, atravesando los nodos intermedios hasta ser recibido por el servidor VPN que en este caso es la VPS.
- El paquete se desencapsula, recuperando el paquete IPv6 original.
- El servidor VPS encamina el paquete IPv6 original por la red global IPv6 hasta su destino.
- Cuando el destino recibe el paquete IPv6, genera un paquete IPv6 de respuesta de vuelta al servidor.
- El servidor recibe el paquete, cifrándolo nuevamente y lo encapsula dentro de un paquete UDP con IPv4, el cual envía de vuelta al cliente.
- El cliente recibe el paquete, lo desencapsula y recupera el paquete IPv6 original para poder procesarlo posteriormente.

Este servicio es una gran solución para el problema del direccionamiento IPv6. Como hemos comentado, es una VPN bastante rápida por su pequeño código base, con apenas 3000 líneas de código. Además, posee una buena seguridad y se integra muy bien con muchos sistemas operativos. El inconveniente es que contratar una VPS no es gratis, por lo que debe pagarse una cuota mensual para poder tener siempre el servidor activo. Por lo demás, es una solución bastante fiable, rápida y estable.

Utilizando la herramienta Wireshark, podemos observar los paquetes enviados por la red para analizar su contenido y poder analizar con más detalle los servicios implementados:

```
> Frame 141032: 1494 bytes on wire (11952 bits), 1494 bytes captured (11952 bits) on interface \Device\NPF_{49A0D2ED-FD5D-4A8B-81FF-7AF14D919879}, id 0
> Ethernet II, Src: HuaweiTechno_ca:01:be (98:17:3f:ca:01:be), Dst: Intel_74:5f:6d (4c:1d:96:74:5f:6d)
< Internet Protocol Version 4, Src: static.41.217.201.195.clients.your-server.de (195.201.217.41), Dst: 192.168.18.5 (192.168.18.5)
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1480
    Identification: 0x4e92 (20114)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 52
    Protocol: UDP (17)
    Header Checksum: 0xc2f2 [validation disabled]
      [Header checksum status: Unverified]
    Source Address: static.41.217.201.195.clients.your-server.de (195.201.217.41)
    Destination Address: 192.168.18.5 (192.168.18.5)
      [Stream index: 0]
  > User Datagram Protocol, Src Port: 51820, Dst Port: 58512
  > WireGuard Protocol
```

Figura 3-13 Paquete IPv4 VPN Wireguard

```

> Frame 87015: 1420 bytes on wire (11360 bits), 1420 bytes captured (11360 bits) on interface \Device\NPF_{D19E8E38-CEAB-2957-8C3D-A26A8783FC20}, id 0
  Raw packet data
< Internet Protocol Version 6, Src: releases.ubuntu.com (2620:2d:4002:1::108), Dst: DESKTOP-29HMC6K.local (2a01:4f8:1c0c:7b00::2)
  0110 .... = Version: 6
  .... 0000 0000 .... .... .... .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
  .... 0001 0110 1101 1000 1100 = Flow Label: 0x16d8c
  Payload Length: 1380
  Next Header: TCP (6)
  Hop Limit: 51
  > Source Address: releases.ubuntu.com (2620:2d:4002:1::108)
  > Destination Address: DESKTOP-29HMC6K.local (2a01:4f8:1c0c:7b00::2)
  [Stream index: 5]
> Transmission Control Protocol, Src Port: 443, Dst Port: 64298, Seq: 59568548, Ack: 522, Len: 1360

```

Figura 3-14 Paquete IPv6 VPN Wireguard

3.2.2.4 Configuración Servicio Wireguard+VPS

Para realizar el proyecto, hemos considerado elegir una VPS a través del proveedor Hetzner [21], que destaca por su reconocida fiabilidad y su costo, el cual es bastante reducido. Además, la VPS viene con una dirección IPv4 pública y un rango de direcciones IPv6 globales, perfecto para nuestro estudio.

En cuanto a la conectividad que nos ofrece, este servicio trabaja tanto con IPv4 como con IPv6, algo que es fundamental para nuestro proyecto. La dirección IPv4 que trae asignada es la **195.201.217.41** y su prefijo IPv6 global la **2a01:4f8:1c0c:7b00::/64**.

Para la configuración del servidor, hemos utilizado la VPS de Hetzner la cual contiene el sistema operativo Ubuntu. Su configuración se puede observar en la imagen siguiente:

```

GNU nano 7.2
/etc/wireguard/wg0.conf
[Interface]
Address = 2a01:4f8:1c0c:7b00::1/64
ListenPort = 51820
PrivateKey = yBKmgrSPFBkpF/w810tWdSh6uzak0fGcL/t1f/3krGI=
```

PostUp = sysctl -w net.ipv6.conf.all.forwarding=1

```

PostUp = ip6tables -A FORWARD -i wg0 -j ACCEPT
PostUp = ip6tables -A FORWARD -o wg0 -j ACCEPT
```

PostDown = ip6tables -D FORWARD -i wg0 -j ACCEPT

```

PostDown = ip6tables -D FORWARD -o wg0 -j ACCEPT
```

[Peer]

```

PublicKey = JpDGnkvselP0phijpkpRF0Q7tlzXMsLkM1loyDK/y1wM=
AllowedIPs = 2a01:4f8:1c0c:7b00::2/128
```

[Peer]

```

PublicKey = iNH2cZbiLifAbDwgwUG0JgApIqlqqcw+Rx9YZXEyS0o=
AllowedIPs = 2a01:4f8:1c0c:7b00::3/128
```

[Peer]

```

PublicKey = HTL0sIwifvx/wz/s+sXsLJZXejAuHeutJZ5whAEnC1Y=
AllowedIPs = 2a01:4f8:1c0c:7b00::4/128
```

Figura 3-15 Configuración Servidor Wireguard

Como podemos observar en la Figura 3-15, en la sección Interface vamos a definir la configuración principal del servidor. En el campo Address especificamos la dirección IPv6 que va a tener el servidor en el túnel. Además, este servidor escucha conexiones en el puerto que le especifiquemos en la opción ListenPort.

PostUp y PostDown son opciones que utilizamos en la configuración de Wireguard que se ejecutan al activar o desactivar la interfaz de red wg0.

Al encender la interfaz establecemos con **sysctl -w net.ipv6.conf.all.forwarding=1** el reenvío de paquetes IPv6. Además, empleamos varias reglas de firewall que permiten el reenvío del tráfico a través del túnel.

Al apagar la interfaz, utilizamos **ip6tables -D FORWARD** para eliminar todas las reglas creadas previamente cuando estaba encendida.

En la sección Peer es donde se configura cada cliente que quiere tener acceso a este servidor. Como

podemos observar, tenemos tres clientes configurados en el servidor y para su configuración simplemente basta con indicar su clave pública y la opción AllowedIPs donde se especifica la dirección o el prefijo del cliente.

Para la configuración del cliente hemos descargado el software en el equipo con Windows y su configuración es la siguiente que se muestra por pantalla:

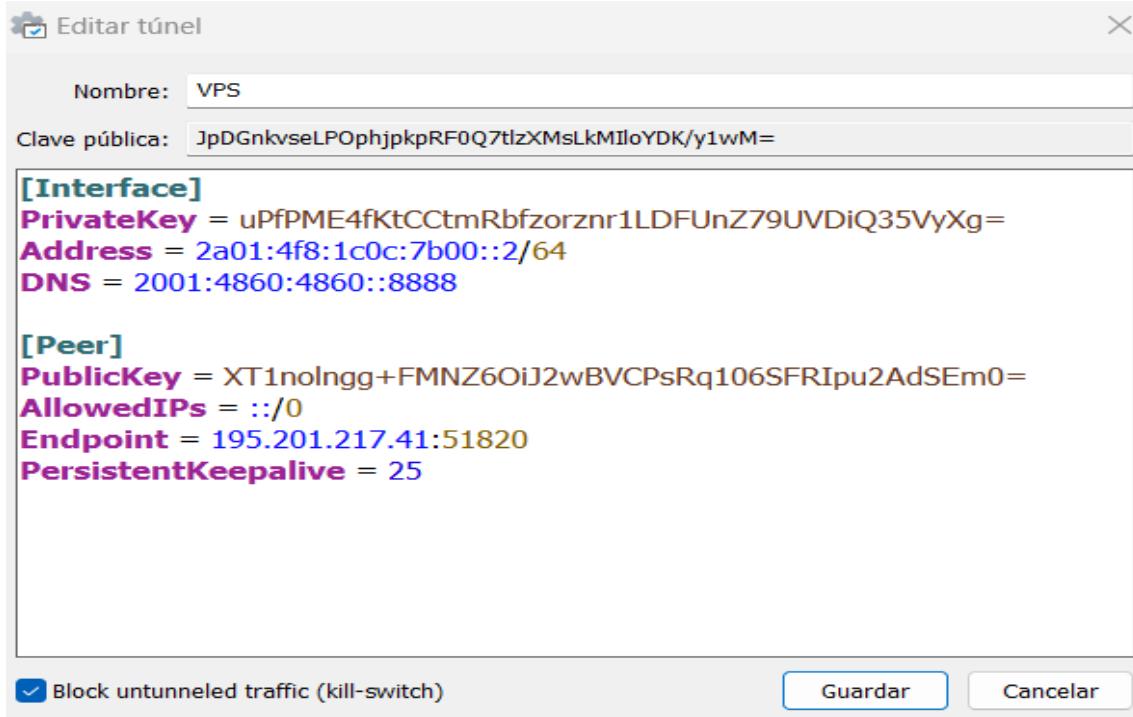


Figura 3-16 Configuración Cliente Wireguard

En la Figura 3-16, en la sección Interface podemos encontrarnos con el campo Address donde asignamos una dirección IPv6 al cliente. Además, a través del campo DNS podemos asignarle el servidor DNS que quiera usar el cliente para resolver los dominios correspondientes.

En cuanto a la sección Peer, vamos a definir la configuración del servidor al que se conecta. En ella asignaremos el campo PublicKey para emparejarse con la clave privada del servidor.

En el campo AllowedIPs hemos especificado que todo el tráfico IPv6 del cliente se envíe a través del túnel a través de la configuración ::/0.

En el campo Endpoint se define la dirección IP y el puerto donde está escuchando el servidor. Este campo es muy importante para el cliente para poder formar el túnel.

Por último, el atributo PersistentKeepalive se utiliza para que en este caso cada 25 segundos, el cliente envíe un paquete keep-alive manteniendo así la conexión activa.

Podemos confirmar su conectividad a través de la siguiente imagen:

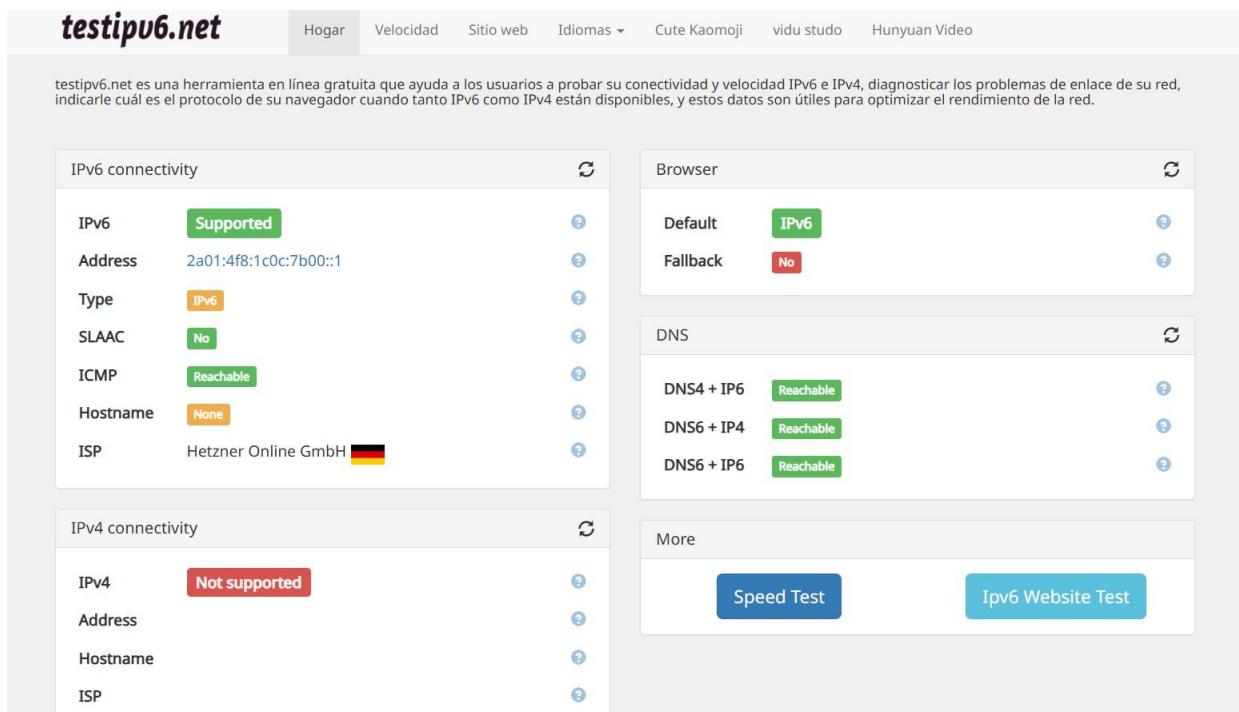


Figura 3-17 Conectividad IPv6 VPN Wireguard [22]

Observamos en la Figura 3-17 como existe conectividad IPv6 y la ubicación del ISP nos indica la VPS que hemos contratado para este proyecto.

3.3 Comparación de Servicios

A continuación, presentamos una tabla con las similitudes y diferencias de cada servicio implementado:

Aspecto	Similitudes	Diferencias
Objetivo	Proporcionar conectividad IPv6 a redes sin soporte nativo	Se aplican distintas tecnologías para llevarlo a cabo.
Encapsulación	Encapsular tráfico IPv6 dentro de paquetes IPv4	Empleo del protocolo UDP para Wireguard+VPS y Hide.me. Uso del protocolo 41 para TunnelBroker.
Configuración	Configuración de cliente	TunnelBroker utiliza comandos, Hide.me necesita ejecutable, Wireguard+VPS configuración cliente-servidor.
Necesidad IPv4	Necesidad de IPv4 para funcionar correctamente	TunnelBroker y Wireguard+VPS necesitan IPv4 libre de CG-NAT para funcionar. Hide.me no lo necesita.
Rendimiento	Latencia extra por encapsulación	TunnelBroker presenta mayor inestabilidad
Conectividad IPv6	Permiten acceder a la red IPv6 global	Asignación de prefijo /64 en los servicios TunnelBroker y Wireguard+VPS. Dirección única en Hide.me
Infraestructura	Extremo externo involucrado	TunnelBroker, Hide.me extremo depende de un proveedor externo. Wireguard+VPS posee control total de

		ambos extremos.
--	--	-----------------

Tabla 3-1 Similitudes y Diferencias Servicios

4 PARÁMETROS DE EVALUACIÓN Y HERRAMIENTAS EMPLEADAS

Es importante conocer la función que realiza cada parámetro de red, así como su impacto en ella en base al valor obtenido en las distintas mediciones. Para ello, vamos a realizar un estudio previo de todos los parámetros de red para conocer la función que realiza cada uno y su impacto en la red. Además, comentaremos varias herramientas de gran utilidad y precisión para llevar a cabo las mediciones.

4.1 Parámetros de red

Los parámetros de red que explicaremos a continuación son esenciales para garantizar un buen funcionamiento de una red. Por tanto, mantener estos parámetros en unos valores óptimos nos permite afirmar que la red es estable, rápida y ofrece un rendimiento adecuado.

4.1.1 Latencia

La latencia mide cuánto tiempo tarda la información o los paquetes de red en llegar desde el origen hasta el destino. Normalmente es un parámetro de red que medimos en el orden de milisegundos [23].

Medir este parámetro es bastante importante para evaluar el rendimiento de una red. Nos permite evaluar la calidad de una red, determinar la velocidad de respuestas en juegos online, videollamadas entre otras aplicaciones.

Como podemos esperar, nos interesa que los servicios implementados tengan una baja latencia, ya que es indicio de un menor retraso en la entrega de paquetes y por consecuente un mayor rendimiento en la red. Una latencia alta, en cambio, es sinónimo de demoras, pérdidas de conectividad e interrupciones.

Los factores que afectan principalmente a la latencia son los siguientes:

- **Número de saltos:** Son los saltos que tiene que realizar el paquete para llegar al destino. Al haber más nodos que el paquete tenga que cruzar para alcanzar el destino, mayor latencia añadimos.
- **Distancia Física:** La distancia física entre los nodos que cruza el paquete puede conllevar a un aumento de la latencia.
- **Medio empleado:** Si la tecnología empleada no es una tecnología moderna o bien la red puede congestionarse con gran facilidad, añadiremos una latencia extra al paquete transmitido.
- **Rendimiento de los servidores:** Un servidor lento puede añadir latencia extra a la hora de procesar el paquete, haciendo que el rendimiento se vea afectado negativamente.

En la práctica, medir la latencia puede resultar muy complejo, por lo que los profesionales de red utilizan el parámetro RTT (Round-Trip Time). El RTT mide el tiempo que tarda el paquete en ir desde el equipo origen hasta el destino y volver, es decir, es el tiempo de ida y vuelta de un paquete. Puede medirse fácilmente con herramientas como Ping, traceroute entre otras [24].

En cuanto a los valores de referencia de RTT que nos indica un mayor o menor rendimiento tenemos lo siguiente [25]:

- **0-30 ms:** Estos valores nos indican una latencia muy buena, ideales para aplicaciones en tiempo real.
- **30-60 ms:** Sigue siendo una latencia buena y aceptable para la mayoría de los servicios.
- **60-100 ms:** Estos valores indican un posible retraso, afectando a algunos servicios.

- **+100 ms:** Indica latencia alta influyendo de manera negativa en la experiencia del usuario.

Para nuestro estudio, medir este parámetro será fundamental para evaluar la calidad de los servicios implementados.

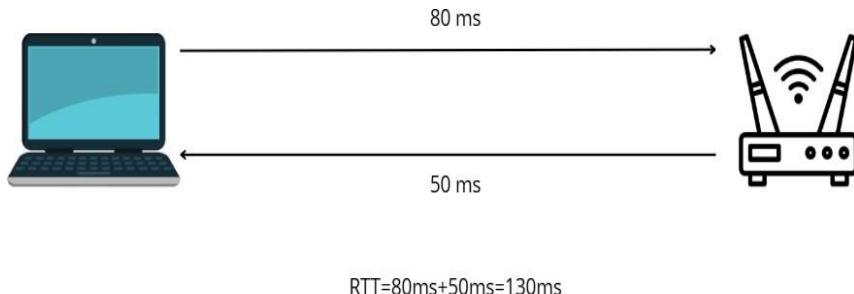


Figura 4-1 RTT

4.1.2 Pérdida de paquetes

La pérdida de paquetes es un parámetro el cual mide los paquetes que no son capaces de llegar al destino, ya sea porque son descartados por nodos intermedios o se pierden durante su transmisión. Este parámetro de red se mide en porcentaje [26].

La medición típica consiste en enviar entre 100-1000 paquetes a un destino y calcular con las herramientas adecuadas el número de paquetes que no han sido capaces de llegar al destino. Posteriormente, para obtener el porcentaje total de paquetes perdidos, realizamos la división de los paquetes perdidos entre el total de paquetes enviados y multiplicamos por 100.

Existen varios fenómenos que normalmente provocan una mayor pérdida de paquetes. Las principales causas son las siguientes:

- **Congestión de red:** Cuando los buffers de los equipos de red como routers, switches y demás se llenan, los paquetes pueden no llegar a ser procesados por estos equipos y por tanto pasan a ser descartados.
- **Redes Wi-Fi:** Utilizar redes Wi-Fi puede aumentar la pérdida de paquetes, debido a factores como interferencias de señales externas o el uso de bandas saturadas.
- **Errores de configuración:** Posibles errores de configuración o parámetros mal configurados como pueden ser el MTU (Maximum Transmission Unit) o TTL (Time To Live), pueden provocar un aumento en el número de paquetes perdidos.
- **Estado de Hardware:** El estado del hardware de la red puede influir en la pérdida de paquetes, ya que equipos defectuosos, cables deteriorados entre muchas otras pueden llevar a una mayor pérdida de paquetes.

Además, diferentes servicios de tunelaje que estudiaremos en este proyecto suelen tener un porcentaje de pérdidas mayor debido a factores como tener más tramos por los que debe pasar el paquete, lo que incrementa la posibilidad de pérdida del paquete.

Herramientas como **ping** nos indica conocer fácilmente el porcentaje de paquetes que no han sido capaces de encontrar el destino. Los valores de referencia para evaluar la calidad de una red en base al parámetro pérdida

de paquetes son los siguientes [27]:

- **0%:** Es el porcentaje de pérdida ideal, los datos llegan correctamente.
- **<1%:** Porcentaje de pérdida estable. En general no apreciada, afectando nada o muy poco a la red.
- **1%-5%:** Aparición de problemas notorios, haciendo que muchas aplicaciones dejen de funcionar correctamente.
- **>5%:** Problema serio, provocando interrupciones y degradación severa de la red.

Lo recomendable y óptimo para una red es tener una pérdida sostenida de paquetes inferior al 1%, asegurando así una buena estabilidad y rendimiento a la red.

Este parámetro en nuestro proyecto tendrá una gran importancia para evaluar los servicios implementados y su estabilidad para ofrecer una buena experiencia al usuario que lo implementa.

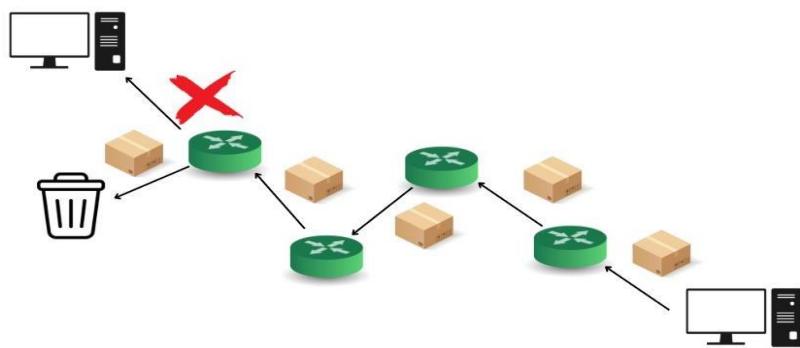


Figura 4-2 Pérdida de paquetes

4.1.3 Ancho de banda

Es un parámetro que mide el volumen de datos que pueden atravesar la red simultáneamente. Su medición se suele realizar en volumen de datos por unidad de tiempo, más concretamente en megabits por segundo [28].

Este parámetro no es una medición directa de la velocidad, sino de capacidad, ya que, aunque es cierto que a mayor ancho de banda mayor velocidad de Internet, esta va a depender de factores como latencia y el rendimiento general de la red.

Para una red doméstica, el ancho de banda disponible está proporcionado por el ISP, el cual debido a la tarifa contratada asignará un límite u otro. En entornos empresariales, influye directamente en la eficiencia y la productividad.

Para este parámetro existen varios factores que pueden influir negativamente en el ancho de banda. Los principales factores son los siguientes:

- Una larga distancia entre el router y el ISP puede provocar un ancho de banda bajo. Además, múltiples dispositivos conectados al mismo tiempo y aplicaciones en segundo plano que utilizan el ancho de banda simultáneamente también pueden influir negativamente y provocar un ancho de banda bajo.
- En redes Wi-Fi el ancho de banda puede verse limitado por factores como las interferencias contra otros dispositivos, además de obstáculos físicos como paredes, muebles y demás.
- Un ancho de banda bajo puede provocar descargas lentas, retrasos en juegos de línea y aplicaciones en tiempo real, degradando la experiencia del usuario.

En cuanto a ventajas de disponer de un ancho de banda elevado y estable, podemos contar con una buena experiencia para el usuario, con navegaciones rápidas, descargas a una buena velocidad y por tanto en un tiempo mucho menor. Además, disponer de una buen rendimiento y experiencia en aplicaciones en tiempo real, como videollamadas, juegos en línea, entre otros.

Este parámetro se puede medir con muchas herramientas. Una de las más populares es la web speedtest.net, que evalúa el ancho de banda de nuestra red y nos proporciona la medida del ancho de banda de descarga y subida.

El ancho de banda recomendado por usuario para cada servicio lo podemos representar en la siguiente tabla:

Servicio	Ancho de banda recomendado
Navegación Web	3-5 Mbps
Correo electrónico	3-5 Mbps
Descarga de archivos	10-50 Mbps
Juegos en línea	50-100 Mbps
Descarga de archivos de gran tamaño	100-150 Mbps
Transmisión de video en diferentes resoluciones	5-15 Mbps

Tabla 4-1 Ancho de banda recomendado [29]

El ancho de banda necesario es el que necesita cada equipo de la red para realizar cada actividad de las mencionadas. Cada una de ellas restará un porcentaje de ancho de banda total disponible.

Por tanto, mientras más usuarios existan en la red con requerimientos altos de red, conviene tener un buen ancho de banda para que todos los servicios previamente analizados estén bien cubiertos y no ofrezcan una mala experiencia al usuario.

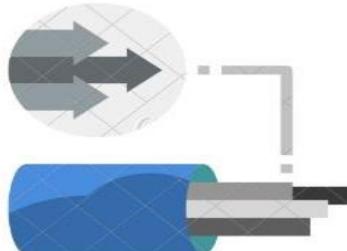


Figura 4-3 Ancho de banda

4.1.4 Jitter

El jitter mide la variación en el tiempo de llegada de los paquetes de datos en una red. Indica si los paquetes llegan de manera constante o presentan variaciones en su retardo [30].

Para poder entenderlo de una manera mucho más práctica, pondremos un ejemplo. Imaginemos que enviamos cada 20 milisegundos dos paquetes a un destino en concreto. El primer paquete llega en 20 ms y el segundo en 22 ms. Si los dos paquetes hubiesen llegado en 20 milisegundos, no existiría jitter. Pero al existir una variación en los tiempos, tendremos un jitter de 2 milisegundos.

Las consecuencias que pueden causar la aparición del jitter son varias. Las más comunes son las siguientes:

- **Hardware de baja calidad:** Un hardware o infraestructura de red de una baja calidad puede provocar que las señales se retrasen durante la transmisión.

- **Red congestionada:** Una red congestionada puede provocar que los paquetes se retrasen, aumentando el jitter.
- **Ruta variable:** Los paquetes no siempre siguen una ruta lineal de origen a destino, muchos pueden tomar diferentes rutas para el mismo destino, desembocando en tiempos de llegadas diferentes.

El jitter tiene un mayor impacto en servicios en tiempo real, como las videollamadas, juegos en línea o llamadas VoIP. Un jitter elevado puede provocar la aparición de interrupciones, retrasos y, en general, problemas que impiden el correcto funcionamiento de estos servicios. Por el contrario, el jitter apenas influye en servicios como la navegación web, descarga de archivos, correo electrónico, entre otros.

En general, los servicios en tiempo real deben de tener un bajo jitter para mantener una calidad constante. Mantener el jitter por debajo de los 30 milisegundos es lo ideal para el correcto funcionamiento de estos servicios.

Para medir este parámetro de red podemos utilizar varias herramientas. Una de las más populares es la web SpeedTest de Cloudflare, la cual hace una medición de varios parámetros de red, entre los que se encuentra el jitter.

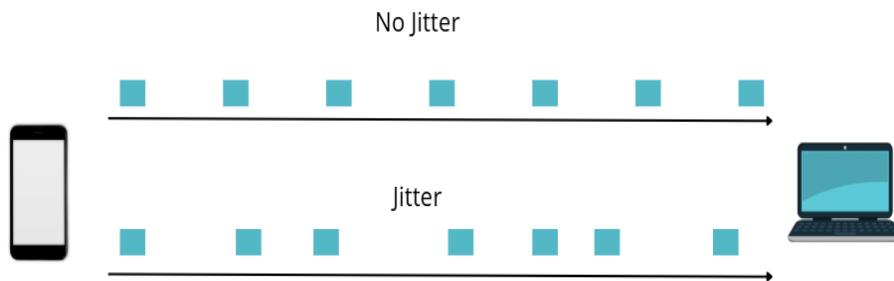


Figura 4-4 Jitter

4.1.5 MTU

La MTU determina el tamaño máximo de un paquete de datos que puede ser enviado a través de la red sin tener que recurrir al proceso de fragmentación. Su unidad de medida es el byte [31].

Si un paquete excede el MTU permitido, deberá ser fragmentado en varios paquetes más pequeños, provocando una mayor latencia y afectando negativamente al rendimiento general de la red.

Una buena configuración de MTU puede mejorar considerablemente la eficiencia de la red, reduciendo la fragmentación de paquetes. En cambio, una mala configuración de MTU puede generar una mayor cantidad de paquetes, trayendo consigo una posible sobrecarga y afectando negativamente al rendimiento general de la red.

El valor de MTU puede variar dependiendo del tipo de conexión de red. Por ejemplo, en redes Ethernet lo común es tener una MTU de 1500 bytes. Sin embargo, este valor puede variar dependiendo de la configuración de red.

Para medir este parámetro de red podemos utilizar varias herramientas:

- En Windows, una herramienta muy popular para comprobar la MTU de una interfaz es **netsh**. El comando **netsh interface ipv4/ipv6 show subinterfaces**, te permite ver entre otros parámetros la MTU que tiene cada interfaz, tanto para IPv4 como para el protocolo IPv6.
- En Linux, los comandos como **ip link show** o **ifconfig** nos permiten ver las interfaces de red y su configuración, donde se indica la MTU.

Además, estas herramientas te dan la opción de poder modificar la MTU que tienen las interfaces por defecto, ya sea de manera temporal o permanente.

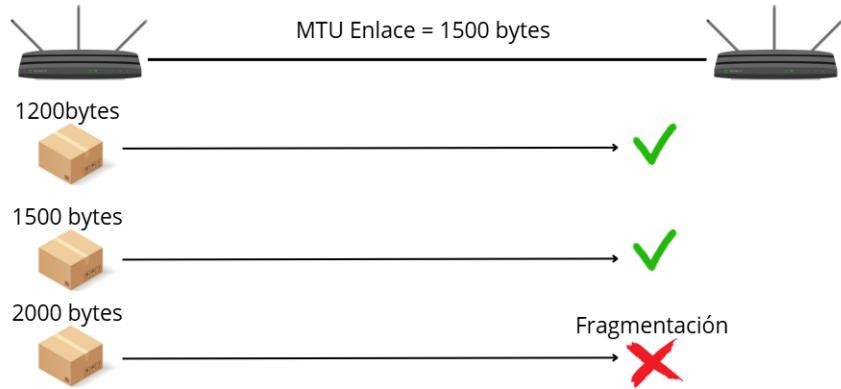


Figura 4-5 MTU

4.1.6 Tiempo de resolución DNS

El tiempo de resolución DNS es un parámetro con el que podremos calcular el tiempo que un equipo tarda en traducir un dominio a su dirección IP. Es un proceso necesario y obligatorio, ya que los equipos se comunican por IPs y no por dominios [32].

Los dominios son utilizados para ofrecer una mejor experiencia de navegación al usuario al no tener que depender de memorizar cada dirección IP de cada servicio al que quiera acceder.

El tiempo de resolución DNS incluye la consulta que un equipo realiza al servidor DNS correspondiente y la respuesta que este le devuelve. Además, existe la posibilidad de que el servidor DNS tenga que consultar a otros servidores para obtener una respuesta definitiva. Este proceso suele ser del orden de milisegundos.

Un tiempo de resolución bajo reduce el tiempo de carga de una página web, mejorando la experiencia del usuario. Para ofrecer una buena experiencia de usuario lo ideal es mantener la resolución DNS por debajo de los 100 milisegundos. Aunque lo óptimo es tener un tiempo de resolución DNS por debajo de los 50 milisegundos [30].

Para medir este parámetro de red utilizamos varias herramientas:

- Para el sistema operativo Windows, utilizamos la herramienta nslookup para poder realizar una consulta al servidor DNS para un dominio asociado. Combinando esta herramienta con la herramienta Wireshark, podremos capturar los paquetes que se traman para esa resolución DNS. A través de esa captura, podemos calcular la diferencia de tiempo entre la solicitud y la respuesta, obteniendo así el tiempo de resolución DNS.
- Para el sistema operativo Linux, utilizamos la herramienta dig que realiza la misma función que nslookup, pero además nos proporciona el tiempo que ha tardado en realizar la consulta, eliminando la necesidad de usar Wireshark para calcular los tiempos de resolución.



Figura 4-6 Tiempo de resolución DNS

4.1.7 Rutas

La ruta es el camino lógico y físico que siguen los paquetes desde su origen hasta el destino. Este camino no

siempre es directo, sino que en muchas ocasiones necesita pasar por nodos intermedios que se encargan de encaminar el paquete hasta el destino.

Los diferentes saltos entre sistemas intermedios son posibles gracias a la tabla de encaminamiento de cada router, la cual es utilizada para encaminar el paquete recibido por la interfaz adecuada, basándose en la dirección IP destino.

Para conocer la ruta completa que sigue un paquete tenemos varias herramientas:

- En Linux, la más popular es la herramienta traceroute, la cual realiza un análisis detallado de cada salto que tiene que realizar el paquete para llegar a su destino.
- En Windows se utiliza tracert, que realiza la misma función que el comando traceroute de Linux.

Estas herramientas nos proporcionan el RTT entre el origen y cada salto, indicando la IP de cada nodo intermedio.

Conocer la ruta que sigue un paquete nos sirve para realizar un diagnóstico de red, pudiendo detectar problemas que estén ocurriendo en algún punto intermedio de esta.

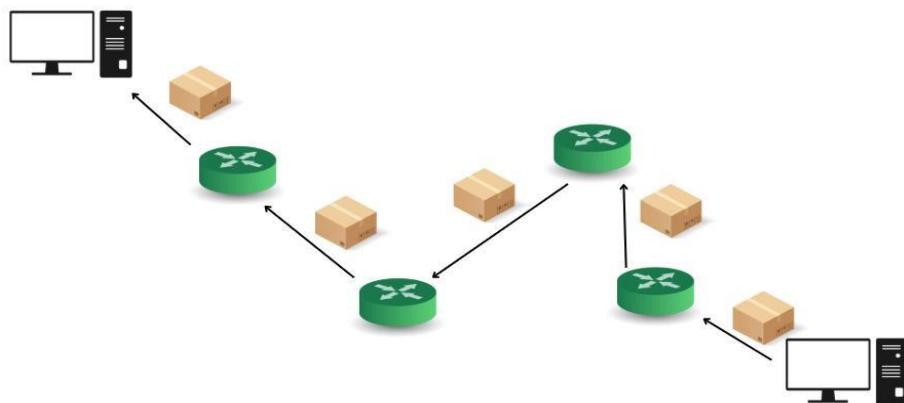


Figura 4-7 Rutas

4.2 Herramientas Utilizadas

En el desarrollo de este proyecto hemos utilizado los sistemas operativos Windows y Linux, concretamente Windows 11 y Ubuntu. Para ello, se han implementado distintas herramientas específicas para el cálculo de cada parámetro de red. A continuación, comentamos las principales herramientas empleadas y su utilidad en el proyecto.

4.2.1 NetScanTools

NetScanTools es un software diseñado para profesionales de redes. Es una herramienta diseñada para sistemas operativos Windows, ofreciendo una interfaz gráfica bastante intuitiva donde podemos llevar a cabo diversas opciones para realizar un análisis profesional de red [33].

En este proyecto, utilizamos la licencia gratis que nos ofrece este software, que permite un uso durante un tiempo limitado para llevar a cabo todas las mediciones sobre los distintos servicios implementados.

En nuestro caso, utilizamos este software para realizar pruebas a los distintos servicios implementados, midiendo tanto el tiempo de ida y vuelta como la pérdida de paquetes para un destino en concreto.

Para llevar a cabo estas mediciones, configuraremos el envío de un número alto de paquetes hacia un destino en

concreto, por ejemplo 1000. Cuando finalice el envío de paquetes, el software nos proporciona los valores máximos, mínimos y media del tiempo de ida y vuelta. Además, nos proporciona el total de paquetes que han sido capaces de llegar al destino, dejando así la oportunidad de poder calcular el porcentaje de pérdida de paquetes con la siguiente fórmula:

$$\text{Pérdida (\%)} = (\text{Número de paquetes perdidos} / \text{Número total de paquetes enviados al destino}) * 100$$

Su capacidad para poder integrar múltiples funciones en una sola opción lo hacen un software bastante interesante y llamativo para los profesionales de redes.

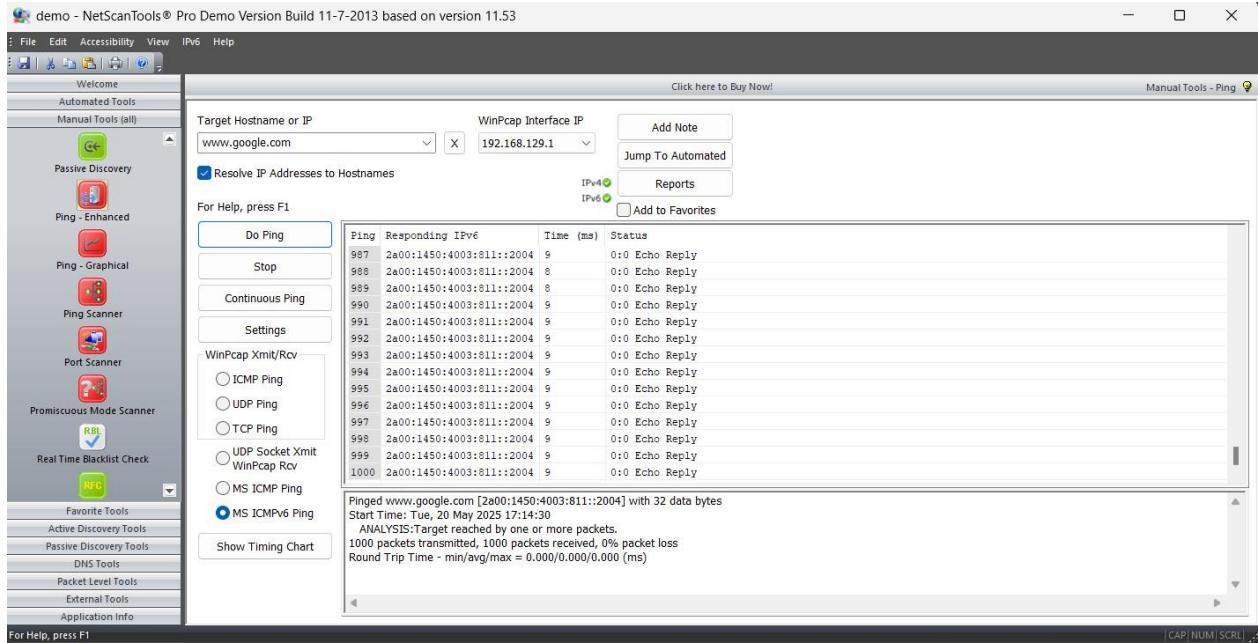


Figura 4-8 Herramienta NetScanTools [33]

4.2.2 PingPlotter

PingPlotter es una herramienta disponible principalmente en sistemas operativos Windows. Dispone de una interfaz gráfica para monitorizar en tiempo real la calidad de una conexión [34].

En este proyecto utilizamos la licencia gratis en la que durante un tiempo limitado podremos hacer uso para poder realizar las mediciones de parámetros de red claves como la latencia, el porcentaje de paquetes perdidos y la ruta que siguen para llegar al destino.

Una de las principales ventajas de esta herramienta es la combinación de varias funcionalidades, además de mostrarlas gráficamente en tiempo real. Es una herramienta que realiza un seguimiento de los paquetes enviados al destino, destacando la ruta que siguen estos, los tiempos de ida y vuelta y el porcentaje de paquetes perdidos en cada salto dado.

Su capacidad para combinar varias funciones y poder mostrarlas en gráficas en tiempo real, lo convierten en un software muy útil para profesionales de redes.

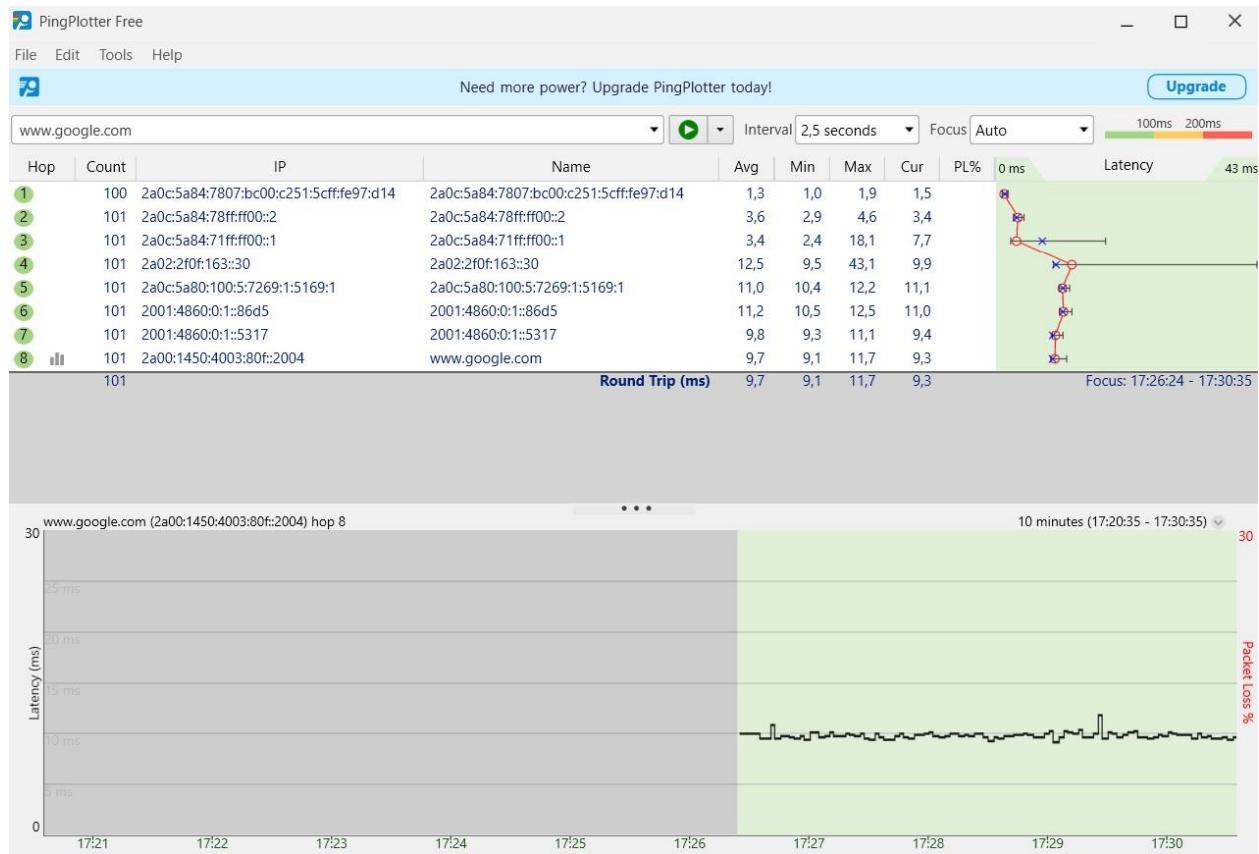


Figura 4-9 Herramienta PingPlotter [34]

4.2.3 Mtr (My traceroute)

Mtr es una herramienta utilizada en sistemas Linux que combina funciones de distintas herramientas como ping y traceroute. Puede ejecutarse por línea de comandos o a través de su interfaz gráfica. Proporciona valores de parámetros de red esenciales como son el tiempo de ida y vuelta, el porcentaje de paquetes perdidos y la ruta que siguen para llegar al destino [35].

Es una herramienta diseñada principalmente para equipos con sistema operativo Linux. La instalación de esta herramienta se realiza a través del terminal y depende del gestor de paquetes que se utilice. En entornos como Debian y Ubuntu se realiza utilizando el comando **sudo apt install mtr**.

Como la información se va actualizando periódicamente, podemos monitorizar constantemente la conexión siendo posible identificar puntos donde la conexión se ve degradada o genera retardos.

Es una herramienta que al combinar varias funcionalidades en una sola y su capacidad para ir actualizando la información periódicamente, la hacen una de las mejores herramientas de Linux para el análisis profesional de una red.

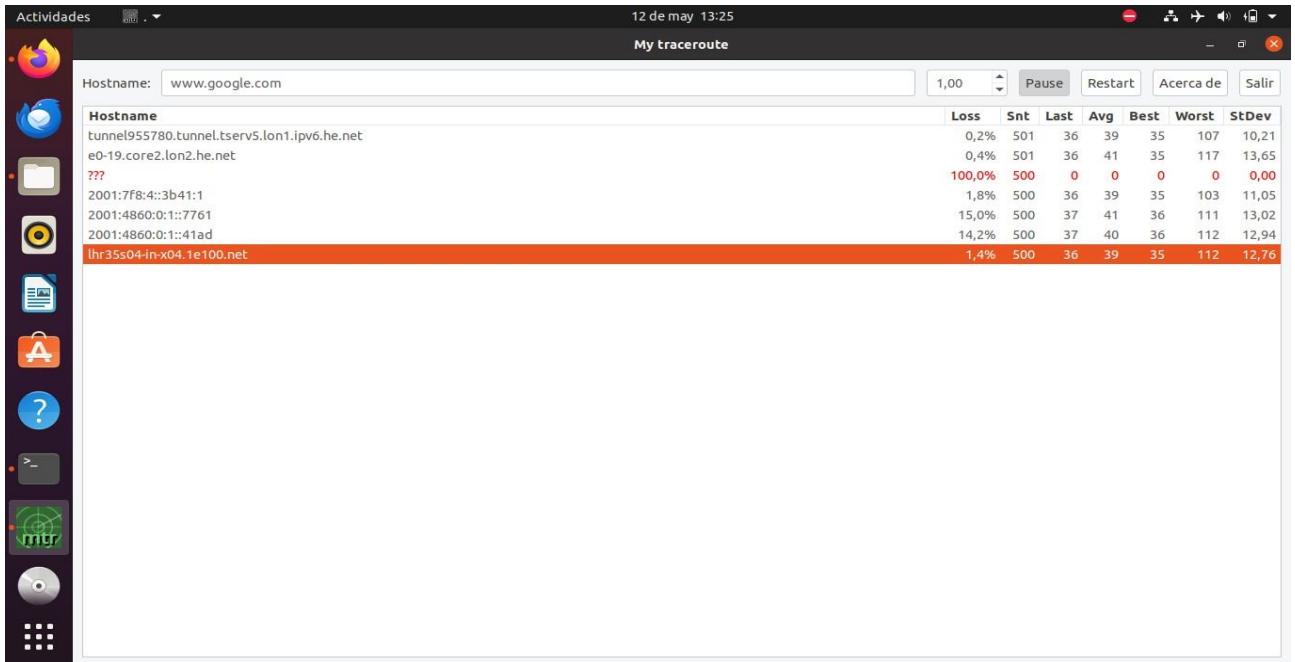


Figura 4-10 Herramienta Mtr [35]

4.2.4 Wireshark

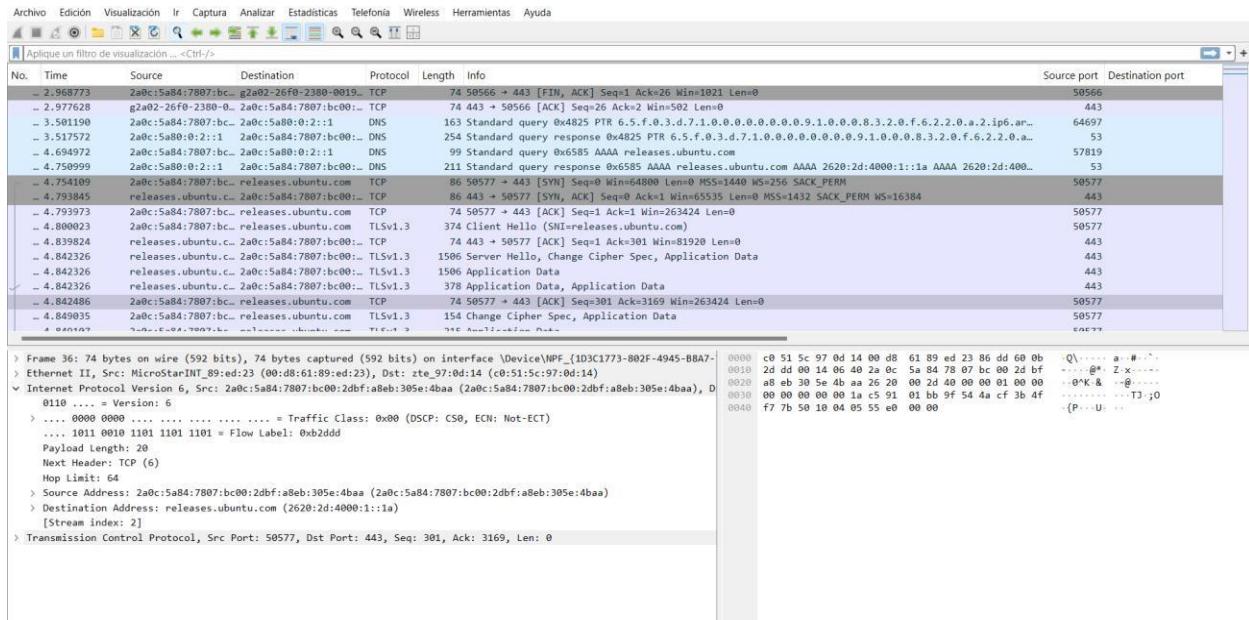
Wireshark es una de las herramientas más populares para el análisis de protocolos de red. Es de código abierto y gratuita. Además, posee una interfaz gráfica fácil de usar para que el usuario pueda capturar todo el tráfico de red en tiempo real y tenga la posibilidad de poder analizarlo de manera detallada [37].

Su poder para filtrar las búsquedas permite al usuario poder analizar un tipo de información específica en la captura correspondiente. Además, soporta cientos de protocolos, ya sea protocolos comunes como TCP, UDP, o protocolos más específicos como SIP.

En este proyecto utilizamos este software principalmente para medir los tiempos de resolución dns, filtrando previamente los paquetes tramitados y analizando y calculando el tiempo que tarda en resolver los dominios seleccionados.

Con cada servicio implementado analizamos el comportamiento detallado de los paquetes enviados para poder estudiar y entender con mayor precisión cada servicio, su comportamiento y como está establecido.

Su capacidad para mostrar el contenido detallado de los paquetes que se tramanan por la red lo hacen una herramienta ideal para analizar el comportamiento de una red de manera detallada y profesional.



4.2.5 Speedtest Cloudflare

Cloudflare Speed es una herramienta en línea que permite medir el rendimiento de la conexión de un usuario. Permite medir parámetros de red claves como:

- Ancho de banda.
- Latencia.
- Jitter.
- Pérdida de paquetes.
- Ubicación del servidor para realizar las pruebas correspondientes.
- IP y ubicación de la red analizada.

Con solo acceder a la web, la prueba empezará a medir todos los parámetros de red. Además, cuenta con una interfaz sencilla, permitiendo al usuario obtener los resultados sin una previa configuración.

Esta herramienta es bastante práctica para medir parámetros interesantes como jitter y ancho de banda que no son obtenibles con herramientas tradicionales.

Su facilidad de uso y precisión a la hora de obtener los resultados la convierten en una herramienta ideal y muy utilizada por profesionales de redes.

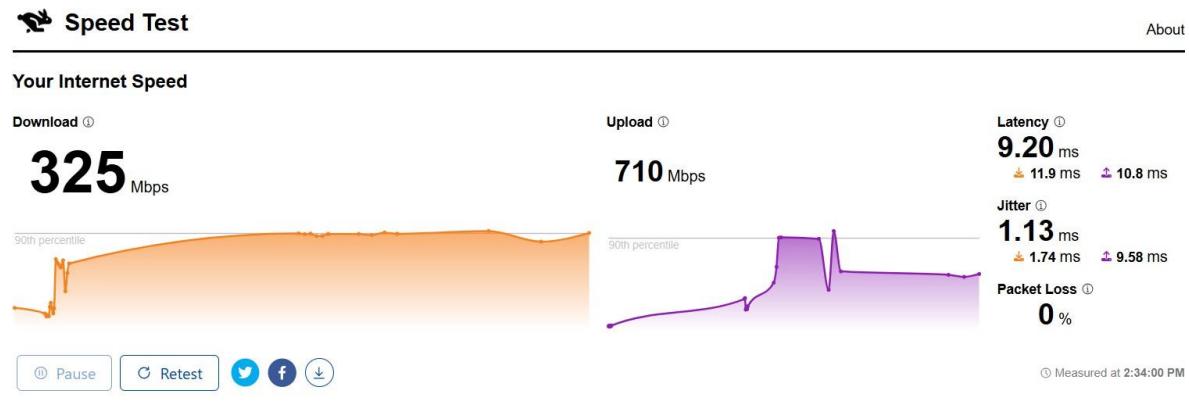


Figura 4-12 Web Speedtest [38]

4.2.6 Netsh/ifconfig

Para obtener la MTU de cada servicio implementado en este proyecto utilizamos diferentes herramientas según el sistema operativo seleccionado:

- En Windows, utilizamos la herramienta **netsh**. Nos permite mostrar y modificar la configuración de red de manera avanzada. Para mostrar la MTU de cada interfaz utilizamos el comando **netsh interface ipv4/ipv6 show subinterfaces**. Este comando muestra una lista de subinterfaces ya sea de IPv4 o IPv6, en la que se especifican parámetros claves entre los que se encuentra la MTU.

C:\Windows\System32>netsh interface ipv6 show subinterfaces					
MTU	MediaSenseState	Bytes de entrada	Bytes de salida	Interfaz	
4294967295	1	0	63144	Loopback Pseudo-Interface 1	
1420	1	17272	2017	VPS	
1500	1	259527	127927	Wi-Fi	
1500	2	10852595614	712812954	Ethernet	
1500	5	0	1943	Conexión de área local* 1	

Figura 4-13 Comando Netsh Interface IPv6 Show Subinterfaces

- En Linux, la herramienta **ifconfig** nos permite ver y modificar las configuraciones de cada interfaz de red. Entre todos los parámetros que nos muestra la herramienta, nos encontramos la MTU de cada interfaz. Tan solo nos basta con pasar por la línea de comandos la entrada **ifconfig** para que nos muestre la configuración de todas las interfaces de red.

4.2.7

└# ifconfig	
docker0:	flags=4099<UP,BROADCAST,MULTICAST> mtu 1500 inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255 ether 02:42:d9:96:4f:ca txqueuelen 0 (Ethernet) RX packets 0 bytes 0 (0.0 B) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 0 bytes 0 (0.0 B) TX errors 0 dropped 6 overruns 0 carrier 0 collisions 0
eth0:	flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet 192.168.18.12 netmask 255.255.255.0 broadcast 192.168.18.255 inet6 fe80::20c:29ff:fe70:f9c8 prefixlen 64 scopeid 0x20<link> ether 00:0c:29:70:f9:c8 txqueuelen 1000 (Ethernet) RX packets 666 bytes 54656 (53.3 KiB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 228 bytes 27870 (27.2 KiB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

Figura 4-14 Comando ifconfig

En este proyecto utilizamos estas herramientas para visualizar la MTU correspondiente de todos los servicios que implementamos, teniendo la posibilidad de modificarla para ver el impacto que tiene en la red.

4.2.8 Dig/nslookup

Para realizar consultas DNS y obtener los tiempos de resolución, empleamos una herramienta en función del sistema operativo seleccionado:

- En Windows, utilizamos la herramienta **nslookup** para realizar consultas DNS, obteniendo la dirección IP asociada a ese dominio. Esta herramienta no muestra directamente el tiempo de resolución, por lo que debemos combinar su uso con la herramienta Wireshark. A través de esta herramienta capturamos los paquetes enviados para la resolución DNS, calculando la diferencia de tiempos de cada uno de ellos para poder obtener el tiempo de resolución DNS.

```
C:\Windows\System32>nslookup www.google.com
Servidor: dns.google
Address: 2001:4860:4860::8888

Respuesta no autoritativa:
Nombre: www.google.com
Addresses: 2a00:1450:4001:82a::2004
           142.250.186.132
```

Figura 4-15 Herramienta nslookup

- En Linux, utilizamos la herramienta **dig** para realizar consultas DNS y obtener la dirección del dominio correspondiente. Esta herramienta incluye el tiempo que le ha llevado resolver ese dominio.

```
[# dig -6 www.google.com AAAA
; <>> DiG 9.20.2-1-Debian <>> -6 www.google.com AAAA
;; global options: +cmd
;; Got answer:
;; →HEADER← opcode: QUERY, status: NOERROR, id: 48846
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.google.com.           IN      AAAA
;;
;; ANSWER SECTION:
www.google.com.        300     IN      AAAA    2a00:1450:4001:82a::2004
;;
;; Query time: 56 msec
;; SERVER: 2001:4860:4860::8888#53(2001:4860:4860::8888) (UDP)
;; WHEN: Wed Jul 02 12:47:51 CEST 2025
;; MSG SIZE rcvd: 71
```

Figura 4-16 Herramienta dig

Estas herramientas son utilizadas en el proyecto para medir el tiempo de resolución DNS y poder evaluar el rendimiento de los servidores DNS en los distintos servicios implementados.

4.3 Consideraciones del entorno de prueba

Para garantizar una medición lo más exacta posible de estos parámetros de red, se han realizado bajo las siguientes condiciones:

- Uso de conexión cableada para evitar pérdidas y minimizar interferencias.
- Pruebas en entornos con recursos suficientes y hardware actualizado.
- Repetición de pruebas para verificar datos extraídos.
- Pruebas en red en momentos sin un tráfico descontrolado.

A través de estas condiciones impuestas y utilizando las herramientas previamente analizadas, podemos realizar un estudio de manera profesional de los distintos servicios de red para asegurarnos de su correcto funcionamiento.

5 RESULTADOS Y ANÁLISIS DE LAS MEDICIONES

5.1 DNS Optimo

Antes de realizar las mediciones de todos los parámetros de red, hemos realizado un estudio previo para identificar el servidor DNS que ofrece mayor rendimiento para cada servicio implementado. Para ello, llevamos a cabo las siguientes pruebas:

- Cálculo del RTT y la pérdida de paquetes como destino la IP del servidor DNS analizado.
- Medición de los tiempos de resolución DNS a diferentes dominios, almacenando la IP que nos devuelve en la resolución.
- Evaluación del RTT y la pérdida de paquetes teniendo como destino las IPs resultantes en la resolución.

Cada vez que se utilice un servidor DNS diferente, procedemos a borrar la cache DNS para evitar resultados falseados. En Windows, ejecutamos el comando **ipconfig /flushdns**. En Linux, se suelen usar comandos como **sudo systemd-resolve --flush-caches**.

5.1.1 Nativo

Para la red nativa con direccionamiento IPv6 ofrecido por la operadora, vamos a realizar el estudio previo del servidor DNS óptimo con tres candidatos. Estos son:

- DNS público de Google, con direcciones IPv6 **2001:4860:4860::8888** y **2001:4860:4860::8844**.
- DNS público de Cloudflare, con direcciones IPv6 **2606:4700:4700::1111** y **2606:4700:4700::1001**.
- DNS Nativo (operadora), con dirección Ipv6 **2a0c:5a80:0:2::1**.

El primer paso es calcular el RTT y el porcentaje de pérdida de paquetes teniendo como destino la dirección IP de cada servidor DNS que estamos analizando. Esta medición la realizamos a través de la herramienta PingPlotter, que nos proporciona una visión de la red en tiempo real proporcionándonos los parámetros que necesitamos.

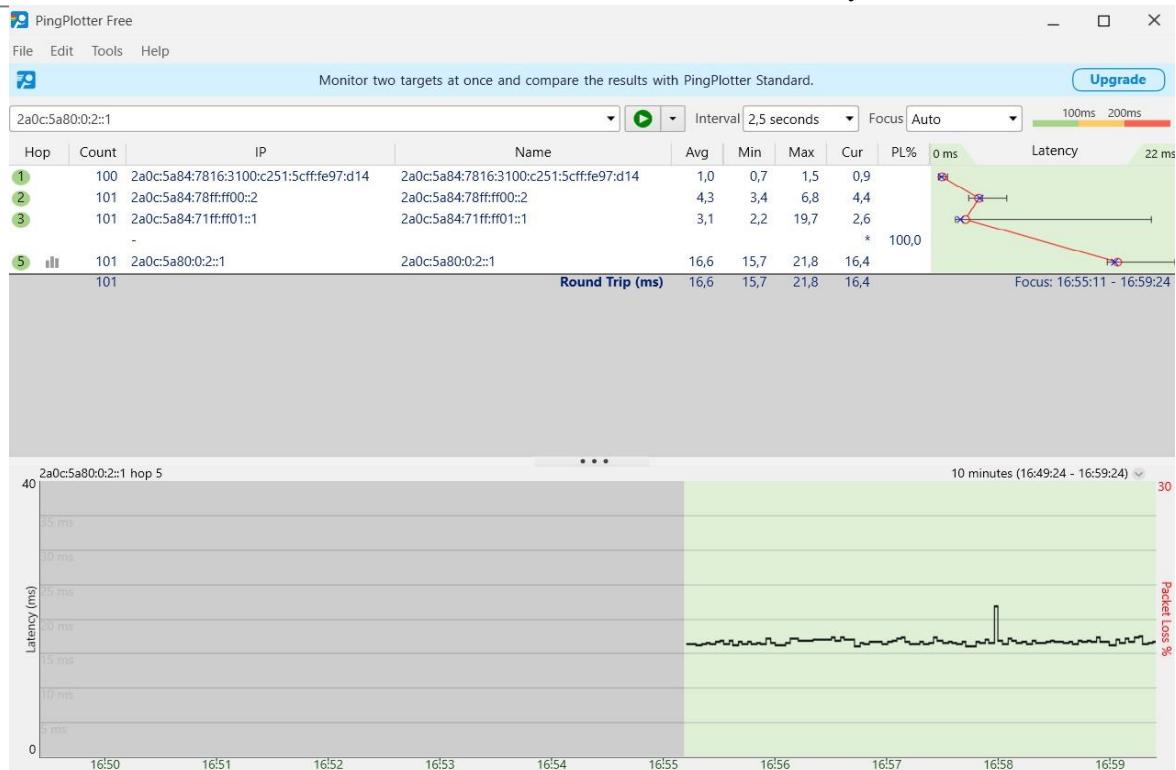


Figura 5-1 Resultado PingPlotter DNS IPv6 Nativo [34]

Para el RTT, tenemos la siguiente comparación:

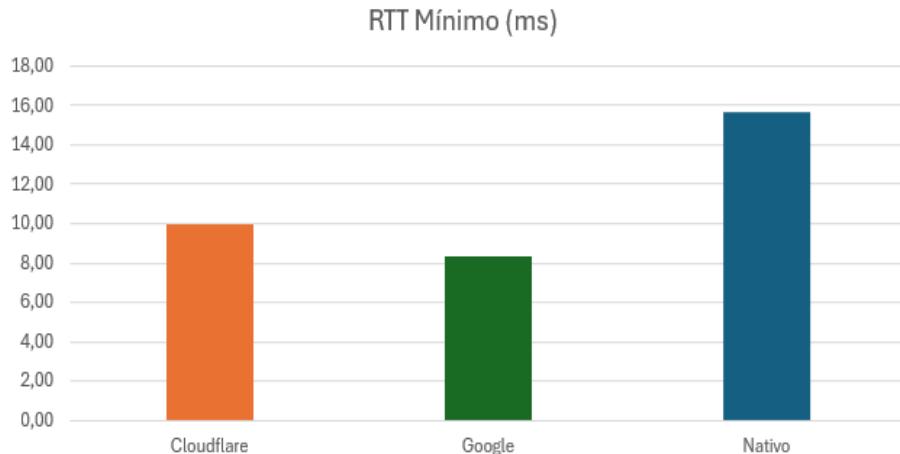


Figura 5-2 RTT Mínima DNS IPv6 Nativo

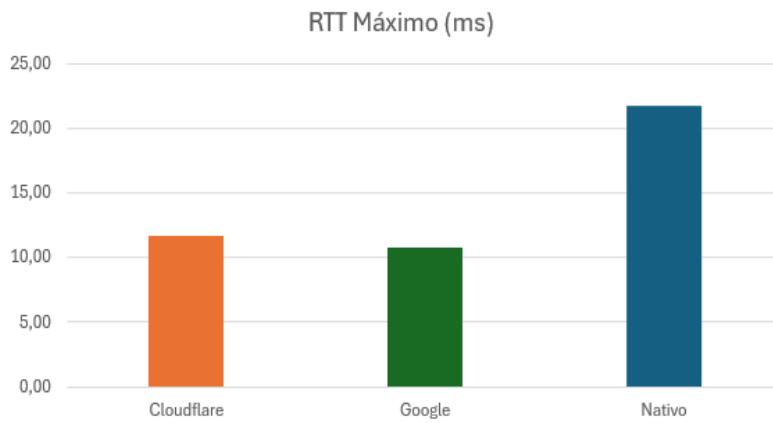


Figura 5-3 RTT Máxima DNS IPv6 Nativo

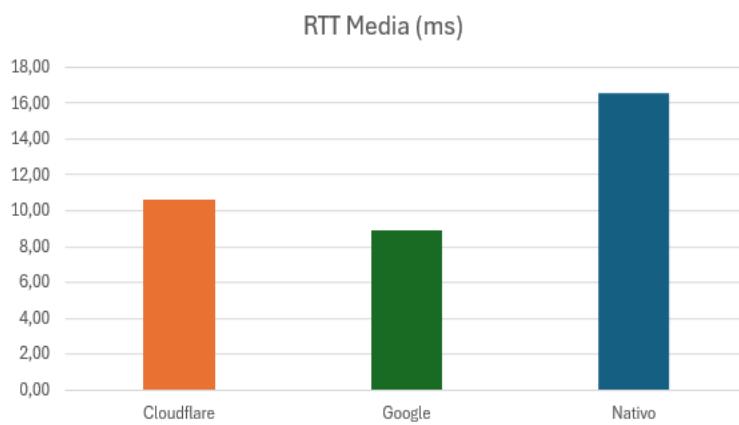


Figura 5-4 RTT Media DNS IPv6 Nativo

Tal como se muestra en las Figuras 5-2, 5-3 y 5-4, podemos observar que el servidor DNS de Google destaca ofreciendo valores de RTT más bajos que el resto de los servidores DNS.

Para la pérdida de paquetes, obtenemos la siguiente tabla:

Servidor	Porcentaje (%)
Nativo	0
Google	0
Cloudflare	0

Tabla 5-1 Paquetes Perdidos IPv6 Nativo

Observando la Tabla 5-1, todos los servidores ofrecen una conexión estable, con pérdidas del 0%, afirmando la estabilidad de cada servidor.

El siguiente paso es de obtener el tiempo de resolución DNS, resolviendo los dominios de varios servicios. Estos servicios son Google.com, Facebook.com y Wikipedia.com.

Los tiempos de resolución para cada destino se muestran en la siguiente gráfica:

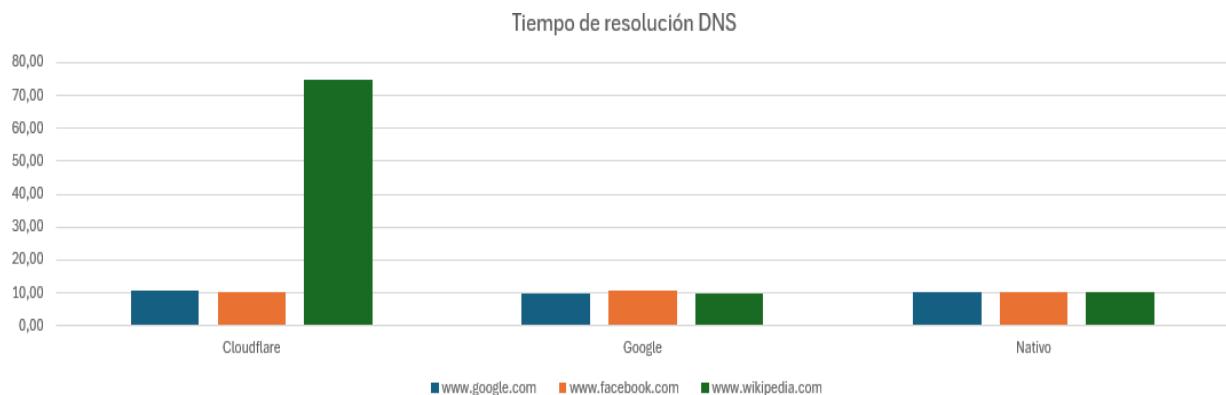


Figura 5-5 Tiempo de resolución DNS IPv6 Nativo

A través de la Figura 5-5, podemos observar como todos los servidores DNS están bastante equilibrados. Sin embargo, Google y el servidor Nativo de la operadora ofrecen tiempos de resolución algo más bajos y estables en comparación a Cloudflare.

Por último, nos queda obtener el tiempo de ida y vuelta de cada dominio resuelto previamente. Este procedimiento vamos a representarlo en la siguiente gráfica:

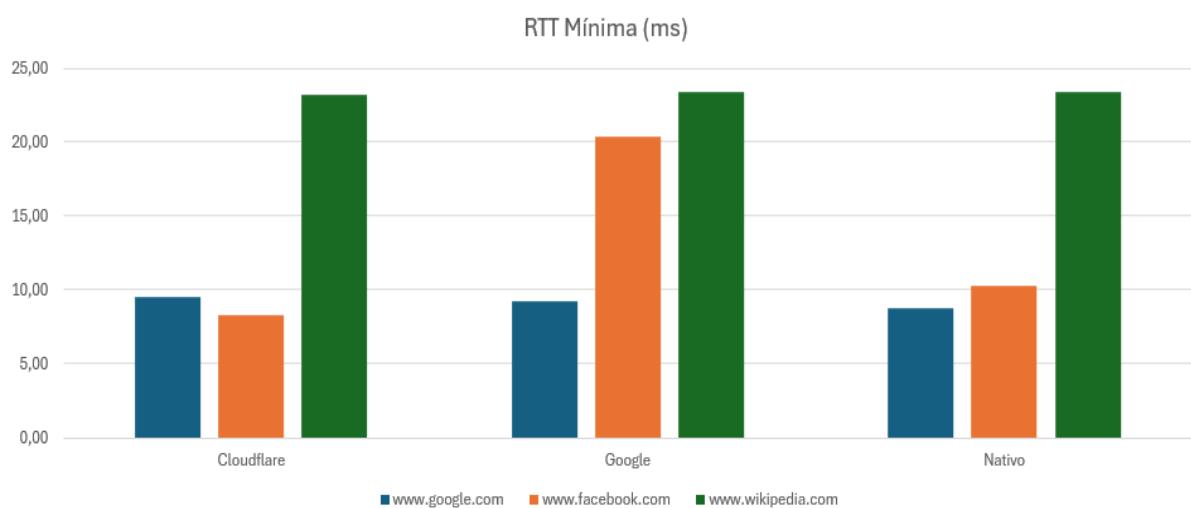


Figura 5-6 RTT Mínima Dominios IPv6 Nativo

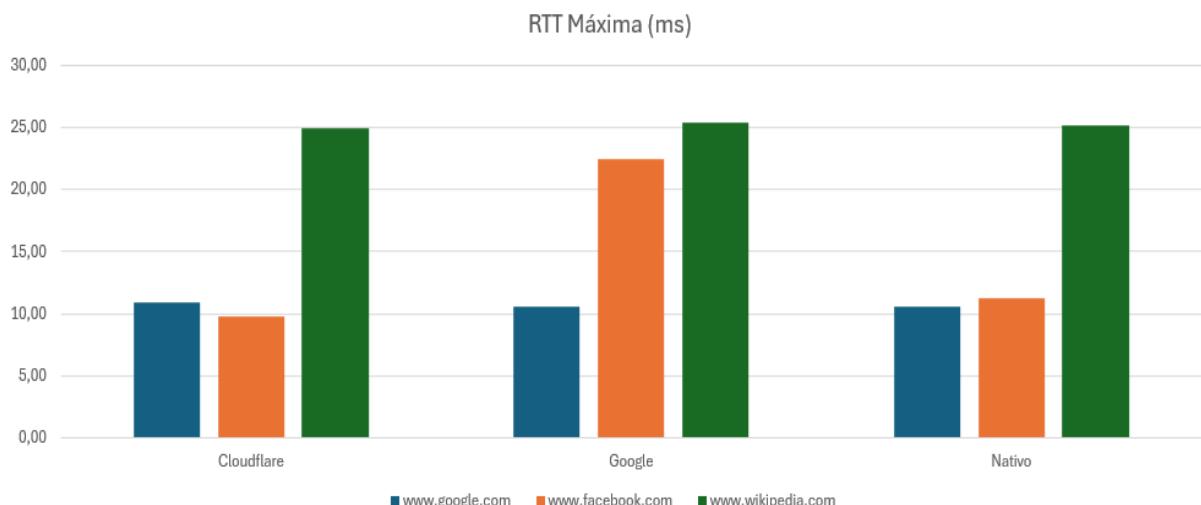


Figura 5-7 RTT Máxima Dominios IPv6 Nativo

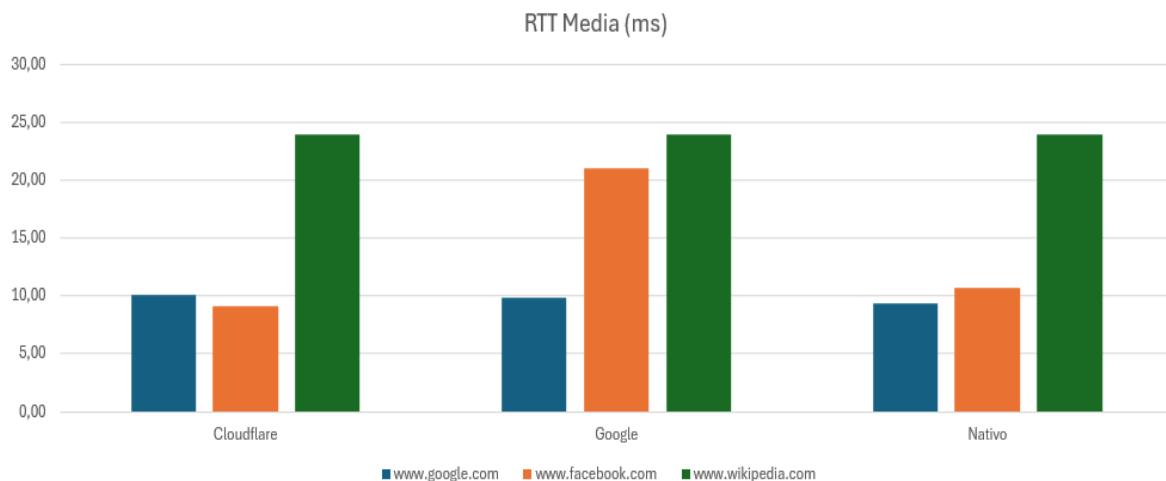


Figura 5-8 RTT Media Dominios IPv6 Nativo

En las Figuras 5-6, 5-7, 5-8, el servidor DNS Nativo ofrece mayor equilibrio, aunque todos poseen buenas velocidades y un excelente rendimiento.

En conclusión, tras analizar todos los datos y sus correspondientes comparaciones, concluimos que el servidor DNS con mayor rendimiento para este servicio es el servidor DNS Nativo. Este servidor DNS con dirección IP **2a0c:5a80:0:2::1** ofrece el mejor rendimiento global en el entorno IPv6, por lo que va a ser el servidor seleccionado para las pruebas posteriores.

5.1.2 TunnelBroker de Hurricane Electric implementado en Windows

Para elegir el servidor DNS óptimo en el servicio proporcionado por Hurricane Electric implementado en el sistema operativo Windows, vamos a utilizar los siguientes servidores DNS:

- Servidor DNS público de Google, con direcciones IPv6 **2001:4860:4860::8888** y **2001:4860:4860::8844**.
- Servidor DNS público de Cloudflare, con direcciones IPv6 **2606:4700:4700::1111** y **2606:4700:4700::1001**.
- Servidor DNS Native de Hurricane Electric, con dirección IPv6 **2001:470:20::2**.

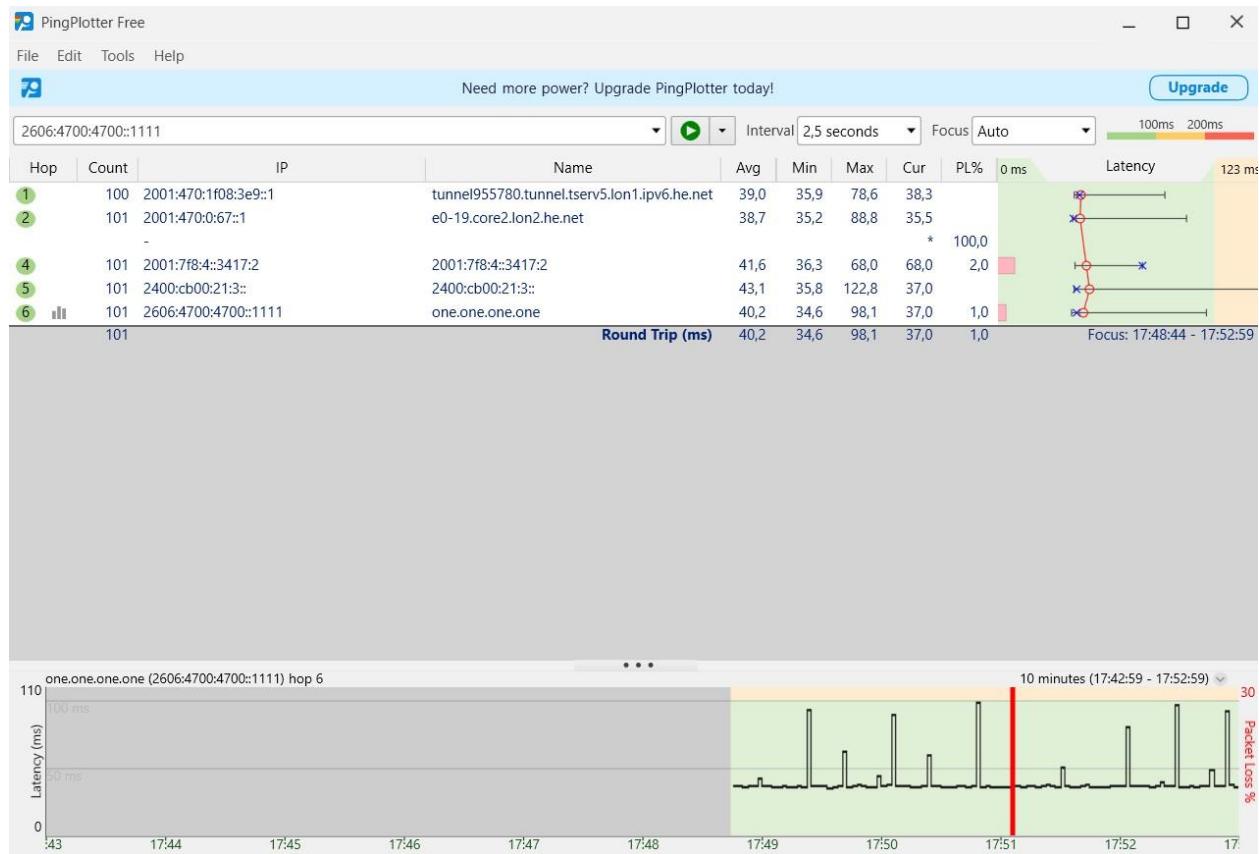


Figura 5-9 Resultado PingPlotter IPv6 Hurricane Electric [34]

Para el cálculo del RTT y la pérdida de paquetes a la dirección IP destino la de cada servidor DNS que estamos analizando, obtenemos la siguiente donde se representan los tiempos máximos, mínimos y la media realizada.

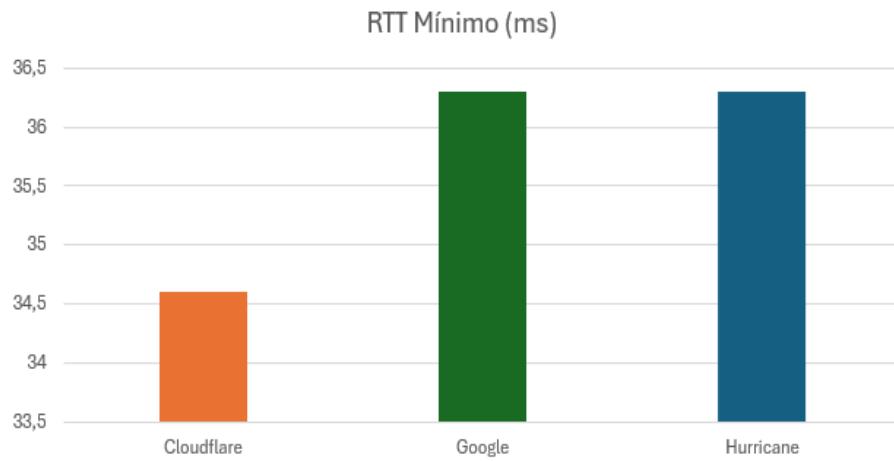


Figura 5-10 RTT Mínima DNS Hurricane Electric Windows

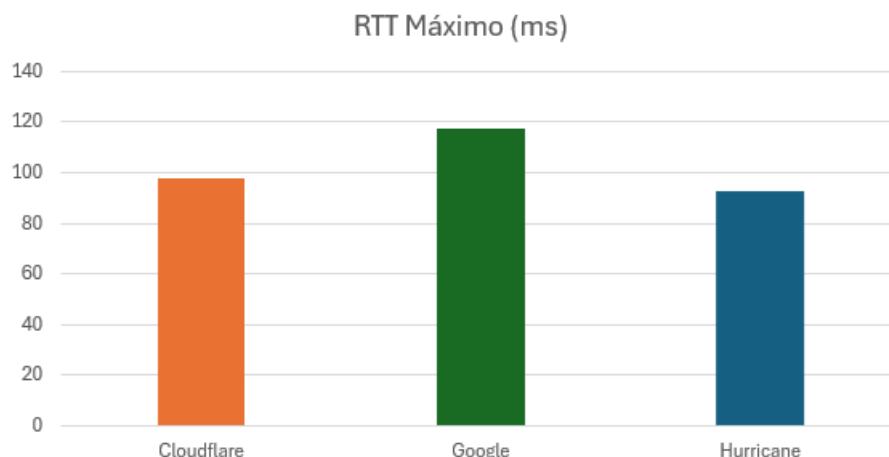


Figura 5-11 RTT Máxima DNS Hurricane Electric Windows

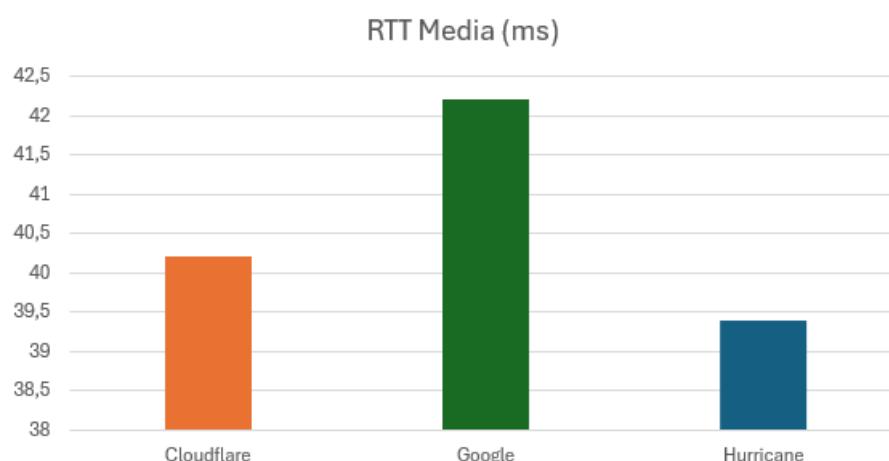


Figura 5-12 RTT Media DNS Hurricane Electric Windows

Observamos en las Figuras 5-10, 5-11 y 5-12 como el servidor que ofrece un RTT más bajo es el servidor Nativo de Hurricane Electric, aunque todos los servidores manejan tiempos bastante parecidos.

Para la pérdida de paquetes, obtenemos la siguiente tabla donde se muestra el porcentaje de pérdida de paquetes de cada servidor DNS:

Servidor	Porcentaje (%)
Hurricane	0
Google	0
Cloudflare	0

Tabla 5-2 Paquetes Perdidos Ipv6 Hurricane Electric Windows

Tras ver la Tabla 5-2, vemos como todos los servidores ofrecen pérdidas del 0%, afirmando que presentan una conexión estable.

Para el tiempo de resolución DNS, utilizamos como destino los dominios Google.com, Facebook.com y Wikipedia.com, comparando la eficiencia de cada servidor DNS.

Esta comparación es representada en la siguiente gráfica:

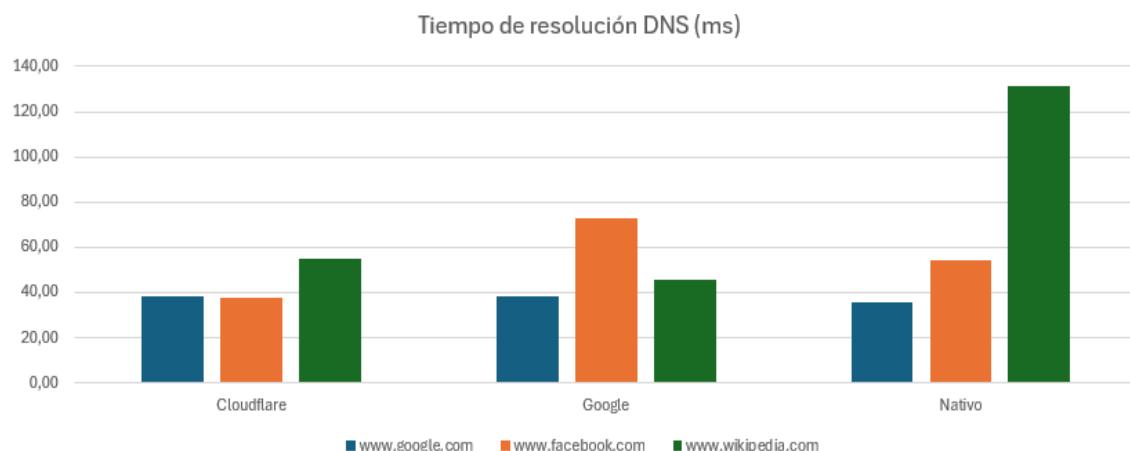


Figura 5-13 Tiempo de resolución DNS Hurricane Electric Windows

En términos generales, observando la Figura 5-13, vemos como el servidor DNS de Cloudflare es el servidor que ofrece los menores tiempos de resolución.

Para los tiempos de ida y vuelta hacia la IP obtenida por las resoluciones anteriores, obtenemos las siguientes gráficas

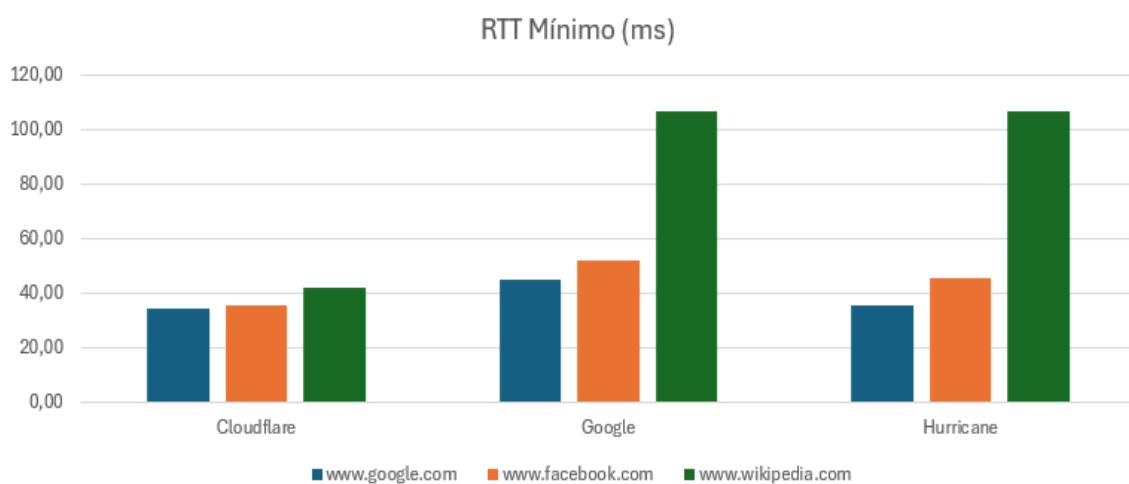


Figura 5-14 RTT Mínima Dominios Hurricane Electric Windows

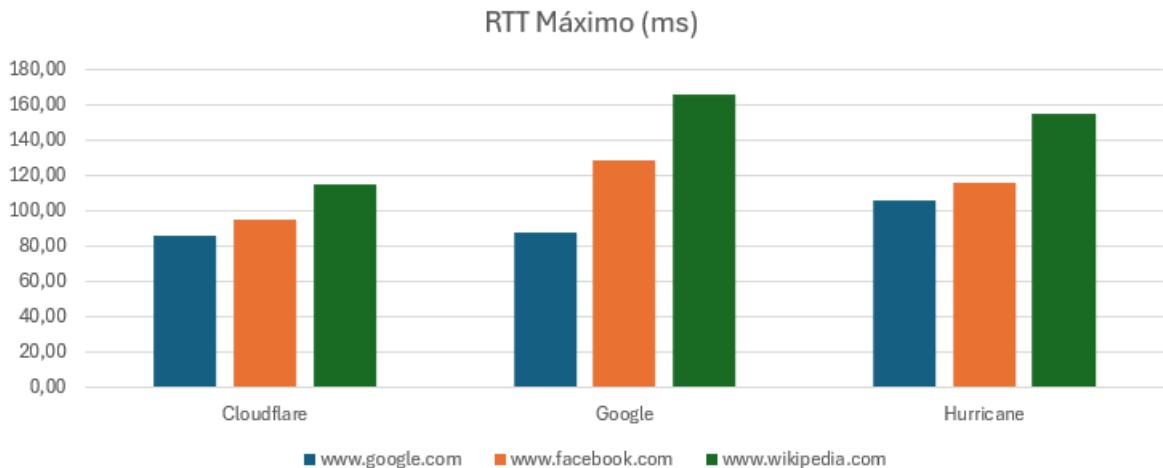


Figura 5-15 RTT Máxima Dominios Hurricane Electric Windows

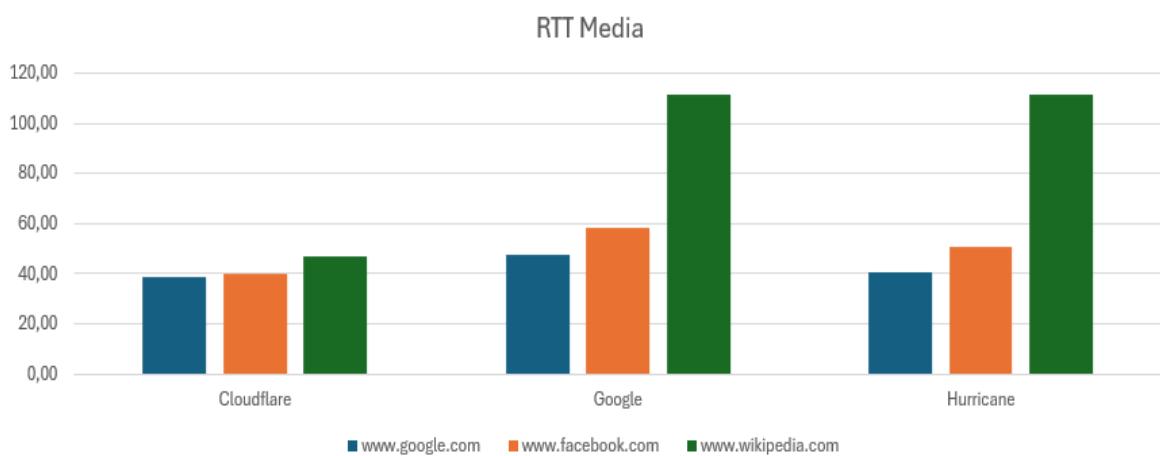


Figura 5-16 RTT Media Dominios Hurricane Electric Windows

De nuevo, observando las Figuras 5-14, 5-15, 5-16, el servidor DNS de Cloudflare es el que ofrece los mejores tiempos totales para los dominios analizados, ofreciendo un tiempo medio aceptable sin picos demasiado exagerados.

En conclusión, en base a los resultados obtenidos, podemos concluir que el servidor DNS óptimo y que permite al servicio ofrecer su mayor rendimiento es el servidor DNS de Cloudflare con dirección IPv6 **2606:4700:4700::1111**.

5.1.3 TunnelBroker de Hurricane Electric implementado en Ubuntu

Para el servicio de Hurricane Electric implementado en un sistema operativo Linux, vamos a utilizar los siguientes servidores DNS:

- Servidor DNS público de Google, con direcciones IPv6 **2001:4860:4860::8888** y **2001:4860:4860::8844**.
- Servidor DNS público de Cloudflare, con direcciones IPv6 **2606:4700:4700::1111** y **2606:4700:4700::1001**.
- Servidor DNS Nativo de Hurricane Electric, con dirección IPv6 **2001:470:20::2**.

Para todo este procedimiento de elegir el servidor DNS óptimo, utilizamos el script comentado anteriormente que automatiza todo el proceso eliminando la necesidad de realizar varias pruebas.

En cuanto al tiempo de ida y vuelta, representamos los tiempos máximos, mínimos y su media en las siguientes gráficas:

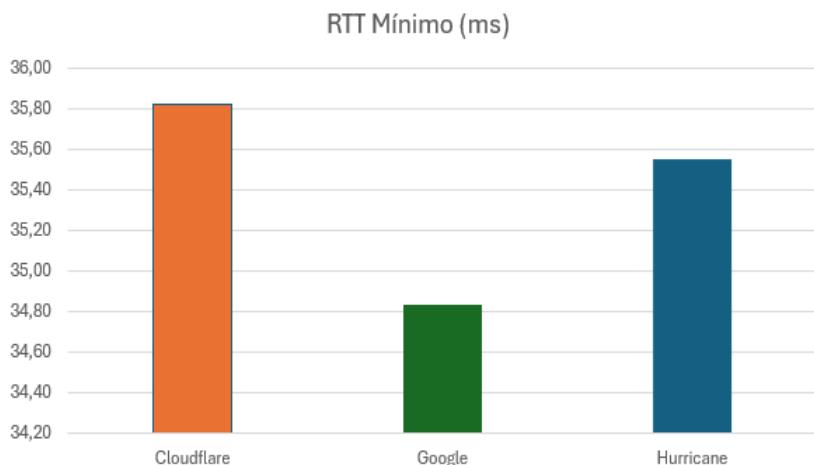


Figura 5-17 RTT Mínima DNS Hurricane Electric Ubuntu

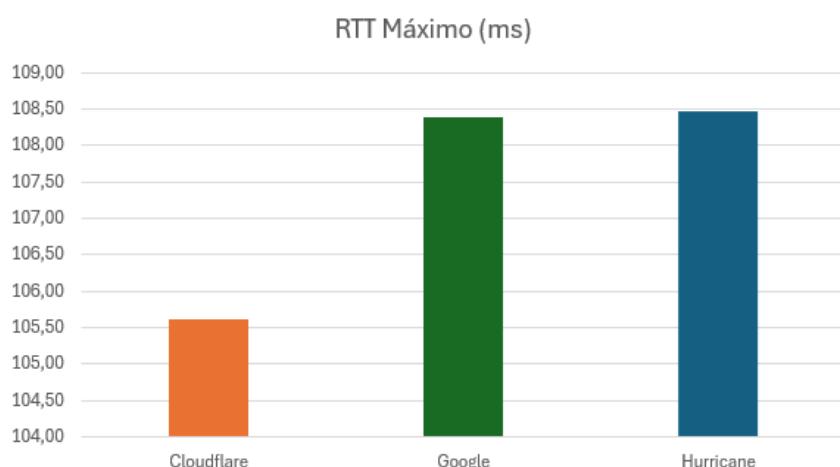


Figura 5-18 RTT Máxima DNS Hurricane Electric Ubuntu

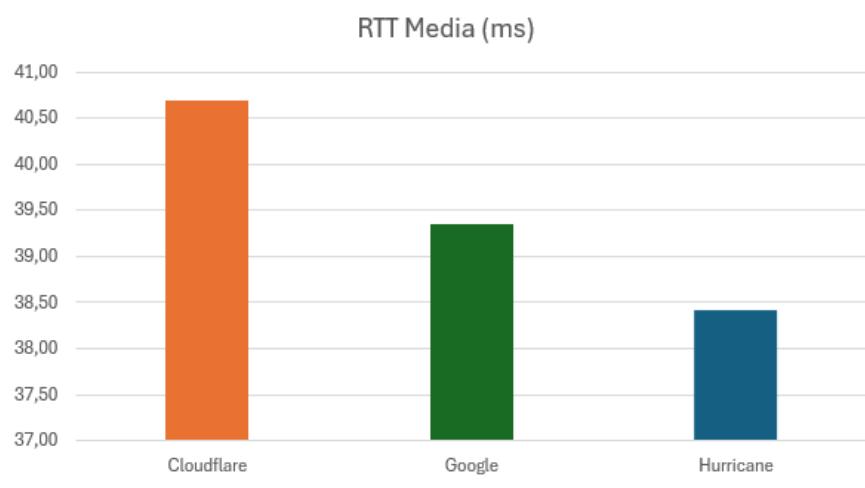


Figura 5-19 RTT Media DNS Hurricane Electric Ubuntu

Como podemos apreciar en las Figuras 5-17, 5-18, 5-19, el servidor DNS de Hurricane Electric es el que ofrece unos tiempos menores, aunque son tiempos bastante equilibrados en cada servidor. Para la pérdida

de paquetes, obtenida también del script, obtenemos la siguiente tabla donde se representan los diferentes porcentajes de paquetes perdidos:

Para la pérdida de paquetes, obtenida también del script, obtenemos la siguiente tabla donde se representan los diferentes porcentajes de paquetes perdidos:

Servidor	Porcentaje (%)
Hurricane	0
Google	0
Cloudflare	0

Tabla 5-3 Paquetes Perdidos Ipv6 Hurricane Electric Ubuntu

Observando la Tabla 5-3, afirmamos que todos los servidores DNS ofrecen una conectividad estable con un porcentaje de pérdida de paquetes del 0%.

Para el tiempo de resolución DNS, hemos utilizado como destino los dominios Google.com, Facebook.com y Wikipedia.com, pasándolo como argumento al script.



Figura 5-20 Tiempos de resolución DNS Hurricane Electric Ubuntu

Tal como se refleja en la Figura 5-20, los tiempos están bastante equilibrados en todos los servidores DNS, sin una diferencia significativa, aunque cada uno destaca ligeramente en uno de los dominios.

El último paso es calcular los distintos tiempos de ida y vuelta de las IP obtenidas por las resoluciones anteriores. Una vez obtenidos los datos, los representamos en las siguientes gráficas:

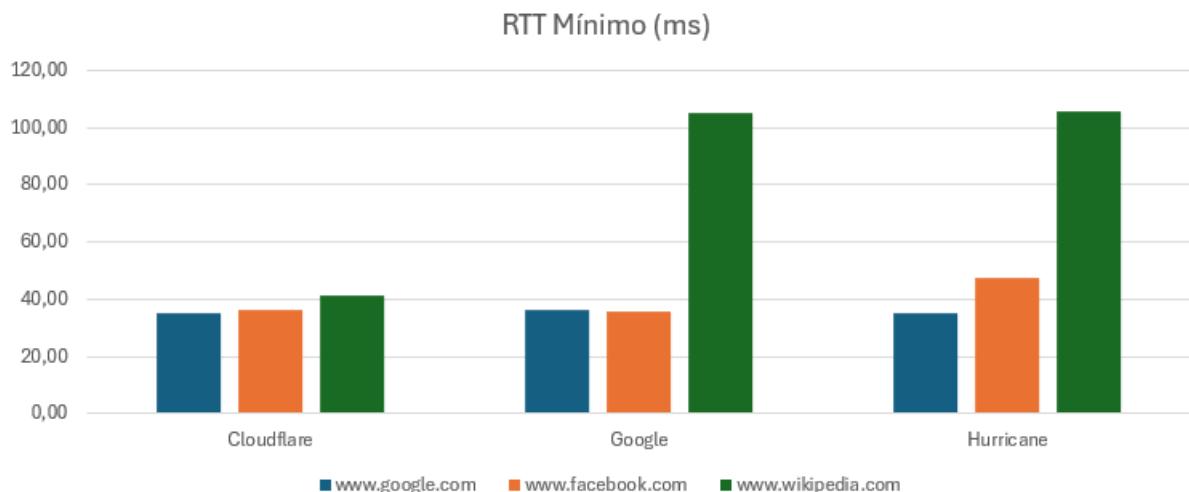


Figura 5-21 RTT Mínima Dominios Hurricane Electric Ubuntu

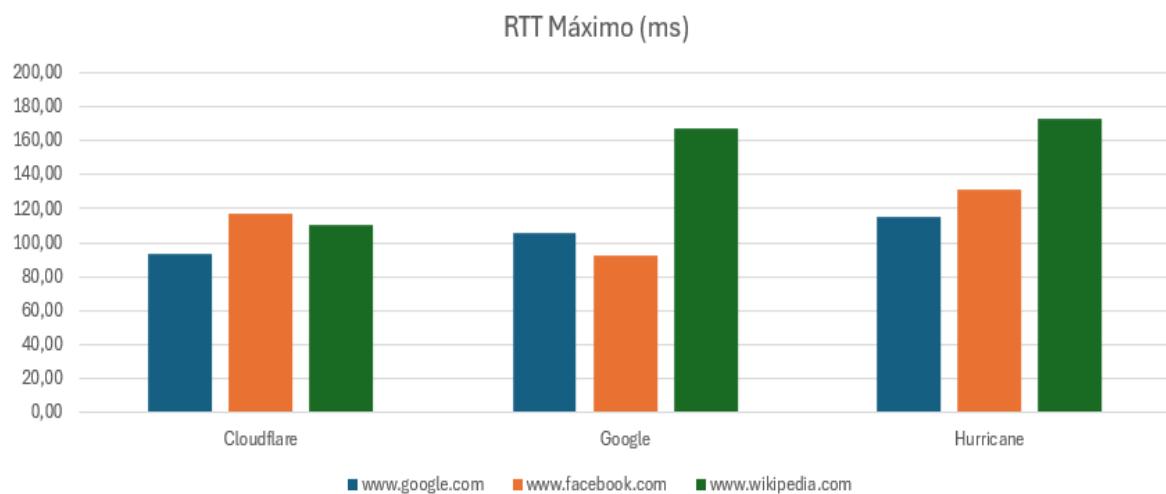


Figura 5-22 RTT Máxima Dominios Hurricane Electric Ubuntu

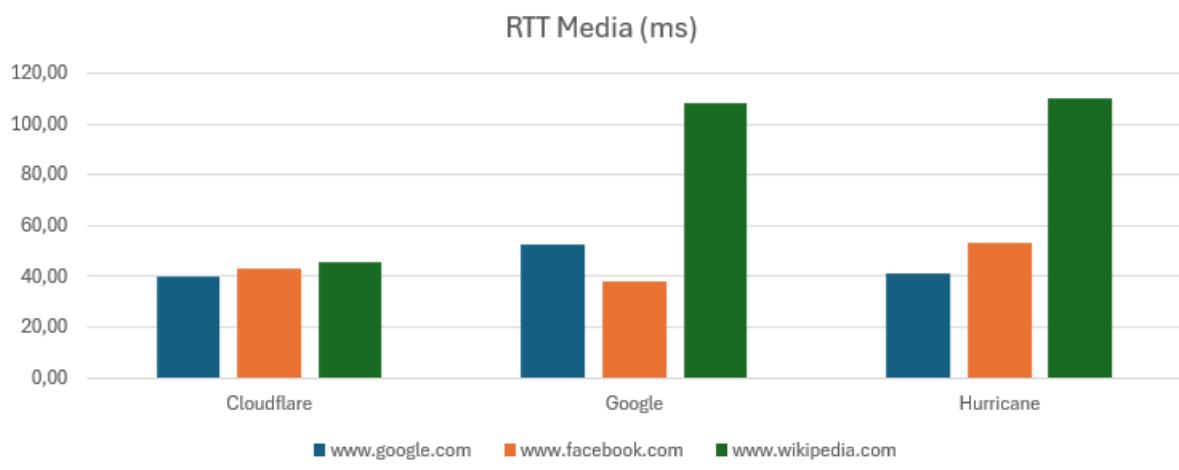


Figura 5-23 RTT Media Dominios Hurricane Electric Ubuntu

Analizando las Figuras 5-21, 5-22, 5-23, observamos como el servidor DNS de Cloudflare ofrece los tiempos más bajos en todas las pruebas, posicionándolo como la opción para obtener el mayor rendimiento del servicio.

Finalmente, nos decantamos por el servidor DNS de Cloudflare como servidor para ser implementado en este servicio que destaca por su fiabilidad y su gran rendimiento.

5.1.4 VPN de Hide.me

A diferencia de los servicios anteriores, para la VPN implementada en este proyecto no es posible configurar manualmente un servidor DNS, ya que la resolución de nombres es un procedimiento realizado por la propia infraestructura de la VPN.

Dado que no se puede realizar una comparación de varios servidores DNS para obtener el que ofrezca mayor rendimiento, vamos a utilizar el servidor DNS nativo proporcionado por la VPN con dirección IP **fd00:6968:6564:cc::1**.

5.1.5 Wireguard+VPS

En el servicio donde se implementa una conexión VPN a través de Wireguard, utilizando como servidor una VPS contratada, vamos a utilizar los siguientes servidores DNS:

- Servidor DNS público de Google, con direcciones IPv6 **2001:4860:4860::8888** y **2001:4860:4860::8844**.
- Servidor DNS público de Cloudflare, con direcciones IPv6 **2606:4700:4700::1111** y **2606:4700:4700::1001**.
- Servidor DNS público de Quad9, con dirección IPv6 **2620:fe::fe**.

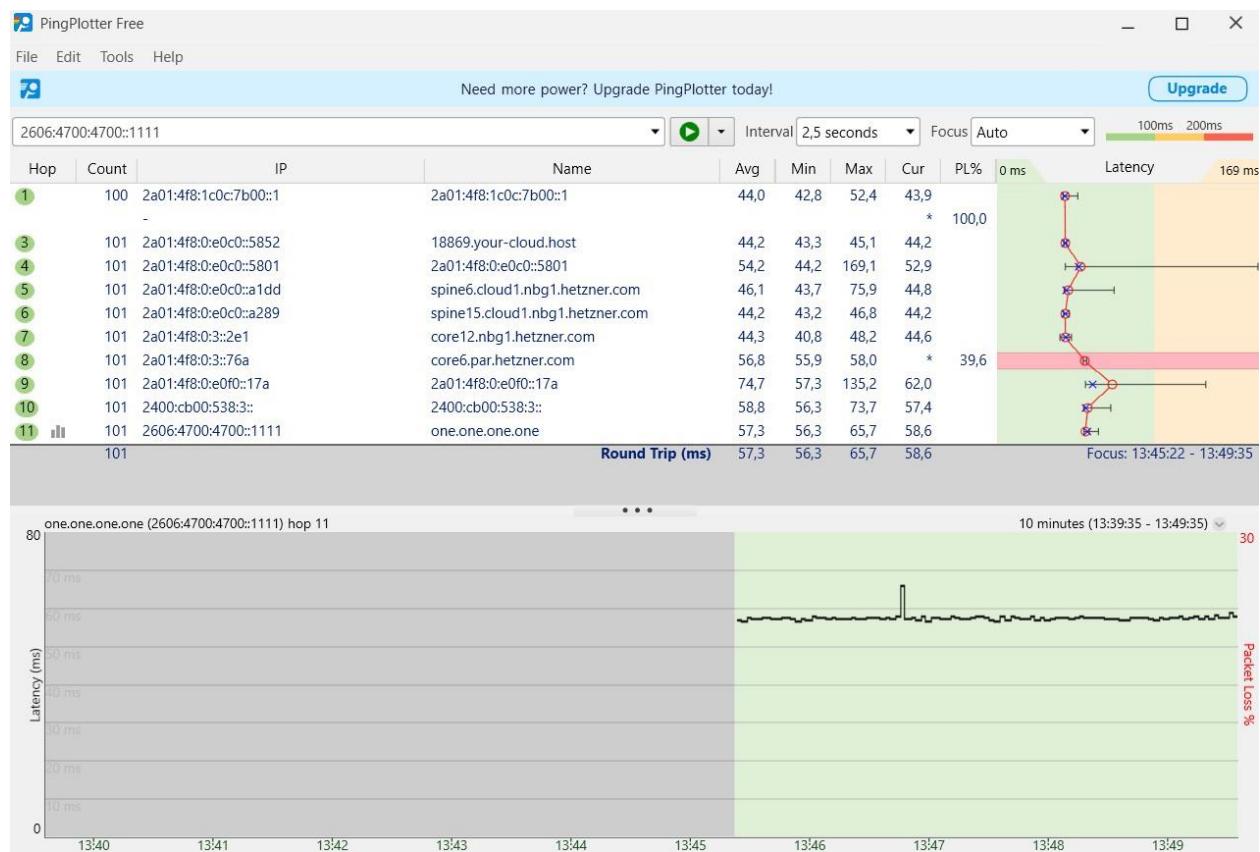


Figura 5-24 Resultado PingPlotter DNS VPN Wireguard [34]

En primer lugar, realizamos la medición de los tiempos de ida y vuelta y la pérdida de paquetes con destino la IP de cada servidor DNS. Para este análisis utilizamos la herramienta PingPlotter.

El tiempo RTT máximo, mínimo y su media podemos observarlas en las siguientes gráficas:

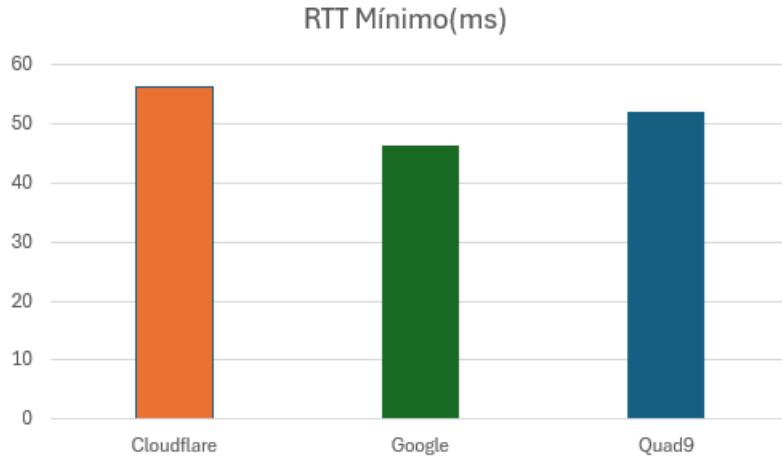


Figura 5-25 RTT Mínima DNS VPN Wireguard

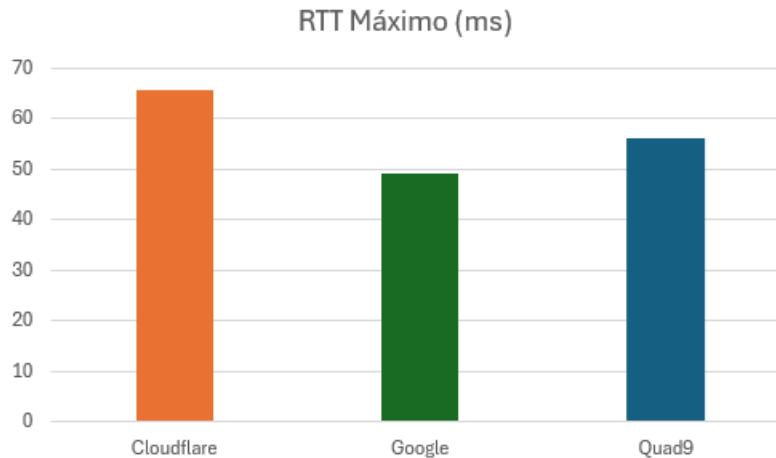


Figura 5-26 RTT Máxima DNS VPN Wireguard

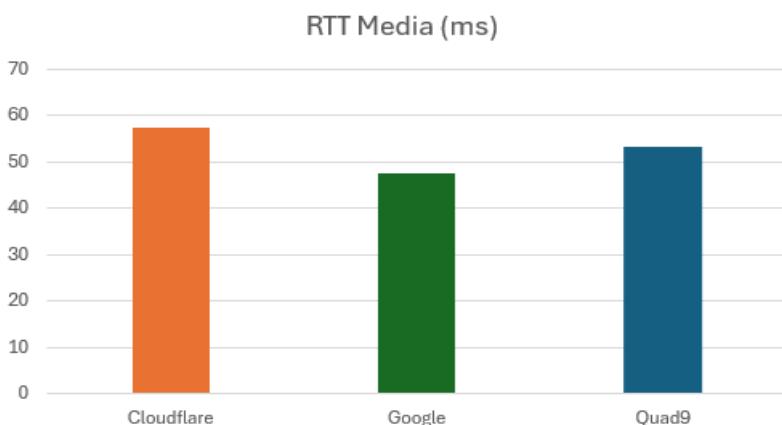


Figura 5-27 RTT Media DNS VPN Wireguard

A través de las Figuras 5-25, 5-26, 5-27, observamos como los resultados muestran que el servidor DNS de Google ofrece los tiempos más bajos, además de una mayor estabilidad.

Para la pérdida de paquetes obtenemos la siguiente tabla:

Servidor	Porcentaje (%)
Quad9	0
Google	0
Cloudflare	0

Tabla 5-4 Paquetes Perdidos VPN Wireguard

A partir de la Tabla 5-4, vemos como todos los servidores DNS ofrecen pérdidas del 0%, por lo que todos son válidos desde el punto de vista de la estabilidad.

Para los tiempos de resolución DNS, utilizamos como dominios a Google.com, Facebook.com y Wikipedia.com. Sus tiempos son medidos y representados en la siguiente gráfica:

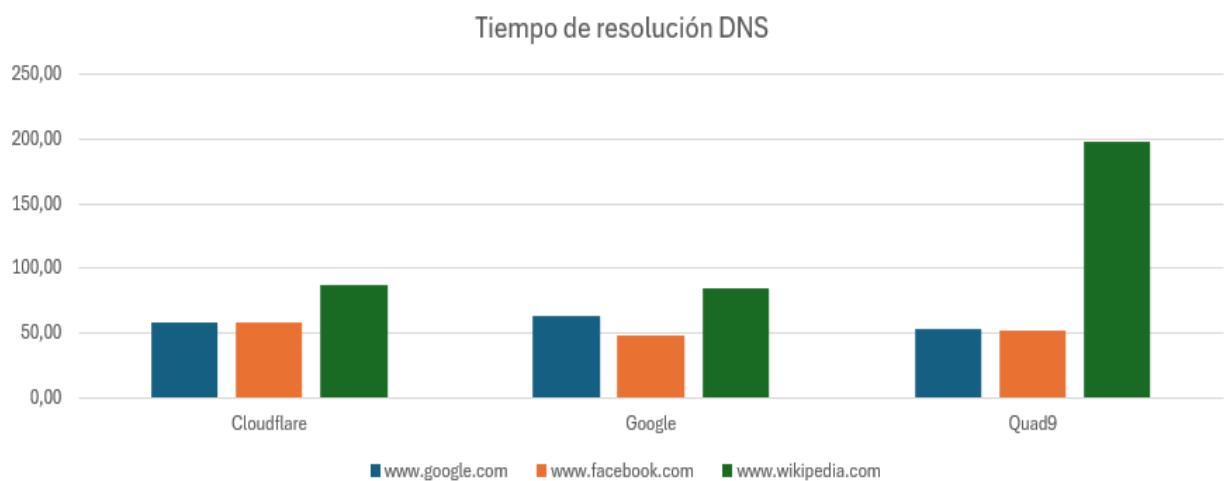


Figura 5-28 Tiempo de resolución DNS VPN Wireguard

Tal como indica la Figura 5-28, los tiempos de resolución son bastante equilibrados en cada servidor, aunque el servidor de Google ofrece un rendimiento algo mejor.

Para terminar el estudio, medimos los tiempos de ida y vuelta de las IP obtenidas en las resoluciones anteriores. Los representamos en las siguientes gráficas:

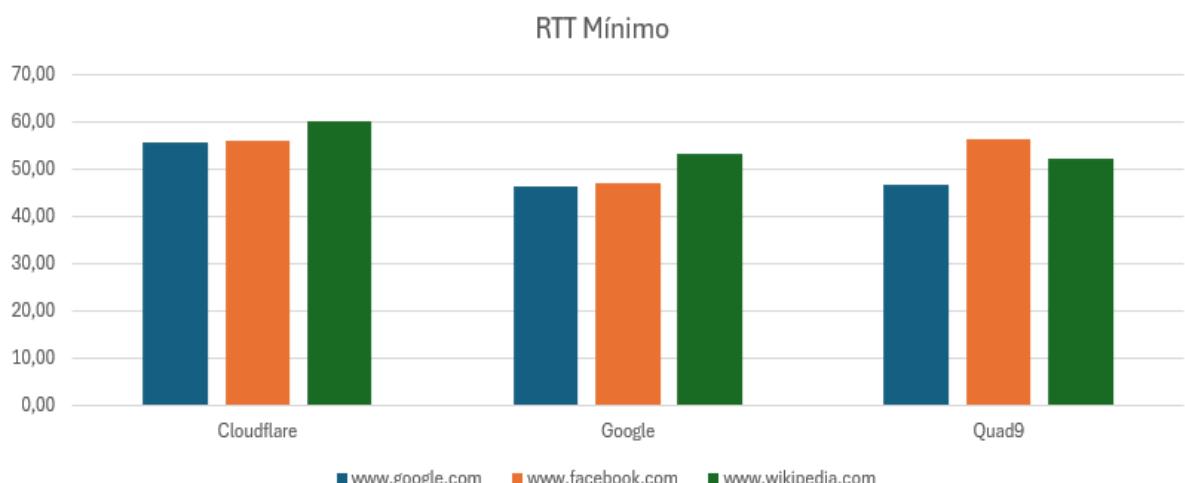


Figura 5-29 RTT Mínima Dominios VPN Wireguard

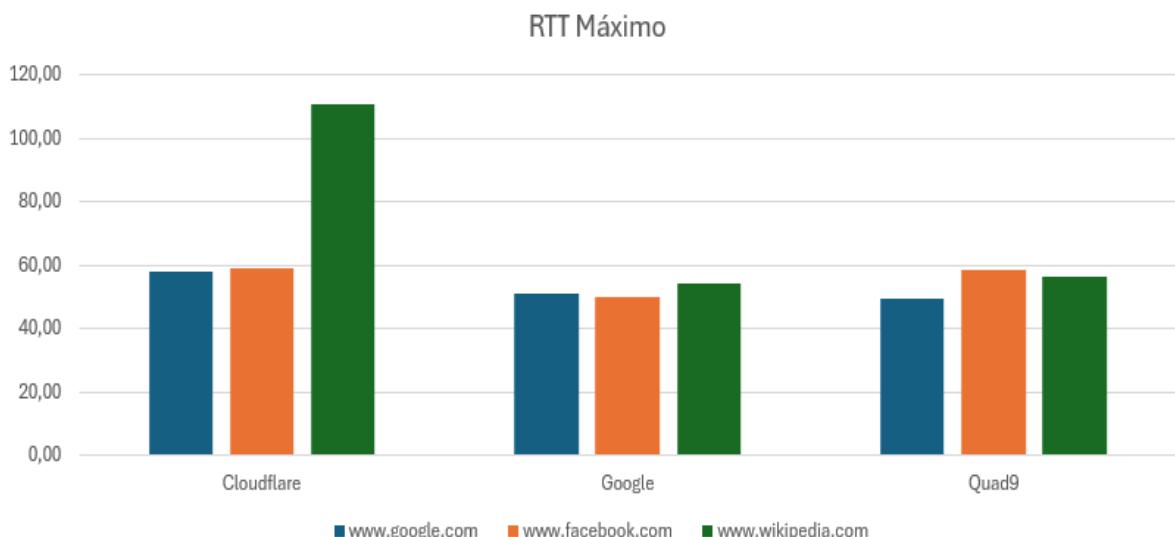


Figura 5-30 RTT Máxima Dominios VPN Wireguard

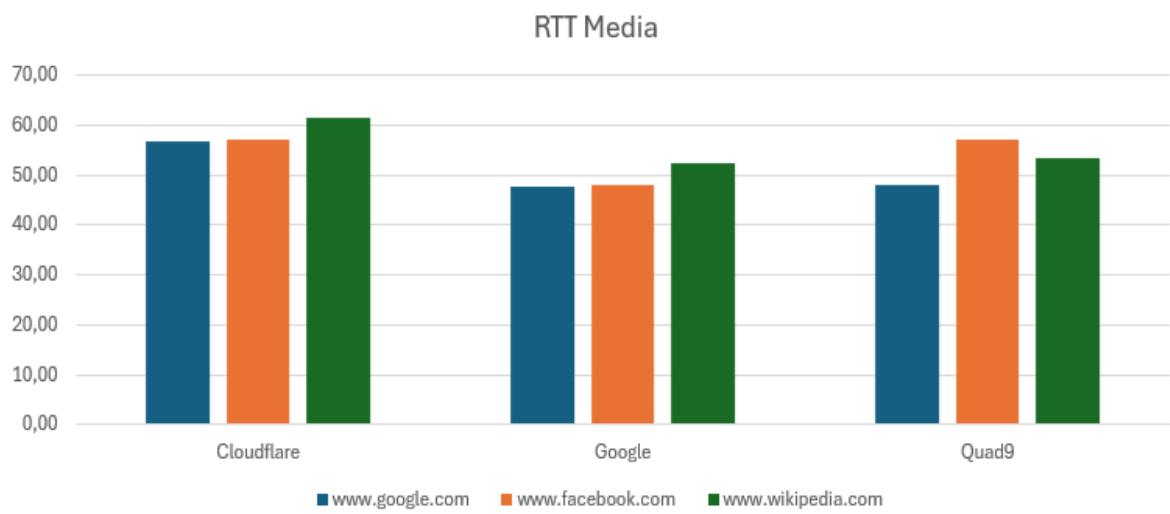


Figura 5-31 RTT Media Dominios VPN Wireguard

Como podemos apreciar en las Figuras 5-29, 5-30, 5-31, nuevamente el servidor DNS de Google mantiene un rendimiento más equilibrado que el resto de los servidores DNS.

En conclusión, el servidor DNS óptimo y que ofrece mejor rendimiento global es el servidor de Google. Para ello, va a ser seleccionado como servidor para ser configurado en el servicio para las pruebas y comparativas posteriores.

5.2 Comparación de servicios

Una vez realizada la comparación de los distintos servidores DNS para obtener el servidor óptimo que proporcione el mayor rendimiento posible al servicio, pasamos a comparar los servicios entre sí para ver las diferencias más notorias y llegar a una conclusión en base a todas las comparaciones realizadas.

5.2.1 RTT

El primer parámetro evaluado es el RTT. Para llevar a cabo esta prueba, empleamos las siguientes herramientas:

- Para el sistema operativo Windows hemos utilizado las herramientas PingPlotter y NetScanTools.

- Para Linux las herramientas utilizadas son Mtr y Ping.

Los resultados máximos, mínimos y su correspondiente media, lo obtenemos a través de la herramienta NetScanTools para Windows y Ping para Linux. Estos resultados se muestran en la siguiente gráfica, donde aparecen todos los servicios con su correspondiente servidor DNS configurado:

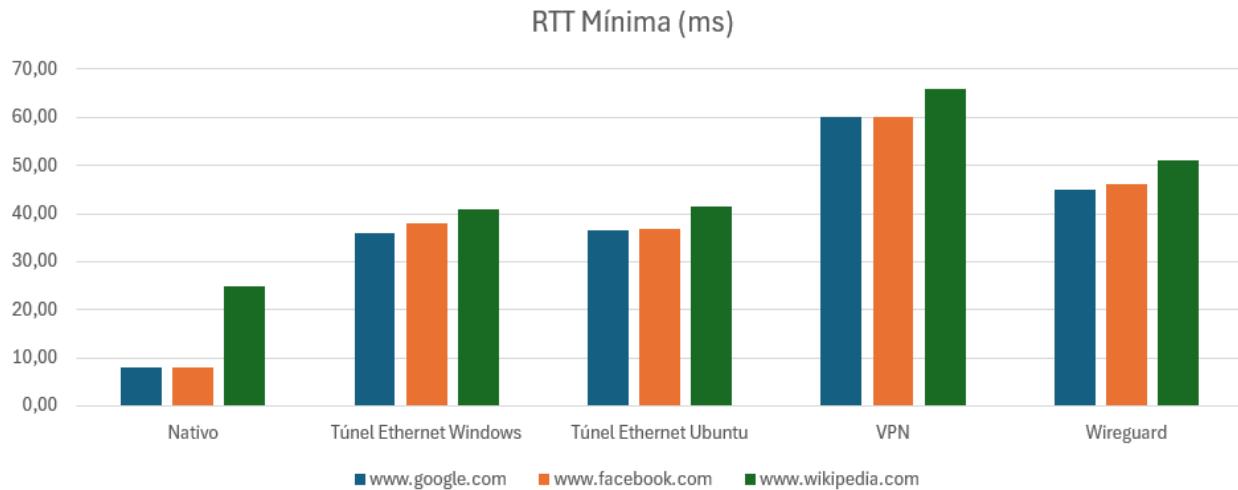


Figura 5-32 RTT Mínima Servicios NetScanTools/Ping

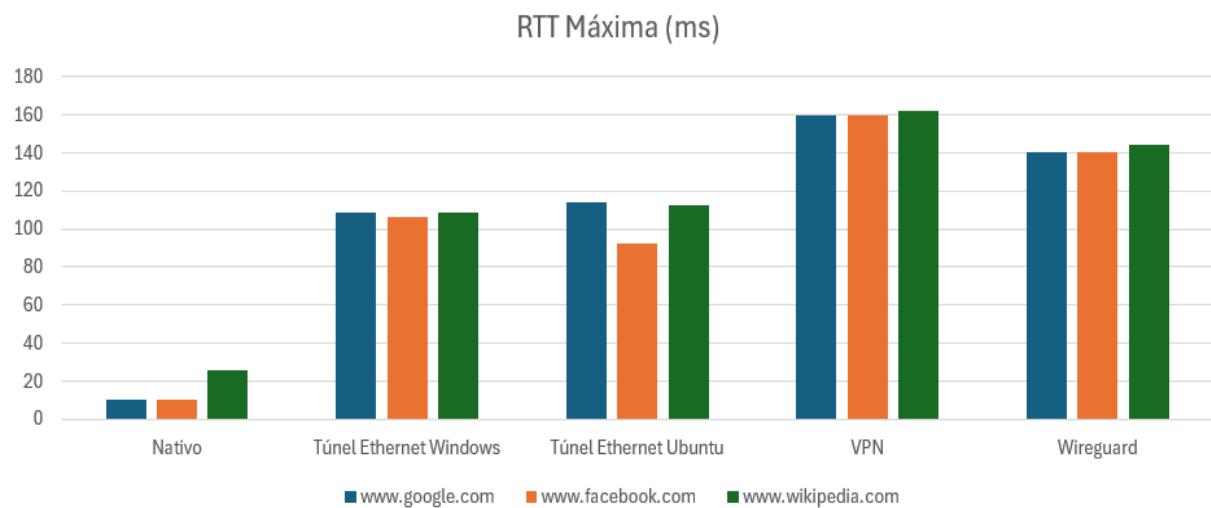


Figura 5-33 RTT Máxima Servicios NetScanTools/Ping

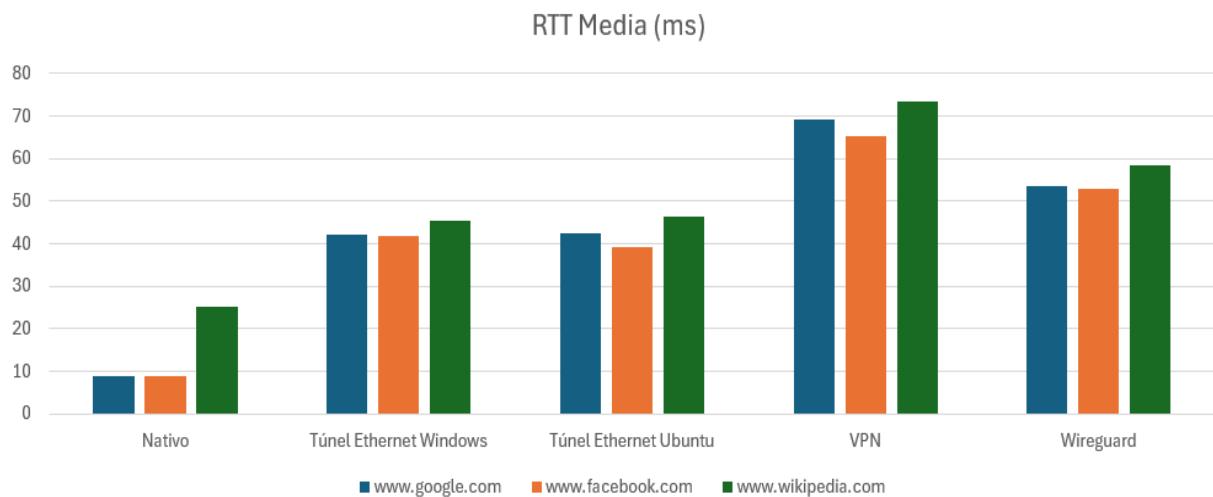


Figura 5-34 RTT Media Servicios NetScanTools/Ping

Tal como se indica en las Figuras 5-32, 5-33, 5-34, el servicio Nativo de la operadora siempre va a ofrecer velocidades más rápidas que cualquier otro servicio. En ausencia de este servicio, podemos observar que los túneles de Hurricane Electric, tanto en Windows como en Linux, ofrecen resultados aceptables, con una latencia media menor que otros servicios implementados como Wireguard. El inconveniente reside en su inestabilidad, ya que presentan mayor variabilidad en los tiempos de respuesta.

El servicio Wireguard+VPS presenta un comportamiento estable, sin picos de latencia, resultando en una opción ideal y llamativa.

La VPN ofrece una gran estabilidad, sin picos máximos ni mínimos, aunque sus tiempos son algo más elevados que el resto de los servicios.

Utilizando la herramienta PingPlotter para Windows y Mtr para Linux, podemos calcular los tiempos de ida y vuelta máximos, mínimos y su media. A través de las siguientes gráficas podemos representar los siguientes datos:

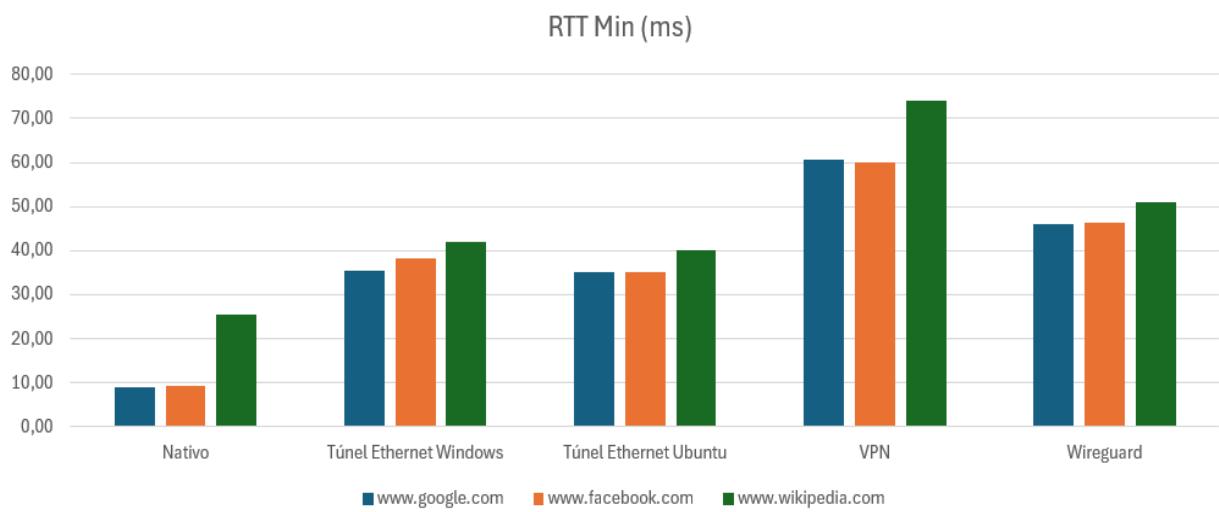


Figura 5-35 RTT Mínima Servicios PingPlotter/Mtr

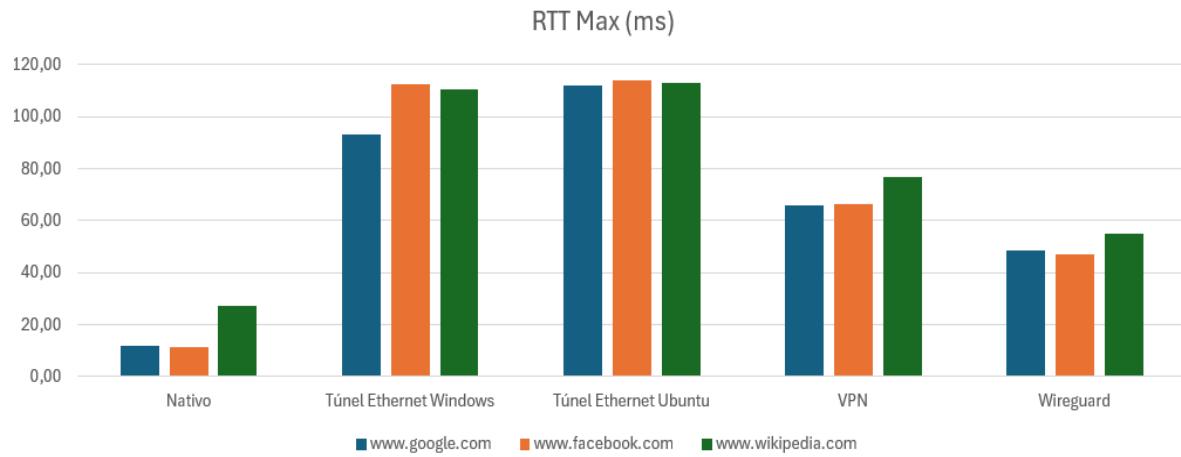


Figura 5-36 RTT Máxima Servicios PingPlotter/Mtr

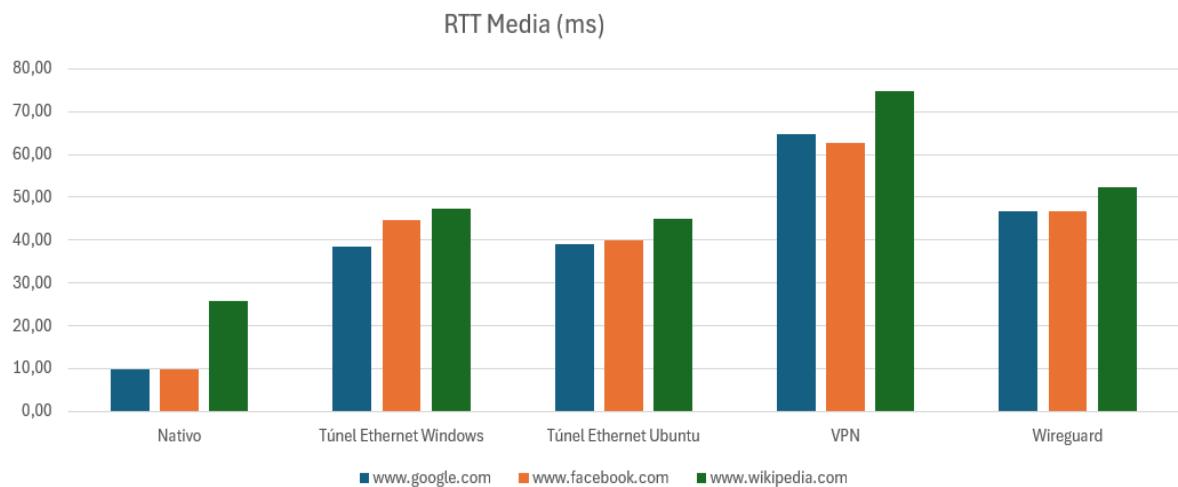


Figura 5-37 RTT Media Servicios PingPlotter/Mtr

A partir de las Figuras 5-35, 5-36, 5-37, podemos observar nuevamente la calidad y garantía del servicio nativo, aunque el servicio Wireguard+VPS ofrece una gran estabilidad, sin oscilaciones ni picos de latencia.

5.2.2 Pérdida de paquetes y ruta utilizada para cada destino

Para la pérdida de paquetes, utilizamos las siguientes herramientas:

- Para Windows utilizamos las herramientas NetScanTools y PingPlotter.
- Para Linux, las herramientas que empleamos son Mtr y Ping.

Los resultados utilizando la herramienta NetScanTools para Windows y Ping para Linux se presentan en la siguiente tabla:

Destino	Native (%)	Hurricane electric Windows (%)	Hurricane Electric Ubuntu (%)	VPN (%)	Wireguard+VPS (%)
www.google.com	0	1	1,4	0	0

www.facebook.com	0	0,3	0,8	0	0
www.wikipedia.com	0	0,7	1,4	0	0

Tabla 5-5 Paquetes Perdidos Servicios

A partir de la Tabla 5-5, el servicio nativo, la VPN y el servicio Wireguard+VPS no presentan pérdidas, reflejando una alta fiabilidad. En cambio, los túneles de Hurricane Electric muestran una ligera inestabilidad con pérdidas entre el 0-1,5%. Aunque no son pérdidas excesivas, podrían provocar algún error esporádico en varios servicios.

Para completar el estudio de este parámetro, vamos a complementarlo con una tabla donde se muestre la ruta y los distintos nodos que atraviesa cada paquete mostrando las pérdidas que sufre cada nodo intermedio.

Su representación se puede ver en las siguientes tablas:

Servicio	Destino	Saltos	IP	Nombre	Porcentaje de pérdidas (%)
Nativo	www.google.com	1	2a0c:5a84:7807:bc00:c251:5cff:fe97:d14	2a0c:5a84:7807:bc00:c251:5cff:fe97:d14	0
		2	2a0c:5a84:78ff:ff00::2	2a0c:5a84:78ff:ff00::2	0
		3	2a0c:5a84:71ff:ff00::1	2a0c:5a84:71ff:ff00::1	0
		4	2a02:2f0f:163::30	2a02:2f0f:163::30	0
		5	2a0c:5a80:100:5:7269:1:5169:1	2a0c:5a80:100:5:7269:1:5169:1	0
		6	2001:4860:0:1::86d5	2001:4860:0:1::86d5	0
		7	2001:4860:0:1::5317	2001:4860:0:1::5317	0
		8	2a00:1450:4003:80f::2004	www.google.com	0
Nativo	www.facebook.com	1	2a0c:5a84:7807:bc00:c251:5cff:fe97:d14	2a0c:5a84:7807:bc00:c251:5cff:fe97:d14	0
		2	2a0c:5a84:78ff:ff00::2	2a0c:5a84:78ff:ff00::2	0
		3	2a0c:5a84:71ff:ff01::1	2a0c:5a84:71ff:ff01::1	0
		4	2a02:2f00:8700::a	2a02:2f00:8700::a	0
		5	2620:0:1cff:dead:beee:1732	ae3.pr01.mad2.tfbnw.net	0
		6	2620:0:1cff:dead:beef:360c	po207.asw02.mad1.tfbnw.net	0

		7	2620:0:1cff:dead:beef ::3885	po2008.psw03.mad1.tfb nw.net	0
		8	2a03:2880:f004:ffff::73	be3.msw1af.01.mad1.tfb nw.net	0
		9	2a03:2880:f104:83:fa ce:b00c:0:25de	www.facebook.com	0
Nativo	www.wikipedia.com	1	2a0c:5a84:7807:bc00:c251:5cff:fe97:d14	2a0c:5a84:7807:bc00:c251:5cff:fe97:d14	0
		2	2a0c:5a84:78ff:ff00::2	2a0c:5a84:78ff:ff00::2	0
		3	2a0c:5a84:71ff:ff00::1	2a0c:5a84:71ff:ff00::1	0
		4	2a02:2f00:8700::a	2a02:2f00:8700::a	0
		5	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		6	2001:470:0:1f7::2	e0-34.core1.bcn1.he.net	36,6
		7	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		8	2001:7f8:54:5::11	wikimedia.mrs.franceix.net	0
		9	2a02:ec80:600:fe08::2	et-0-0-50.asw1-b12-drmrs.wikimedia.org	0
		10	2a02:ec80:600:ed1a::3	www.wikipedia.com	0

Tabla 5-6 Rutas IPv6 Nativo

Tras analizar la Tabla 5-6, podemos observar que el encaminamiento hacia los distintos destinos se realiza de manera estable, con un número de saltos aceptables y sin pérdidas significativa en la mayoría de los nodos.

Además, podemos concluir que:

- Tanto en Google como en Facebook todos los saltos intermedios presentan pérdidas del 0%, indicando un recorrido óptimo y estable.
- Para Wikipedia, aunque el destino se alcance sin pérdidas, varios saltos intermedios presentan pérdidas del 100%, indicando posibles equipos que descartan paquetes ICMP.

En general, la calidad de las rutas IPv6 hacia estos destinos puede considerarse buena, presentando pérdidas ínfimas.

Servicio	Destino	Saltos	IP	Nombre	Porcentaje de pérdidas
----------	---------	--------	----	--------	------------------------

					(%)
Hurricane Windows	www.google.com	1	2001:470:1f08:3e9::1	tunnel955780.tunnel.tserv5.lon1.ipv6.he.net	1
		2	2001:470:0:67::1	e0-19.core2.lon2.he.net	1
		3	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		4	2001:7f8:4::3b41:1	2001:7f8:4::3b41:1	1
		5	2001:4860:0:1::7761	2001:4860:0:1::7761	0
		6	2001:4860:0:1::54d1	2001:4860:0:1::54d1	9,9
		7	2a00:1450:4009:81f::2004	www.google.com	0
Hurricane Windows	www.facebook.com	1	2001:470:1f08:3e9::1	tunnel955780.tunnel.tserv5.lon1.ipv6.he.net	1
		2	2001:470:0:67::1	e0-19.core2.lon2.he.net	1
		3	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		4	2001:7f8:4::b62:1	ge-0.linx.londen03.uk.bb.gin.ntt.net	1
		5	2001:728:0:5000::1989	2001:728:0:5000::1989	0
		6	2620:0:1cff:dead:beef::5e30	po404.asw02.lhr7.tfbnw.net	7,9
		7	2620:0:1cff:dead:beef::4e09	po291.psw02.lhr8.tfbnw.net	1
		8	2a03:2880:f058:ffff::4d	be2.msw1am.01.lhr8.tfbnw.net	0
		9	2a03:2880:f158:82:face:b00c:0:25de	edge-star-mini6-shv-01-lhr8.facebook.com	0

		10	2620:0:1cff:dead:be ef::57f7	po248.psw04.fra5.tf bnw.net	2
		11	2a03:2880:f083:ffff: :ab	be4.msw1ah.01.fra5 .tfbnw.net	1
		12	2a03:2880:f176:84:f ace:b00c:0:25de	www.facebook.com	0
Hurricane Windows	www.wikipedia.com	1	2001:470:1f08:3e9:: 1	tunnel955780.tunnel .tserv5.lon1.ipv6.he. net	0
		2	2001:470:0:67::1	e0- 19.core2.lon2.he.net	1
		3	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		4	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		5	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		6	2001:7f8:1::a501:49 07:1	ae1-380.cr1- esams.wikimedia.or g	0
		7	2a02:ec80:300:fe04: .2	et-0-0-48.asw1- bw27- esams.wikimedia.or g	33,7
		8	2a02:ec80:300:ed1a: .3	www.wikipedia.com	1

Tabla 5-7 Rutas IPv6 Hurricane Electric Windows

Analizando la tabla 5-7, observamos que los paquetes hacia los destinos siguen unas rutas un tanto más largas que el servicio Nativo, presentando además unas pérdidas superiores.

Podemos concluir lo siguiente:

- **Pérdidas frecuentes en nodos intermedios:** En las rutas hacia los destinos se observan pérdidas en varios nodos intermedios, algunas superando incluso el 10%. Estas pérdidas sugieren la existencia de posible congestión o limitaciones en la infraestructura del túnel o nodos intermedios.
- **Número de saltos:** El uso de túnel induce más saltos que otros servicios, reflejando un camino menos optimizado debido a factores como la encapsulación adicional.
- **Tiempos de espera agotados para la solicitud:** En varios saltos intermedios aparece esta respuesta, pudiendo indicar posibles filtrados ICMPv6.

Servicio	Destino	Saltos	IP	Nombre	Porcentaje de pérdidas (%)
Hurricane Ubuntu	www.google.com	1	2001:470:1f08:3e9 ::1	tunnel955780.tunnel .tserv5.lon1.ipv6.he. net	0,2
		2	2001:470:0:67::1	e0-19.core2.lon2.he.net	0,4
		3	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		4	2001:7f8:4::3b41:1	2001:7f8:4::3b41:1	1,8
		5	2001:4860:0:1::7dd9	2001:4860:0:1::7dd9	15
		6	2001:4860:0:1::73c9	2001:4860:0:1::73c9	14,2
		7	2a00:1450:4009:827::2004	lhr48s49-in-x04.1e100.net	1,4
Hurricane Ubuntu	www.facebook.com	1	2001:470:1f08:3e9 ::1	tunnel955780.tunnel .tserv5.lon1.ipv6.he. net	0,4
		2	2001:470:0:67::1	e0-19.core2.lon2.he.net	0,4
		3	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		4	2001:7f8:4::b62:1	ge-0.linx.londen03.uk.bb.gin.ntt.net	0,4
		5	2001:728:0:5000::1989	2001:728:0:5000::1989	0,4
		6	2620:0:1cff:dead:b eef::5e30	po404.asw02.lhr7.tf bnw.net	0,8
		7	2620:0:1cff:dead:b eef::4e09	po291.psw02.lhr8.tf bnw.net	1
		8	2a03:2880:f058:fff f::4d	be2.msw1am.01.lhr8.tfbnw.net	8,2
		9	2a03:2880:f158:82	edge-star-mini6-shv-	0,8

			:face:b00c:0:25de	01-lhr8.facebook.com	
Hurricane Ubuntu	www.wikipedia.com	1	2001:470:1f08:3e9 ::1	tunnel955780.tunnel .tserv5.lon1.ipv6.he. net	0,6
		2	2001:470:0:67::1	e0-19.core2.lon2.he.net	0,6
		3	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		4	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		5	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		6	2001:7f8:1::a501:4907:1	ae1-380.cr1-esams.wikimedia.org	0,4
		7	2a02:ec80:300:fe04::2	et-0-0-48.asw1-bw27-esams.wikimedia.org	36,9
		8	2a02:ec80:300:ed1a::3	ncredir-lb.esams.wikimedia.org	1,4

Tabla 5-8 Rutas IPv6 Hurricane Electric Windows

A través de la Tabla 5-8 analizada, podemos concluir lo siguiente:

- Pérdidas frecuentes en nodos intermedios:** En las rutas hacia los diferentes destinos, se detectan pérdidas frecuentes en nodos intermedios, algunas superando incluso el 10%. Este comportamiento puede generar congestión o sobrecarga en los nodos.
- Tiempos de respuesta agotados:** Algunas rutas aparecen con el letrero de tiempo de espera agotado y un porcentaje de pérdida del 100%. Esto sugiere un posible filtrado de paquetes ICMPv6 que, aunque no impiden el acceso al destino final, dificultan el diagnóstico completo de la ruta que sigue el paquete trámitedo.
- Camino menos optimizado:** El uso del túnel implica realizar procedimientos como la encapsulación adicional, provocando un aumento de la latencia y el rendimiento general de la red.

Servicio	Destino	Saltos	IP	Nombre	Porcentaje de pérdidas (%)
----------	---------	--------	----	--------	----------------------------

VPN	www.google.com	1	fd00:6968:6564:cc::1	fd00:6968:6564:cc::1	0
		2	2001:bc8:1201:721::1	2001:bc8:1201:721::1	0
		3	2001:bc8:1000:1::36	2001:bc8:1000:1::36	0
		4	2001:bc8:1000:1::118	2001:bc8:1000:1::118	0
		5	2001:bc8:1000:1::132	2001:bc8:1000:1::132	0
		6	2001:bc8:1000:1::de	2001:bc8:1000:1::de	0
		7	2001:bc8:0:2::3	2001:bc8:0:2::3	2
		8	2001:4860:1:1::644	2001:4860:1:1::644	0
		9	2001:4860:0:1::7f87	2001:4860:0:1::7f87	0
		10	2001:4860:0:1::216d	2001:4860:0:1::216d	0
		11	2a00:1450:4007:80d::2004	par10s21-in-x04.1e100.net	0
VPN	www.facebook.com	1	fd00:6968:6564:cc::1	fd00:6968:6564:cc::1	0
		2	2001:bc8:1201:721::1	2001:bc8:1201:721::1	0
		3	2001:bc8:1000:1::3a	2001:bc8:1000:1::3a	0
		4	2001:bc8:1000:1::11c	2001:bc8:1000:1::11c	0
		5	2001:bc8:1000:1::12a	2001:bc8:1000:1::12a	0
		6	2001:bc8:1000:1::d4	2001:bc8:1000:1::d4	20,8
		7	2001:bc8:0:2::27	2001:bc8:0:2::27	39,6
		8	2620:0:1cff:dead:beee::ab2	ae0.pr06.cdg5.tfbnw.net	0
		9	2620:0:1cff:dead:beef::64c6	po202.asw04.cdg4.tfbnw.net	0
		10	2620:0:1cff:dead:beef::224d	po1008.psw03.cdg4.tfbnw.net	0

		11	2a03:2880:f08e:ffff: :341	be7.msw1ae.02.cdg 4.tfbnw.net	0
		12	2a03:2880:f17b:187 :fase:b00c:0:25de	www.facebook.com	0
VPN	www.wikipedia.com	1	fd00:6968:6564:cc:: 1	fd00:6968:6564:cc:: 1	0
		2	2001:bc8:1201:721:: 1	2001:bc8:1201:721:: 1	0
		3	2001:bc8:1000:1::3a	2001:bc8:1000:1::3a	0
		4	2001:bc8:1000:1::44	2001:bc8:1000:1::44	0
		5	2001:bc8:1000:1::12 c	2001:bc8:1000:1::12 c	0
		6	2001:bc8:1000:1::d4	2001:bc8:1000:1::d4	0
		7	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		8	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		9	2001:7f8:36::3a3b:0 :1	2001:7f8:36::3a3b:0 :1	0
		10	2a02:ec80:600:fe06: :2	et-0-0-48.asw1-b12- drmrss.wikimedia.or g	0
		11	2a02:ec80:600:ed1a: :3	ncredir- lb.drmrss.wikimedia. org	0

Tabla 5-9 Rutas IPv6 VPN Hide.me

A través de la tabla 5-9, podemos llegar a la siguiente conclusión:

- **Varias pérdidas en nodos intermedios:** Se presentan pérdidas en algunos nodos intermedios, algunas siendo incluso del 20-30%, y aunque no impiden llegar al destino, podrían generar una posible degradación.
- **Tiempos de espera agotados:** En la ruta hacia el destino Wikipedia, observamos varios saltos con el mensaje de tiempo de espera agotados para esta solicitud. Esta alerta sugiere un posible filtrado del tráfico ICMPv6, sin afectar al encaminamiento real.
- **Mayor número de saltos:** El uso de la VPN implica un mayor número de saltos que los servicios anteriormente analizados, rondando entre los 10-12 saltos para cada destino.

Servicio	Destino	Saltos	IP	Nombre	Porcentaje de pérdidas (%)
Wireguard+VPS	www.google.com	1	2a01:4f8:1c0c:7b 00::1	2a01:4f8:1c0c:7b 00::1	0
		2	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		3	2a01:4f8:0:e0c0:: 5852	18869.your- cloud.host	0
		4	2a01:4f8:0:e0c0:: 5801	2a01:4f8:0:e0c0:: 5801	0
		5	2a01:4f8:0:e0c0:: a1dd	spine6.cloud1.nb g1.hetzner.com	0
		6	2a01:4f8:0:e0c0:: a1b5	spine16.cloud1.n bg1.hetzner.com	0
		7	2a01:4f8:0:e0c0:: a1e1	core11.nbg1.hetz ner.com	0
		8	2a01:4f8:0:3::32 6	core5.fra.hetzner. com	0
		9	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		10	2a00:1450:8461: :1	2a00:1450:8461: :1	33,7
		11	2001:4860:0:1::9 e	2001:4860:0:1::9 e	0
		12	2001:4860:0:1::8 886	2001:4860:0:1::8 886	0

Wireguard+VPS	www.facebook.com	13	2001:4860::c:40 03:3648	2001:4860::c:40 03:3648	0
		14	2001:4860::9:40 01:31f2	2001:4860::9:40 01:31f2	91,3
		15	2001:4860:0:1::8 69b	2001:4860:0:1::8 69b	0
		16	2001:4860:0:1::3 167	2001:4860:0:1::3 167	0
		17	2a00:1450:4001: 810::2004	fra16s50-in- x04.1e100.net	0
		1	2a01:4f8:1c0c:7b 00::1	2a01:4f8:1c0c:7b 00::1	0
		2	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		3	2a01:4f8:0:e0c0:: 5852	18869.your- cloud.host	0
		4	2a01:4f8:0:e0c0:: 5801	2a01:4f8:0:e0c0:: 5801	0
		5	2a01:4f8:0:e0c0:: a1dd	spine6.cloud1.nb g1.hetzner.com	0
		6	2a01:4f8:0:e0c0:: a289	spine15.cloud1.n bg1.hetzner.com	0
		7	2a01:4f8:0:3::2a 1	core11.nbg1.hetz ner.com	0
		8	2a01:4f8:0:3::1d	core4.fra.hetzner. com	0

		9	2620:0:1cff:dead :beee::19f0	ae1.pr02.fra2.tfb nw.net	0
		10	2620:0:1cff:dead :beef::7b0	po181.asw01.fra 5.tfbnw.net	0
		11	2620:0:1cff:dead :beef::ba7	po215.psw01.fra 3.tfbnw.net	0
		12	2a03:2880:f084:f fff::22b	be1.msw1av.02.f ra3.tfbnw.net	0
		13	2a03:2880:f177: 185:face:b00c:0: 25de	edge-star-mini6- shv-02- fra3.facebook.co m	0
Wireguard+VPS	www.wikipedia.com	1	2a01:4f8:1c0c:7b 00::1	2a01:4f8:1c0c:7b 00::1	1
		2	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	100
		3	2a01:4f8:0:e0c0:: 5852	18869.your- cloud.host	2
		4	2a01:4f8:0:e0c0:: 5801	2a01:4f8:0:e0c0:: 5801	0
		5	2a01:4f8:0:e0c0:: a1d9	spine5.cloud1.nb g1.hetzner.com	0
		6	2a01:4f8:0:e0c0:: a281	spine15.cloud1.n bg1.hetzner.com	0
		7	2a01:4f8:0:3::2a 1	core11.nbg1.hetz ner.com	1

	8	2a01:4f8:0:3::32 6	core5.fra.hetzner. com	0
	9	2001:7f8:13::a50 1:4907:1	2001:7f8:13::a50 1:4907:1	0
	10	Tiempo de espera agotado para esta solicitud.	Tiempo de espera agotado para esta solicitud.	1,7
	11	2a02:ec80:300:e d1a::3	ncredir- lb.esams.wikime dia.org	1

Tabla 5-10 Rutas IPv6 VPN Wireguard

Analizando la tabla 5-10, podemos llegar a la siguiente conclusión:

- **Pérdidas en nodos intermedios:** En algunos nodos intermedios ocurren pérdidas que pueden ir desde pérdidas insignificantes con porcentajes que ronda el 1% hasta pérdidas bastante elevadas sobre pasando en varios casos el 30%.
- **Tiempos de espera agotados para esta solicitud:** En varios nodos intermedios aparece este mensaje con unas pérdidas del 100%, sugiriendo que estos nodos filtran las respuestas ICMPv6.
- **Mayor número de saltos:** Para este servicio, los paquetes deben de realizar mayor número de saltos para llegar al destino que los servicios analizados anteriormente. Este número de saltos varía entre 13 y 17 dependiendo del dominio. A pesar de este incremento de saltos, no implica necesariamente un mayor retardo si la red está bien gestionada, que es el caso de este servicio.

5.2.3 Tiempo de resolución DNS

Para medir el tiempo de resolución DNS utilizamos las siguientes herramientas:

- Para Windows, utilizamos la combinación de las herramientas Nslookup y Wireshark para capturar los paquetes tramitados y medir con precisión le tiempo entre consulta y respuesta.
- Para Linux, utilizamos la herramienta dig, que permite obtener información detallada de la resolución DNS.

Los resultados obtenidos se muestran en la siguiente gráfica:

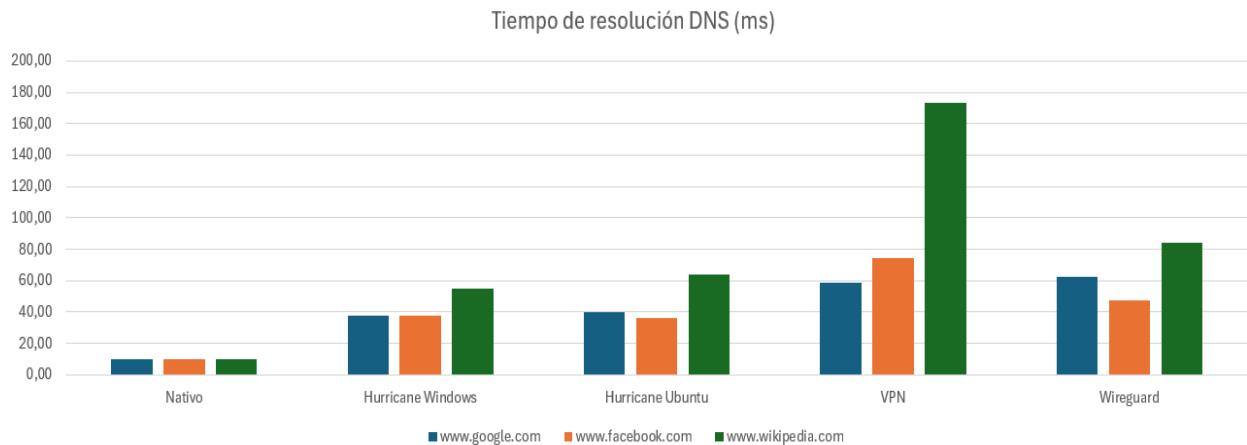


Figura 5-38 Tiempos de resolución DNS Servicios

A partir de la Figura 5-38, observamos como el servicio Nativo es el que ofrece los tiempos de resolución más bajos. Los demás servicios ofrecen unos tiempos de resolución DNS bastante similares, aunque podemos apreciar una ligera ventaja en los servicios de Hurricane Electric.

5.2.4 Prueba en descargas de gran tamaño

Para esta prueba, hemos evaluado el rendimiento de los servicios en descargas de archivos con un gran volumen. Para ello, con cada servicio hemos realizado una descarga de una ISO de Ubuntu, la cual tiene un tamaño aproximado de 5Gb.

Este tipo de prueba permite medir el rendimiento de cada servicio en una descarga de gran tamaño, además de comprobar su estabilidad y la capacidad para mantener las tasas constantes durante toda la descarga.

Para medir el parámetro, utilizamos la herramienta curl con el parámetro -6 para especificar que nuestra intención es utilizar IPv6.

Los resultados, expresados en el tiempo total de descarga, se muestran en la siguiente gráfica:

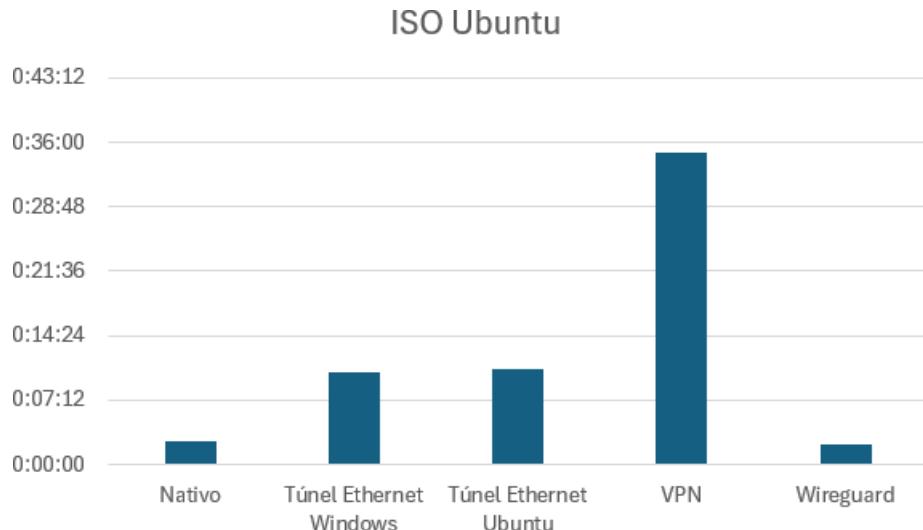


Figura 5-39 Tiempos Descargas Gran Volumen Servicios

A partir de la Figura 5-39, podemos extraer las siguientes conclusiones:

- El servicio Wireguard+VPS ofrece una excelente velocidad y un rendimiento sobresaliente ante descargas de este tipo. Además, mantiene velocidades altas y constantes durante toda la descarga, posicionándolo como una solución excelente para este tipo de uso.
- Los servicios de Hurricane Electric ofrecen un rendimiento aceptable. Aunque no alcanzan los niveles de velocidad de Wireguard, ofrecen una velocidad que permite obtener una buena experiencia a la mayoría de los usuarios.
- El servicio VPN, por su parte, presenta claras limitaciones en descargas de gran peso, afectando negativamente en la experiencia del usuario que lo implementa.
- El servicio nativo IPv6 ofrece unas velocidades muy altas, manteniéndose como la opción más cómoda y óptima para usar.

5.2.5 Ancho de banda

Para calcular el ancho de banda hemos utilizado la web SpeedTest, ampliamente reconocida por ofrecer resultados con bastante precisión del ancho de banda tanto en descarga como en subida.

Los resultados, expresados en Mbps, se muestran en la siguiente tabla:

Método IPv6	Mbps
Nativo	325
Túnel Ethernet Windows	99,7
Túnel Ethernet Linux	88,3
VPN Windows	84,9
Wireguard	517

Tabla 5-11 Ancho de banda Servicios

Los resultados reflejados en la Tabla 5-11, nos indica que:

- Los servicios de Hurricane Electric y VPN ofrecen un ancho de banda en torno a los 100 Mbps en cualquier red aplicada, imponiendo limitaciones en el rendimiento. Estas limitaciones pueden venir de factores como la encapsulación adicional que realizan.
- Para el servicio Wireguard+VPS, ofrece un excelente ancho de banda, alcanzando un gran rendimiento. Este ancho de banda viene dado por la eficiencia del protocolo Wireguard y por la capacidad del VPS contratado, que permite aprovechar al máximo el potencial de la red local del usuario.
- Para el servicio Nativo, el ancho de banda depende de la tarifa contratada por el usuario con su proveedor de servicios.

5.2.6 Jitter

Para calcular el Jitter, al igual que el ancho de banda, utilizamos la web speedtest, que permite obtener una medición con bastante precisión de este parámetro de red.

Los resultados, expresados en milisegundos, se muestran en la siguiente tabla:

Método IPv6	Tiempo (ms)
Nativo	1,13

Túnel Ethernet Windows	3,5
Túnel Ethernet Linux	2,58
VPN Windows	2,92
Wireguard	2,34

Tabla 5-12 Jitter Servicios

A través de la Tabla 5-12, tenemos en cuenta que:

- El servicio Nativo es el que cuenta con el jitter más bajo, ofreciendo el mejor rendimiento para aplicaciones en tiempo real.
- El servicio de Hurricane Electric implementado en Windows ofrece un jitter ligeramente superior, pudiendo afectar de manera ligera en las aplicaciones en tiempo real, aunque sigue siendo un tiempo bastante bajo y que se mantiene dentro de los márgenes aceptables.
- Para los demás servicios, ofrecen un jitter bastante similar, con un tiempo bastante bajo, haciéndolo ideal como alternativa al servicio Nativo.

5.2.7 Número de saltos

Tras haber estudiado la ruta por las que pasa cada paquete de cada servicio, realizamos un análisis del número de saltos que realizan los paquetes para llegar hasta el destino. Para ello, utilizamos la herramienta PingPlotter en Windows y Mtr en Linux teniendo como destino a Google.com, Facebook.com y Wikipedia.com.

En la siguiente tabla representamos el número de saltos totales que debe realizar el paquete para llegar al destino correspondiente:

Servicio	Saltos Google	Saltos Facebook	Saltos Wikipedia
Nativo	8	9	10
Túnel Ethernet Windows	7	12	8
Túnel Ethernet Linux	7	9	8
VPN Windows	11	12	11
Wireguard	17	13	11

Tabla 5-13 Número Saltos Servicios

La tabla 5-13, el servicio Nativo presenta generalmente un número menor de saltos, lo cual es algo habitual ya que no necesita de túnel y su ruta es más directa para llegar al destino. Por otro lado, los servicios de túnel introducen varios saltos adicionales, los cuales se pueden deber a factores como el encapsulamiento.

5.2.8 MTU

Para la MTU utilizamos la herramienta **netsh** a través del comando **netsh interface ipv6 show subinterfaces** en Windows. Para Linux utilizamos el comando **ifconfig**. A través de estas herramientas podemos obtener la MTU que utiliza cada interfaz configurada

Método IPv6	Tamaño
Nativo	1500

Túnel Ethernet Windows	1280
Túnel Ethernet Linux	1480
VPN Windows	1280
Wireguard	1420

Tabla 5-14 MTU Servicios

Como podemos apreciar en la Tabla 5-14, el valor de 1500 bytes de MTU para el servicio nativo es esperado, ya que en redes Ethernet se suele usar este tamaño de MTU. Por el contrario, servicios que implican túneles o cifrado como las VPN reducen su MTU para evitar la fragmentación al agregar encabezados adicionales. La fragmentación puede afectar negativamente a la red, provocando latencias adicionales y en general una reducción en el rendimiento. Observamos también varios servicios que usan 1280 de MTU. Este tamaño es el mínimo utilizado para el protocolo IPv6 [31].

6 CONCLUSIONES Y MEJORAS FUTURAS

Tras realizar todas las mediciones de cada parámetro de red para cada uno de los servicios implementados y su posterior comparación, podemos elaborar una conclusión para cada servicio. Para ello implementamos una tabla comparativa donde podremos observar en que destaca cada servicio implementado.

Parámetro	Conectividad Nativa	TunnelBroker (Hurricane Electric)	VPN con soporte IPv6	Wireguard+VPS con soporte IPv6
Velocidad	Muy alta	Media	Baja	Alta
Rendimiento	Muy alto	Alto	Medio	Muy alto
Coste	Tarifa	Gratis	Plan de prueba Gratis/Pago mensual	Pago mensual
Compatibilidad	Alta	Alta	Alta	Alta
Flexibilidad	Media	Alta	Baja	Alta
Estabilidad	Muy alta	Media	Alta	Alta
Configuración	Automática	Media	media	Alta
Soporte Multiplataforma	Muy alto	Alto	Alto	Alto

Tabla 6-1 Valoración Servicios

En conclusión, observando la Tabla 6-1, lo más recomendable siempre será utilizar el plan de direccionamiento IPv6 proporcionado por la operadora de red. Sin embargo, si necesitamos IPv6 para alguna tarea en concreto, utilizar TunnelBroker de Hurricane Electric es una gran solución, ya que ofrece un alto rendimiento y su uso es gratuito.

Si no existiera la posibilidad de poder acceder a un plan de direccionamiento IPv6 por parte de la operadora y su uso es urgente, una excelente solución es utilizar Wireguard+VPS. Permite unas velocidades muy altas y pérdidas prácticamente nulas, convirtiéndolo en una solución profesional con un excelente rendimiento. Su mayor inconveniente es el costo ya que no es un servicio gratis y es necesario realizar un pago mensual para mantener la VPS, pero es un costo que se puede afrontar ya que no es para nada desproporcionado.

El direccionamiento IPv6 es algo que está en auge y que en un futuro no tan próximo se utilizará de manera mundial. Por tanto, tener estas soluciones a mano nos permite estudiar a fondo este protocolo y hacer uso de un direccionamiento IPv6 en diferentes casos donde su uso sea necesario.

Finalmente, sobre este proyecto, queremos comentar que es un proyecto que puede ser retomado en un futuro con varias líneas de mejora.

- **Ampliación de servicio:** Existe la posibilidad de implementar más servicios que proporcionen direccionamiento IPv6 a una red. Un ejemplo podría ser implementar una VPN de pago y ver sus diferencias con VPNs gratuitas en términos de velocidad o utilizar una VPS y configurar el túnel VPN a través de OpenVPN en lugar de Wireguard y analizar sus diferencias.
- **Comportamiento en diferente tramo horario:** Podríamos realizar un estudio más exhaustivo, con la

posibilidad de realizar mediciones en diferentes tramos horarios y comparar en qué momento el servicio nos proporciona un mayor rendimiento.

- **Uso de herramientas alternativas:** Para la toma de datos podríamos utilizar otras herramientas diferentes a las utilizadas en el proyecto y comparar la potencia de cada una de ellas
- **Evaluación en diferentes sistemas operativos:** Implementar los servicios en otros sistemas operativos que no sean Windows o Linux y comprobar cómo sería configurarlo en un entorno diferente a lo habitual podría ser una idea bastante interesante.

En conclusión, se trata de un proyecto abierto a múltiples extensiones con la posibilidad de poder ir modificándolo constantemente, convirtiéndolo en un proyecto muy interesante para pruebas o desarrollos más avanzados.

ANEXO A. EQUIPOS UTILIZADOS

En este apartado vamos a mostrar los equipos donde se han llevado a cabo la instalación y configuración de los distintos servicios implementados y la posterior medición de los distintos parámetros de red.

A.1 MSI GF63 Thin 9SC

Este equipo se utilizó como entorno de pruebas principal para los servicios que requerían del sistema operativo Windows. Además, es el alojador de las distintas herramientas empleadas para el cálculo de los diferentes parámetros de red. Las características más relevantes de este equipo son las siguientes:

- Hardware
 - Intel® Core™ i7-9750H @ 2.60 GHz (6 núcleos, 12 hilos)
 - Memoria RAM de 16,0 GB (15,8 GB usable)
 - Arquitectura de 64 bits
 - Gráfica GTX 1650 Max-Q de Nvidia
 - Almacenamiento en 500 GB de SSD tipo PCIe 3.0 ×4
 - Conectividad a través de wifi 802.11 ac, Bluetooth 5.0, Ethernet
- Sistema operativo
 - Windows 11 Home Single Language, versión 23H2 (build 22631.5549)

A.2 Laptop HP

Este equipo fue utilizado principalmente para implementar los servicios que requerían del uso del sistema operativo Linux. Además, fue utilizado para el desarrollo del script utilizado para el cálculo del servidor DNS óptimo. Las características de este equipo son las siguientes:

- Hardware
 - Memoria RAM de 4 GB
 - Procesador Intel® Celeron® N2840 @ 2.16 GHz (2 núcleos)
 - Gráfica integrada Intel HD Graphics (Bay Trail)
 - Almacenamiento HDD de 500 GB
- Sistema operativo
 - Ubuntu 20.04.6 LTS
 - Sistema operativo de 64 bits
 - GNOME 3.36.8
 - Sistema de ventanas X11

A.3 Proveedor de red contratado y tipo de conexión empleado

Para el desarrollo de este proyecto se ha hecho de dos redes domésticas distintas, proporcionadas por diferentes proveedores. En una de ellas el proveedor de red proporcionaba direccionamiento IPv6 de manera nativa y en la otra no. Esta situación nos permitió poder configurar estos servicios y comprobar las diferencias entre unos y otros.

- Red sin soporte nativo de IPv6

En esta red como hemos comentado anteriormente, hemos implementado la mayoría de los servicios que proporcionan conectividad IPv6 mediante túneles. Esta conexión corresponde a una tarifa contratada a FiberPlus, el cual es una empresa local de telecomunicaciones especializada en fibra óptica y telefonía móvil. La velocidad contratada es de 600 Mb simétricos y su tipo de conectividad se realiza a través de una dirección IPv4 pública, ya que la operadora de red no nos proporciona direccionamiento IPv6 global.

- Red con soporte nativo IPv6

En esta red hemos hecho las pruebas de IPv6 nativo, el cual no tiene la necesidad de utilizar servicios de túnel. La velocidad contratada en esta red es de 500 Mb simétricos y el tipo de conectividad es a través de una dirección IPv4 pública y una serie de prefijos de red IPv6.

Para las pruebas realizadas en ambas redes, hemos utilizado una conexión cableada a través de cables Ethernet de categoría 6. Este tipo de cable permite una velocidad de hasta 1 Gbps por segundo, haciendo que los servicios implementados obtengan el mayor rendimiento posible, ya que una conexión por Wi-Fi podría provocar un mayor número de pérdidas por diversas condiciones como las interferencias.

ANEXO B. VPS CONTRATADA

Para el desarrollo de este proyecto optamos por utilizar una VPS a través del proveedor Hetzner, una empresa de origen alemán, que destaca por su fiabilidad y una gran calidad en sus servicios. A través de la página oficial de Hetzner, podemos registrarnos y obtener una VPS realizando el correspondiente pago, el cual es mensual.

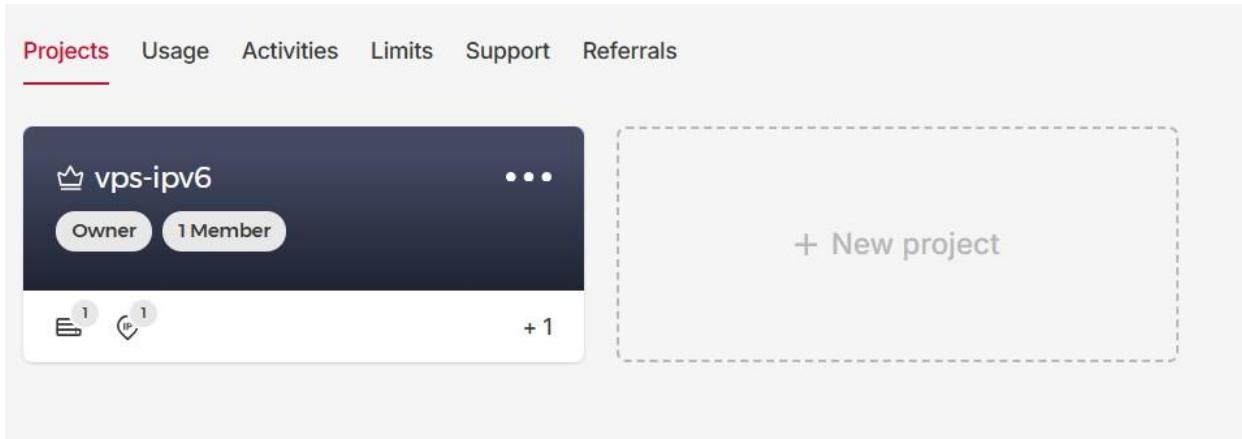


Figura B-1 Servidor VPS

En nuestro proyecto, el tipo de instancia seleccionada para la VPS fue el modelo llamado cx22, el cual dispone de los siguientes recursos:

- 4 GB de memoria RAM
- 40 GB de almacenamiento Local
- Hasta 20 Tb de tráfico mensual incluido
- 2 núcleos de CPU virtuales

En cuanto al precio, no es una VPS generalmente cara. En nuestro caso, el pago a mensual a realizar es de 3,98€, lo que lo convierte en una opción más que rentable para desarrollo de pruebas o proyectos como el que estamos realizando.

En cuanto a la conectividad que nos ofrece, este trabaja tanto con IPv4 como con IPv6, algo que es fundamental para nuestro proyecto. La dirección IPv4 que trae asignada es la **195.201.217.41** y su prefijo IPv6 global la **2a01:4f8:1c0c:7b00::/64**

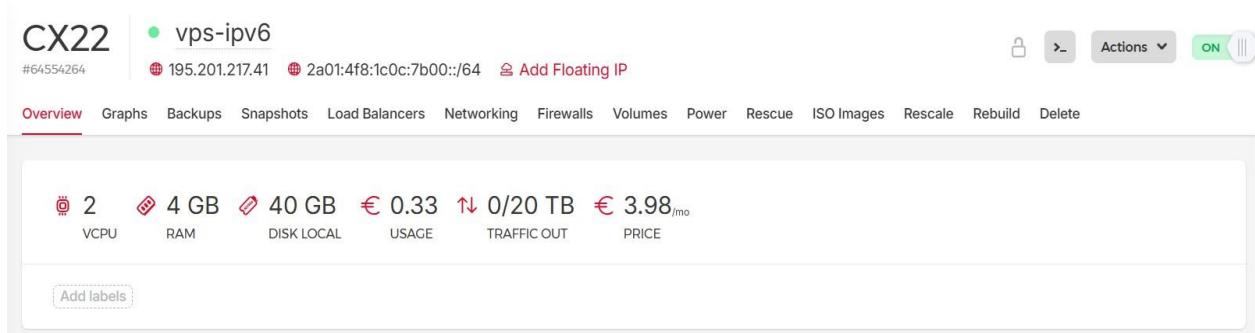


Figura B-2 Configuraciones VPS

Además, otro aspecto relevante es su ubicación geográfica, ubicada físicamente en Nuremberg (Alemania). Esta ubicación ha sido relevante para evaluar la latencia real desde España hasta un nodo remoto en el centro de Europa.



Figura B-3 Ubicación VPS

Una de las ventajas que nos ofrece la VPS es la posibilidad del usuario de poder personalizar varios parámetros en el momento de la contratación y su creación. Una de las opciones es la elección de sistema operativo, el cual hemos optado por utilizar Linux, utilizando la distribución Debian.

Para poder acceder a la VPS de manera remota, utilizamos el protocolo SSH (Secure Shell), que permite el acceso seguro a través de la red, ya que sus conexiones están cifradas.

ANEXO C. SCRIPT PARA DNS ÓPTIMO

Para el procedimiento de obtención del servidor DNS con mejor rendimiento para el servicio implementado en Linux, desarrollamos un script que automatice todo el proceso. El script consta del siguiente código:

```
#!/bin/bash

#Colours
greenColour="\e[0;32m\033[1m"
endColour="\033[0m\e[0m"
redColour="\e[0;31m\033[1m"
blueColour="\e[0;34m\033[1m"
yellowColour="\e[0;33m\033[1m"
purpleColour="\e[0;35m\033[1m"
turquoiseColour="\e[0;36m\033[1m"
grayColour="\e[0;37m\033[1m"

function ctrl_c(){
    echo -e "\n$redColour[+] Saliendo del script...$endColour\n"
    exit 0
}

trap ctrl_c INT

function helpPannel(){
    echo -e "\n$yellowColour Panel de ayuda$endColour"
    echo -e "\n$yellowColour d)$endColour$turquoiseColour Direccion IPv6 del servidor DNS$endColour"
    echo -e "\n$yellowColour u)$endColour$turquoiseColour Dominios a analizar$endColour\n"
}

function analizarDNS(){

DNS=$1
echo -e "\n$yellowColour[+]\$endColour$blueColour No se detectó caché DNS local. No es necesario vaciarla.\$endColour\n"

echo -e "$yellowColour[+]\$endColour$blueColour Estableciendo servidor DNS temporalmente en \$greenColour\$DNS\$endColour$blueColour...\$endColour\n"
```

```

if [ -L /etc/resolv.conf ]; then
    echo "$redColour[i]$endColour$greenColour /etc/resolv.conf es un enlace simbólico. Se reemplazará
temporalmente.$endColour"
    sudo rm /etc/resolv.conf
fi
#sudo tee /etc/resolv.conf> /dev/null
sudo bash -c "echo 'nameserver $DNS' > /etc/resolv.conf"

echo -e "$yellowColour[+]$endColour$blueColour Comprobando el contenido actual de
$greenColour/etc/resolv.conf$endColour$blueColour...$endColour\n"

#Latencia media=$(ping6 www.google.com -c 1 | tail -n 1 | awk -F '=' '{print $2}' | awk -F '/' '{print $2}')
#Latencia maxima=$(ping6 www.google.com -c 1 | tail -n 1 | awk -F '=' '{print $2}' | awk -F '/' '{print $3}')
#Latencia minima=$(ping6 www.google.com -c 1 | tail -n 1 | awk -F '=' '{print $2}' | awk -F '/' '{print $1}')

echo -e "$yellowColour[+]$endColour$blueColour Medicion de la Latencia a servidor DNS con
IP$endColour $greenColour$DNS$endColour$blueColour... $endColour\n"
Result=$(ping6 $DNS -c 100 | tail -n 5)
Latencia_minima=$(echo "$Result" | tail -n 1 | awk -F '=' '{print $2}' | awk -F '/' '{print $1}')
Latencia_maxima=$(echo "$Result" | tail -n 1 | awk -F '=' '{print $2}' | awk -F '/' '{print $3}')
Latencia_media=$(echo "$Result" | tail -n 1 | awk -F '=' '{print $2}' | awk -F '/' '{print $2}')
echo -e "$purpleColour La latencia minima es $endColour$greenColour$Latencia_minima$endColour\n"
echo -e "$purpleColour La latencia maxima es $endColour$greenColour$Latencia_maxima$endColour\n"
echo -e "$purpleColour La latencia media es $endColour$greenColour$Latencia_media$endColour\n"

Perdida=$(echo "$Result" | tail -n 2 | awk -F 'received,' '{print $2}' | awk -F ',' '{print $1}')
echo -e "$purpleColour El porcentaje de perdida de paquetes
es:$endColour$greenColour$Perdida$endColour\n"
}

function analizarDominio1(){

Dominio1=$1
echo -e "$yellowColour[+]$endColour$blueColour Tiempo de resolucion DNS del Dominio
$endColour$greenColour$Dominio1$endColour\n"
Resolucion=$(dig -6 $Dominio1 AAAA)
#echo "$Resolucion"
IP=$(echo "$Resolucion" | tail -n 6 | head -n 1 | awk '{print $NF}')
echo -e "$purpleColour La IP para el dominio$endColour $greenColour$Dominio1$endColour$blueColour
es: $endColour $greenColour$IP$endColour\n"

Tiempo=$(echo "$Resolucion" | tail -n 4 | head -n 1 | awk -F ':' '{print $2}') #echo -e "$Tiempo"

```

```
echo -e "$purpleColour El tiempo de resolucion DNS para este dominio  
es:$endColour$greenColour$Tiempo$endColour\n"
```

```
echo -e "$yellowColour[+]$endColour$blueColour Calculo de la latencia de la IP del dominio  
introducido...\n$endColour"
```

```
Resultado=$(ping6 $IP -c 100 | tail -n 5)
```

```
Latencia_minima2=$(echo "$Resultado" | tail -n 1 | awk -F '=' '{print $2}' | awk -F '/' '{print $1}')
```

```
Latencia_maxima2=$(echo "$Resultado" | tail -n 1 | awk -F '=' '{print $2}' | awk -F '/' '{print $3}')
```

```
Latencia_media2=$(echo "$Resultado" | tail -n 1 | awk -F '=' '{print $2}' | awk -F '/' '{print $2}')
```

```
echo -e "$purpleColour La latencia minima es $endColour$greenColour$Latencia_minima2$endColour\n"
```

```
echo -e "$purpleColour La latencia maxima es $endColour$greenColour$Latencia_maxima2$endColour\n"
```

```
echo -e "$purpleColour La latencia media es $endColour$greenColour$Latencia_media2$endColour\n"
```

```
Perdida2=$(echo "$Resultado" | tail -n 2 | awk -F 'received,' '{print $2}' | awk -F ',' '{print $1}')
```

```
echo -e "$purpleColour El porcentaje de perdida de paquetes  
es:$endColour$greenColour$Perdida2$endColour\n"
```

```
}
```

```
function analizarDominio2(){
```

```
echo -e "$yellowColour[+]$endColour$blueColour Tiempo de resolucion DNS del Dominio  
$endColour$greenColour$Dominio2$endColour\n"
```

```
Resolucion2=$(dig -6 $Dominio2 AAAA)
```

```
#echo "$Resolucion2"
```

```
IP2=$(echo "$Resolucion2" | tail -n 6 | head -n 1 | awk '{print $NF}')
```

```
echo -e "$purpleColour La IP para el dominio$endColour $greenColour$Dominio2$endColour$blueColour  
es: $endColour$greenColour$IP2$endColour\n"
```

```
Tiempo2=$(echo "$Resolucion2" | tail -n 4 | head -n 1 | awk -F ':' '{print $2}')
```

```
#echo -e "$Tiempo2"
```

```
echo -e "$purpleColour El tiempo de resolucion DNS para este dominio  
es:$endColour$greenColour$Tiempo2$endColour\n"
```

```
echo -e "$yellowColour[+]$endColour$blueColour Calculo de la latencia de la IP del dominio  
introducido...\n$endColour"
```

```
Resultado2=$(ping6 $IP2 -c 100 | tail -n 5)
```

```
Latencia_minima3=$(echo "$Resultado2" | tail -n 1 | awk -F '=' '{print $2}' | awk -F '/' '{print $1}')
```

```
Latencia_maxima3=$(echo "$Resultado2" | tail -n 1 | awk -F '=' '{print $2}' | awk -F '/' '{print $3}')
```

```
Latencia_media3=$(echo "$Resultado2" | tail -n 1 | awk -F '=' '{print $2}' | awk -F '/' '{print $2}')
```

```
echo -e "$purpleColour La latencia minima es $endColour$greenColour$Latencia_minima3$endColour\n"
```

```
echo -e "$purpleColour La latencia maxima es $endColour$greenColour$Latencia_maxima3$endColour\n"
```

```
echo -e "$purpleColour La latencia media es $endColour$greenColour$Latencia_media3$endColour\n"
```

```

Perdida3=$(echo "$Resultado2" | tail -n 2 | awk -F 'received,''{print $2}' | awk -F ',''{print $1}')

echo -e "$purpleColour El porcentaje de perdida de paquetes
es:$endColour$greenColour$Perdida3$endColour\n"
}

function analizarDominio3(){
echo -e "$yellowColour[+]$endColour$blueColour Tiempo de resolucion DNS del Dominio
$endColour$greenColour$Dominio3$endColour\n"

Resolucion3=$(dig -6 $Dominio3 AAAA)
#echo "$Resolucion3"

IP3=$(echo "$Resolucion3" | tail -n 6 | head -n 1 | awk '{print $NF}')
echo -e "$purpleColour La IP para el dominio$endColour $greenColour$Dominio3$endColour$blueColour
es: $endColour$greenColour$IP3$endColour\n"

Tiempo3=$(echo "$Resolucion3" | tail -n 4 | head -n 1 | awk -F ':' '{print $2}')
#echo -e "$Tiempo3"

echo -e "$purpleColour El tiempo de resolucion DNS para este dominio
es:$endColour$greenColour$Tiempo3$endColour\n"

echo -e "$yellowColour[+]$endColour$blueColour Calculo de la latencia de la IP del dominio
introducido...\n$endColour"

Resultado3=$(ping6 $IP3 -c 100 | tail -n 5)

Latencia_minima4=$(echo "$Resultado3" | tail -n 1 | awk -F '=' '{print $2}' | awk -F '/' '{print $1}')
Latencia_maxima4=$(echo "$Resultado3" | tail -n 1 | awk -F '=' '{print $2}' | awk -F '/' '{print $3}')
Latencia_media4=$(echo "$Resultado3" | tail -n 1 | awk -F '=' '{print $2}' | awk -F '/' '{print $2}')

echo -e "$purpleColour La latencia minima es $endColour$greenColour$Latencia_minima4$endColour\n"
echo -e "$purpleColour La latencia maxima es $endColour$greenColour$Latencia_maxima4$endColour\n"
echo -e "$purpleColour La latencia media es $endColour$greenColour$Latencia_media4$endColour\n"

Perdida4=$(echo "$Resultado3" | tail -n 2 | awk -F 'received,''{print $2}' | awk -F ',''{print $1}')

echo -e "$purpleColour El porcentaje de perdida de paquetes
es:$endColour$greenColour$Perdida4$endColour\n"
}

declare -i parameter_counter=0

while getopts "d:u:h" arg; do
  case $arg in

```

```
d) DNS=$OPTARG; let parameter_counter+=1;;
u)
Dominio1=$O
PTARG shift
$((OPTIND -
1))
Dominio2=$1
Dominio3=$2
let parameter_counter+=2;;
h) helpPannel
exit 0;; esac
done
```

```
if[ $parameter_counter -eq 1 ];
then analizarDNS $DNS
elif[ $parameter_counter -eq 3 ];
then analizarDNS $DNS
analizarDominio1
$Dominio1

if[[ -n "$Dominio2" ]]; then
analizarDominio2
fi

if[[ -n "$Dominio3" ]]; then
analizarDominio3
fi
else
helpPannel
fi
```

REFERENCIAS

- [1] M. Ford, «An Eye On The Numbers: IPv6 Deployment,» Internet Society Pulse, 9-jun-2022. [En línea]. Available: <https://pulse.internetsociety.org/blog/an-eye-on-the-numbers-ipv6-deployment>.
- [2] S. Deering, «Internet Protocol, Version 6 (IPv6) Specification,» RFC 8200, [En línea]. Available: <https://tools.ietf.org/html/rfc8200>.
- [3] R. Hinden, «rfc 4291 - IPv6 Adressing Architecture,» [En línea]. Available: <https://tools.ietf.org/html/rfc4291>.
- [4] J. M. V. Torres, Introducción a IPv6.
- [5] A. Conta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," RFC 4443, March 2006. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc4443>
- [6] T. Narten, «Neighbor Discovery for IP version 6 (IPv6),» [En línea]. Available: <https://tools.ietf.org/html/rfc4861>.
- [7] S. Thomson, «IPv6 Stateless Address Autoconfiguration,» [En línea]. Available: <https://tools.ietf.org/html/rfc4862>.
- [8] J. McCann, «Path MTU Discovery for IP version 6,» [En línea]. Available: <https://tools.ietf.org/html/rfc8201>.
- [9] S. Deering, «Multicast Listener Discovery (MLD) for IPv6,» [En línea]. Available: <https://tools.ietf.org/html/rfc2710>.
- [10] S. Kent, «Security Architecture for the Internet Protocol,» [En línea]. Available: <https://tools.ietf.org/html/rfc4301>.
- [11] S. Kent, «IP Authentication Header,» [En línea]. Available: <https://tools.ietf.org/html/rfc4302>.
- [12] S. Kent, «IP Encapsulating Security Payload (ESP),» [En línea]. Available: <https://tools.ietf.org/html/rfc4303>.
- [13] C. Kaufman, «Internet Key Exchange Protocol Version 2 (IKEv2),» [En línea]. Available: <https://tools.ietf.org/html/rfc7296>.
- [14] D. McGrew, «Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH),» [En línea]. Available: <https://tools.ietf.org/html/rfc7321>.
- [15] «IPv6Ready.me - Prueba de compatibilidad IPv6,» [En línea]. Available: https://ipv6ready.me/index.html.es_ES.
- [16] «Hurricane Electric,» [En línea]. Available: <http://he.net/>
- [17] E. Nordmark, «Basic Transition Mechanisms for IPv6 Hosts and Routers», RFC 4213, octubre 2005. [En línea]. Available: <https://datatracker.ietf.org/doc/html/rfc4213>.
- [18] C. Huitema, “Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs),” RFC 4380, Internet Engineering Task Force (IETF), Feb. 2006. [En línea]. Available: <https://datatracker.ietf.org/doc/html/rfc4380>

- https://datatracker.ietf.org/doc/html/rfc4380.
- [19] «hide.me,» [En línea]. Available: <https://hide.me/es/>
 - [20] «WireGuard,» [En línea]. Available: <https://www.wireguard.com/>
 - [21] «VPS,» [En línea]. Available: <https://www.hetzner.com/>.
 - [22] «TestIPv6.net - Prueba de IPv6,» [En línea]. Available: <https://www.testipv6.net/es>
 - [23] Amazon Web Services, «¿Qué es la latencia?,» [En línea]. Available: <https://aws.amazon.com/es/what-is/latency/>.
 - [24] B. Jonglez y J. Chroboczek, «Delay-Based Metric Extension for the Babel Routing Protocol,» RFC 9616, Sept. 2024. [En línea]. Available: <https://datatracker.ietf.org/doc/html/rfc9616>.
 - [25] Ibertronica, «Latencia: ¿qué es?,» [En línea]. Available: <https://ibertronica.es/blog/actualidad/latencia-que-es/>.
 - [26] VAS Experts, «Network packet loss – Pérdida de paquetes de red,» [En línea]. Available: <https://vasexperts.com/es/resources/glossary/network-packet-loss/>.
 - [27] Obkio, «What Is Acceptable Packet Loss? (And How to Fix It),» [En línea]. Available: <https://obkio.com/blog/acceptable-packet-loss/>.
 - [28] CCNA Desde Cero, «¿Qué es ancho de banda o bandwidth?,» [En línea]. Available: <https://ccnadesdecero.es/que-es-ancho-de-banda-o-bandwidth/>.
 - [29] CenturyLink, «What internet speed do I need?,» [En línea]. Available: <https://www.centurylink.com/home/help/internet/what-internet-speed-do-i-need.html>.
 - [30] C. Demichelis, «IP Packet Delay Variation Metric for IPPM,» [En línea]. Available: <https://tools.ietf.org/html/rfc3393>.
 - [31] J. Postel, «Internet Protocol,» [En línea]. Available: <https://tools.ietf.org/html/rfc791>.
 - [32] P. Mockapetris, «Domain Names - Implementation and Specification,» [En línea]. Available: <https://tools.ietf.org/html/rfc1035>.
 - [33] Catchpoint, «Slow DNS: Understanding DNS Performance Best Practices and Monitoring,» [En línea]. Available: <https://www.catchpoint.com/dns-monitoring/slow-dns>.
 - [34] Northwest Performance Software, «NetScanTools Pro - Network Discovery and Diagnostic Tools,» [En línea]. Available: <https://www.netscantools.com/nstpro.html>.
 - [35] Pingman Tools, «PingPlotter - Network troubleshooting made easy,» [En línea]. Available: <https://www.pingplotter.com/>.
 - [36] Ubunlog, “MTR, herramienta de diagnóstico de red,” Ubunlog, [En línea]. Available: <https://ubunlog.com/mtr-herramienta-diagnostico-red/>.
 - [37] The Wireshark Foundation, «Wireshark: Go Deep,» [En línea]. Available: <https://www.wireshark.org/>.
 - [38] Cloudflare, «Cloudflare Speed Test,» [En línea]. Available: <https://speed.cloudflare.com/>

ÍNDICE DE CONCEPTOS

Definimos los siguientes conceptos:

- **Dirección MAC:** Identificador de 48 bits que se asigna a cada equipo de red. Es un identificador único para cada dispositivo utilizado para la comunicación entre ellos por la red.
- **DNS:** El servidor DNS tiene la función de traducir los nombres de dominio a su dirección IP asociada. Posee una base de datos donde almacena esta relación, permitiendo que las distintas webs carguen sus páginas de manera correcta.
- **Caché DNS:** Almacena temporalmente las direcciones IP de los dominios que se ha visitado en esa unidad de tiempo. Permite una navegación más rápida y eficiente.
- **Encapsulamiento:** Proceso para añadir más información a un elemento de datos cuando viaja por la red desde un origen hasta un destino.
- **Cabecera de extensión:** Estructuras añadidas entre la cabecera del protocolo IPv6 y las cabeceras de protocolos de nivel superior, permitiendo realizar funciones opcionales.
- **Flujo:** Conjunto de paquetes que comparten una estrategia de encaminamiento común. Son paquetes tratados de la misma manera.
- **Enlace:** Segmento de red que hace referencia a un conjunto o agrupación de varios equipos conectados entre sí en una red.
- **Host:** Dispositivo final de la red que genera o consume tráfico.
- **Interfaz:** Elemento que permite a los equipos conectarse a una red.
- **Nodo:** Equipo conectado a la red.
- **Paquete:** Unidad de datos enviados por la red.
- **RFC:** Documentos que describen estándares y protocolos de Internet.
- **Router:** Equipo que conecta varias redes y retransmite los paquetes que no van destinados a él.
- **TTL:** Campo que indica el número máximo de saltos que un paquete puede realizar por la red antes de pasar a ser descartado.
- **IANA:** Organización internacional encargada de la asignación de direcciones IP.
- **RIR:** Organismo regional encargado de asignar bloques de direcciones IP a los distintos proveedores de red.
- **CIDR:** Notación utilizada para expresar los prefijos de red.