



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Trabajo 1: Túnel EoIP con routers dd-wrt

TRABAJO RCO

Grado en Ingeniería Informática

Autor: Javier Blasco Romeu
Álvaro Camino Tirapu
Alejandro Salinas Delgado
Grupo: 169

Curso 2023-2024

Resumen

En este proyecto, nos sumergimos en la configuración de un túnel EoIP entre dos routers de nuestra red local: ddwrt-noX y ddwrt-X. La esencia de esta tarea radica en aprovechar la tecnología DDWRT para establecer una conexión entre dos routers, incluso si se encuentran en ubicaciones completamente remotas. El punto de llevar a cabo esto, es conseguir que los routers parezcan estar conectados como si un cable físico los uniera.

Una vez hayamos completado la configuración, pondremos a prueba la funcionalidad del túnel con una serie de pruebas exhaustivas. Observaremos cómo esta tecnología permite la transmisión de paquetes IP, explorando su capacidad para simular una conexión directa.

Palabras clave: EoIP, Túnel, IP, DHCP, router, datagrama, LAN

Abstract

In this project, we delve into setting up an EoIP tunnel between two routers in our local network: ddwrt-noX and ddwrt-X. The essence of this task lies in harnessing DDWRT technology to establish a connection between these routers, even if they are located in entirely remote locations. The goal is to make these routers appear connected as if a physical cable were linking them.

Once we have completed the configuration, we will test the tunnel's functionality through a series of comprehensive tests. We will observe how this technology facilitates the transmission of IP packets, exploring its ability to simulate a direct connection.

Key words: EoIP, Tunnel, IP, DHCP, router , datagram, LAN

Índice general

Índice general	V
Índice de figuras	VII
Índice de tablas	VIII

1	Introducción	1
1.1	Objetivos	3
2	Configuración	5
3	Pruebas de funcionamiento	13
4	Funcionamiento del túnel	17
5	Conclusiones	21
5.0.1	Ampliación grupo 3 alumnos	21
5.0.2	Conclusiones finales	22
	Bibliografía	23

Índice de figuras

1.1	Esquema de red personalizado para el grupo 169	3
2.1	Configuración para ddwrt-noX	5
2.2	Configuración para ddwrt-X	5
2.3	Ifconfig para IP WAN de ddwrt-noX	6
2.4	Ifconfig para IP WAN de ddwrt-X	6
2.5	IP dd-wrt NoX ANTES	7
2.6	IP dd-wrt NoX DESPUÉS	7
2.7	Configuración de VMnet1	7
2.8	Comprobación de acceso a la ip estática asignada mediante MobaXterm .	8
2.9	Comprobación de acceso a la ip estática asignada mediante MobaXterm .	8
2.10	Ifdown y ifup para aplicar los cambios	8
2.11	Cambio defroute=no en ens33	9
2.12	Ifconfig ens37 y modificaciones en archivo de configuración	9
2.13	IP de rco-X asignada mediante DHCP	10
2.14	Ping desde rco-X hasta RCO-noX Via VMnet1	10
2.15	Ping desde rco-X a rco-noX via VMnet8	11
2.16	Ping desde ddwrt-noX(VMnet8) a RCO-X(VMnet2)	11
2.17	Interfaz de ddwrt-noX	12
2.18	Interfaz de ddwrt-X	12
3.1	Ping a RCO-X	13
3.2	Interfaz de ddwrt-noX	14
3.3	Interfaz de ddwrt-X	14
3.4	Página web de RCO-noX	15
3.5	Log de la Página web de RCO-noX	15
3.6	Log de la Página web de RCO-noX tras el cambio	15
3.7	Nueva tabla de routing	16
3.8	Acceso con la nueva regla	16
4.1	Datagrama EtherIP [1]	17
4.2	Cabecera GRE [2]	18
4.3	Parte de la trama capturada con wireshark perteneciente a la cabecera EOIP	19
4.4	Cabecera del protocolo IP exterior a la trama EOIP	19
4.5	Cabecera del protocolo IP interior a la trama EOIP	19
5.1	Captura tiempos transmisión fichero	21
5.2	Captura del paso de los datos por la red vmnet1	21
5.3	captura de la trama desde rco-X a rco-noX pasando por el túnel	22

Índice de tablas

1.1	Valores para la configuración según esquema	3
1.2	Valores para la configuración reales	3

CAPÍTULO 1

Introducción

En este trabajo fundamentalmente vamos a describir el proceso que nos ha llevado a la puesta en marcha del Túnel EOIP. Para ello nos serviremos del esquema de red virtualizado propuesto en la figura 1.1, basado en una red virtual con 4 máquinas VMware: los routers ddwrt-noX y ddwrt-X, y los hosts RCO-noX y RCO-X.

Profundizando un poco en el concepto de túnel, cuando configuramos EoIP con la opción **BRIDGE**, estamos esencialmente creando una conexión virtual que simula la existencia de un puente switch entre las dos redes. En otras palabras, los paquetes Ethernet enviados desde un extremo del túnel son encapsulados y transmitidos a través de Internet, para luego ser desencapsulados en el otro extremo y entregados a la red local **como si nunca hubieran abandonado esa red**.

Para entender un poco mejor de qué estamos hablando cuando nos referimos a túneles EOIP, haremos una breve explicación acerca de los túneles, el encapsulamiento de paquetes y otros conceptos que facilitarán una mayor comprensión de las conclusiones y datos extraídos.

El túnel **Ethernet sobre IP (EoIP)** de MikroTik es un protocolo basado en *GRE RFC 1701* que establece un túnel Ethernet entre dos routers sobre una conexión IP. Como mencionamos anteriormente cuando se activa la función bridge, el tráfico es transportado por una pasarela, de forma que su comportamiento actúa como si hubiera un cable físico entre los dos extremos, como si hubiera una conexión directa entre las interfaces de red, dando así la imagen de red única.

Para ello, se utilizan protocolos de encapsulación como **GRE** el cual facilita el establecimiento de conexiones punto a punto directas a través de la red,

Por ello, antes de llegar a nuestro objetivo, será necesario mencionar una serie de modificaciones previas en lo que se refiere a la configuración de las IPs tanto estáticas como dinámicas (DHCP), así como las configuraciones de los adaptadores de red en uso.

Configurar el servicio tunel EoIP

1. Comprobar que la configuración es la misma que en la práctica 0.
2. Averiguar la IP de la Wan de los routers asignada por DHCP.
 - a) Demostración desde la consola con la orden `ifconfig eth0` y el resultado.
3. Cambiar de las IPs de las redes locales en los dos routers.
 - a) Demostración del antes y después de la modificación.
4. Cambiar en el PC la IP estática del VMnet1.
 - a) Demostración del antes y el después de la modificación.
 - b) Capturar una vez accedemos al router `ddwrt-noX` con su IP local.
5. Cambiar la configuración de la tarjeta Network Adapter 2 en RCO-noX.
 - a) Mostrar modificación en la configuración
6. Cambiar en RCO-noX
 - a) Capturar el gateway de la interfaz `ens33` desactivada.
 - b) Capturar el archivo de configuración de `ens37` después de cambiar la IP y el gateway.
 - c) Capturar el resultado que nos tiene que aparecer tras usar el comando `ifup` e `ifdown`.
7. Configurar el túnel EoIP
 - a) Comprobación de que `rco-X` tiene asignada ip correcta por DHCP
 - b) Comprobación de comunicación entre diferentes componentes de las diferentes VMnets via pings
 - c) Capturar la interfaz web de los DOS routers con la IP de la Wan correspondiente

1.1 Objetivos

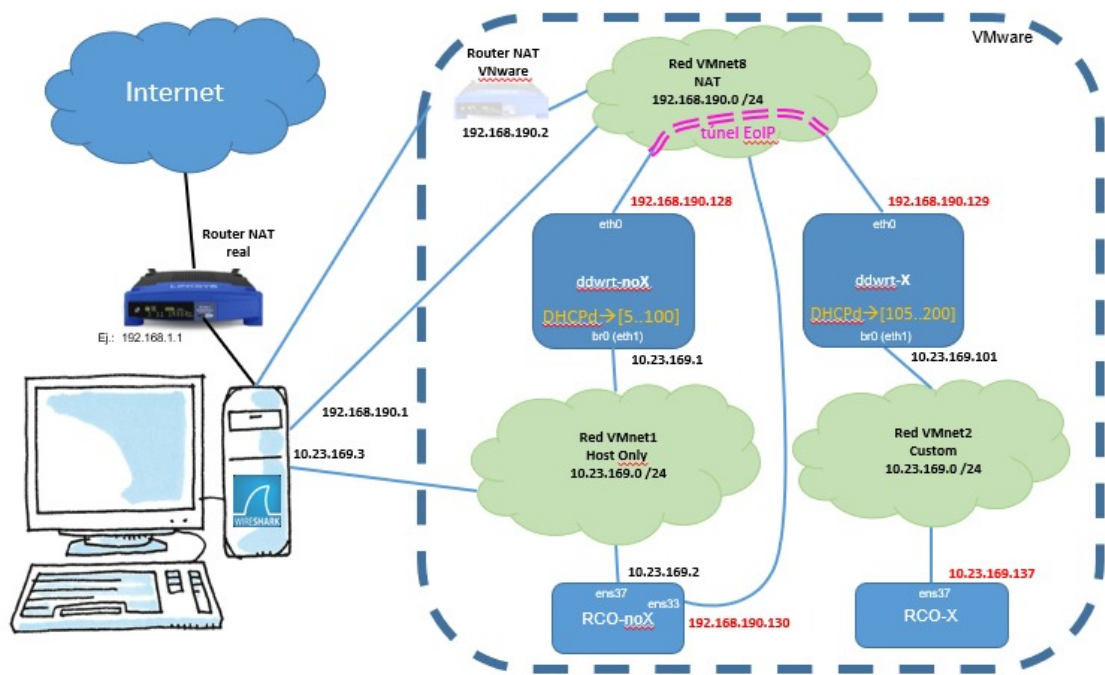


Figura 1.1: Esquema de red personalizado para el grupo 169

Con los datos que podemos obtener del esquema de la figura 1.1 podemos hacer la tabla 1.1. Igualmente esta primera tabla es la que habría que poner.

	PC	dd-wrt NoX	dd-wrt X	RCO-noX	RCO-X
VMnet8	192.168.239.1	dhcp-1	dhcp-2	dhcp-3	-
VMnet1	10.23.1.3	10.23.1.1	-	10.23.1.2	-
VMnet2	-		10.23.1.101	-	dhcp-4
EoIP Tunel	-	Tunnel 1	Tunnel 1	-	-
EoiP Remote IP	-	dhcp-2	dhcp-1	-	-

Tabla 1.1: Valores para la configuración según esquema

Una vez tengamos todo funcionando, anotaremos las IPs asignadas por DHCP, como se muestra en la tabla 1.1. El resultado sera la tabla 1.2 que tenemos a continuación.

Esta tabla, junto con la figura anterior completa (con las direcciones en rojo correctas) se pondrán en el capítulo siguiente de configuración.

	PC	dd-wrt NoX	dd-wrt X	RCO-noX	RCO-X
VMnet8	192.168.190.1	192.168.190.128	192.168.190.129	192.168.190.130	-
VMnet1	10.23.169.3	10.23.169.1	-	10.23.169.2	-
VMnet2	-		10.23.169.101	-	10.23.169.137
EoIP Tunel	-	Tunnel 1	Tunnel 1	-	-
EoiP Remote IP	-	192.168.190.128	192.168.190.129	-	-

Tabla 1.2: Valores para la configuración reales

CAPÍTULO 2

Configuración

1. Comprobar que la configuración es la misma que en la práctica 0.

Como mencionamos anteriormente, el primer paso de todos es asegurarnos de que la configuración de los dos routers **se encuentra como la práctica 0**. Revisamos que efectivamente ddwrt-noX y ddwrt-X, tienen las configuraciones de Adaptadores que les corresponde. Mostramos a continuación las figuras 2.1 y 2.2 las cuales muestran las respectivas configuraciones.

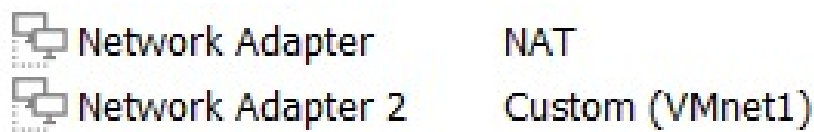


Figura 2.1: Configuración para ddwrt-noX

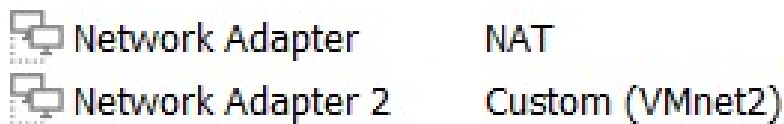
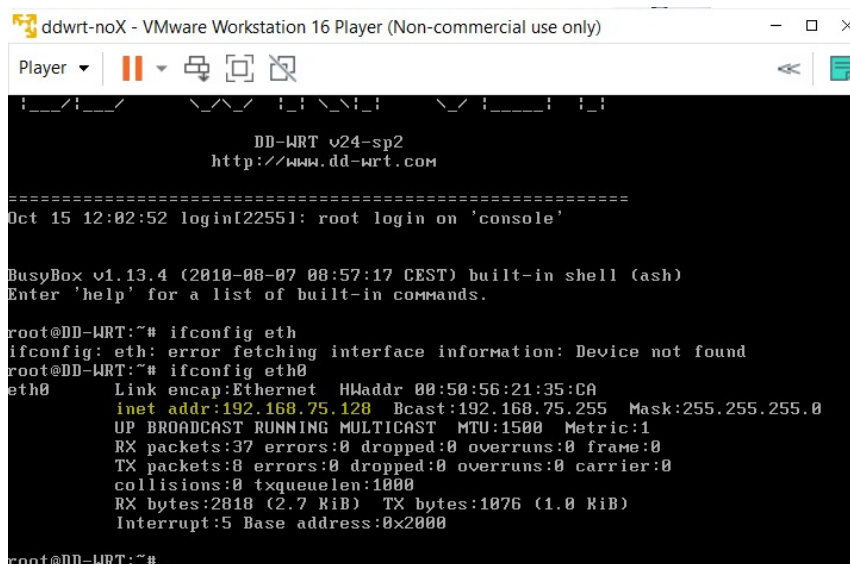


Figura 2.2: Configuración para ddwrt-X

2. Averiguar la IP de la Wan de los routers asignada por DHCP.

Para obtener la IP de la Wan asignada por DHCP de ambos routers hemos tenido que hacer un ifconfig como se muestra en las figuras de a continuación 2.3 (para el router ddwrt-noX) y 2.4 (para el router ddwrt-X).



```

Oct 15 12:02:52 login[22551]: root login on 'console'

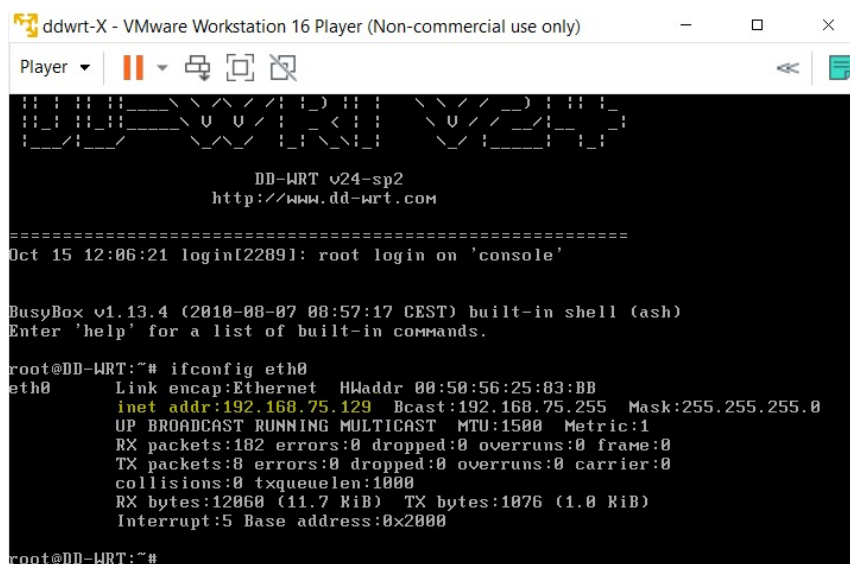
BusyBox v1.13.4 (2010-08-07 08:57:17 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

root@DD-WRT:~# ifconfig eth
ifconfig: eth: error fetching interface information: Device not found
root@DD-WRT:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:56:21:35:CA
          inet addr:192.168.75.128  Bcast:192.168.75.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:37 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2818 (2.7 KiB)  TX bytes:1076 (1.0 KiB)
          Interrupt:5 Base address:0x2000

root@DD-WRT:~#

```

Figura 2.3: Ifconfig para IP WAN de ddwrt-noX



```

Oct 15 12:06:21 login[22891]: root login on 'console'

BusyBox v1.13.4 (2010-08-07 08:57:17 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

root@DD-WRT:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:56:25:83:BB
          inet addr:192.168.75.129  Bcast:192.168.75.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:182 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12060 (11.7 KiB)  TX bytes:1076 (1.0 KiB)
          Interrupt:5 Base address:0x2000

root@DD-WRT:~#

```

Figura 2.4: Ifconfig para IP WAN de ddwrt-X

3. Cambiar de las IPs de las redes locales en los dos routers.

Para poder trabajar con la red correctamente, otro de los cambios que tendremos que hacer en la configuración interna de los routers es cambiar las IPs locales que venían por defecto de la práctica 0. Como ejemplo hemos tomado el router dd-wrt NoX para mostrar su configuración antes con la IP por defecto (figura 2.5) y después de su modificación (figura 2.6). Para ello, hemos modificado el tercer octeto de la Local IP Address por el número de nuestro grupo (169) siendo al final la IP: 10.23.169.1. También podemos observar que la Subnet Mask no ha sido modificada en la figura 2.6 en comparación con la figura 2.5.

Network Setup				
Router IP				
Local IP Address	192	168	200	1
Subnet Mask	255	255	255	0
Gateway	0	0	0	0
Local DNS	0	0	0	0

Figura 2.5: IP dd-wrt NoX ANTES

Network Setup				
Router IP				
Local IP Address	10	23	169	1
Subnet Mask	255	255	255	0
Gateway	0	0	0	0
Local DNS	0	0	0	0

Figura 2.6: IP dd-wrt NoX DESPUÉS

4. Cambiar en el PC la IP estática del VMnet1.

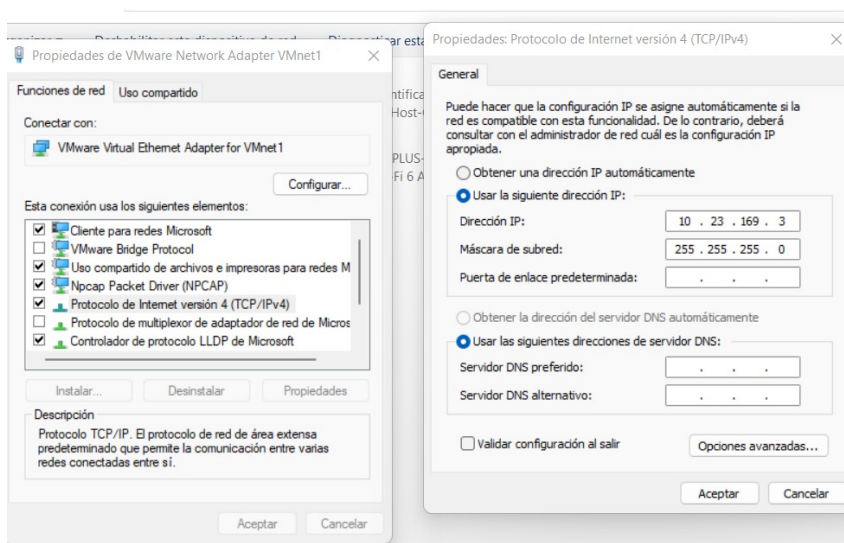
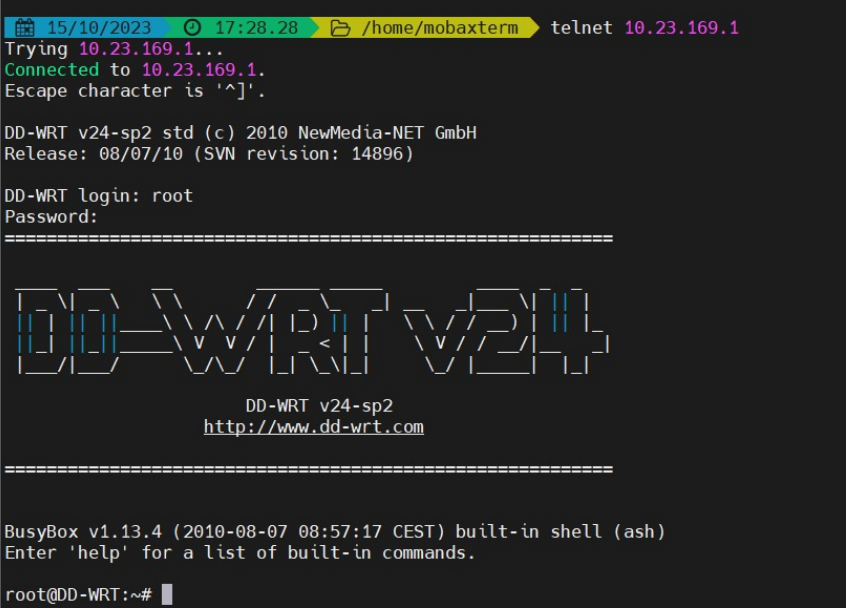


Figura 2.7: Configuración de VMnet1

Para poder realizar el cambio tendremos que acceder a la ventana de 'Configuración de Red e Internet' seguido de 'Conexiones de red', donde ya encontramos el adaptador de red VMnet1. Accedemos a las propiedades del VMnet1, mostrándose así la pestaña de la izquierda de la figura 2.7, y situados en la pestaña de arriba 'Funciones de red' abrimos las '*Propiedades del protocolo de Internet versión 4 (TCP/IPv4)*' (pestaña de la derecha de la figura 2.7). En esa nueva pestaña tenemos que seleccionar la opción '*Usar la siguiente dirección IP*' y añadimos la IP 10.23.169.3 (tercer dígito nuestro número de grupo) y la máscara de subred 255.255.255.0. Una vez configurado todo de manera correcta podemos comprobar el acceso a la IP estática asignada mediante **MobaXterm** como se muestra en la figura 2.8 que está a continuación.



```

15/10/2023 17:28.28 /home/mobaxterm telnet 10.23.169.1
Trying 10.23.169.1...
Connected to 10.23.169.1.
Escape character is '^'.

DD-WRT v24-sp2 std (c) 2010 NewMedia-NET GmbH
Release: 08/07/10 (SVN revision: 14896)

DD-WRT login: root
Password:
=====
DD-WRT v24-sp2
http://www.dd-wrt.com
=====
BusyBox v1.13.4 (2010-08-07 08:57:17 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

root@DD-WRT:~#

```

Figura 2.8: Comprobación de acceso a la ip estática asignada mediante MobaXterm

5. Cambiar la configuración de la tarjeta Network Adapter 2 en RCO-noX.

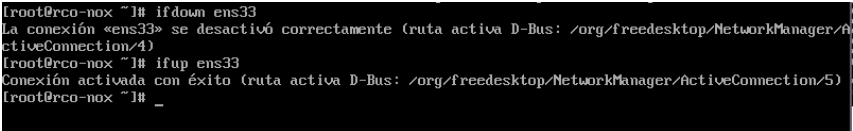
Nos aseguramos de cambiar Network Adapter 1 de Bridged a NAT como vemos en la figura 2.9.

	Network Adapter	NAT
	Network Adapter 2	Custom (VMnet1)

Figura 2.9: Comprobación de acceso a la ip estática asignada mediante MobaXterm

6. Cambios en RCO-noX

A continuación, nos metemos en la configuración de la tarjeta del adaptador de red de RCO-noX, como se nos indica en la práctica. Llegamos finalmente al fichero de configuración de la **interfaz ens33** donde indicamos **DEFROUTE=no**, como se observa en la figura 2.11. Tras hacer ifdown e ifup, como se muestra en la figura 2.10, comprobamos que los ajustes se han guardado y la interfaz se ha levantado correctamente.



```

[root@rc0-nox ~]# ifdown ens33
La conexión «ens33» se desactivó correctamente (ruta activa D-Bus: /org/freedesktop/NetworkManager/ActiveConnection/4)
[root@rc0-nox ~]# ifup ens33
Conexión activada con éxito (ruta activa D-Bus: /org/freedesktop/NetworkManager/ActiveConnection/5)
[root@rc0-nox ~]# _

```

Figura 2.10: Ifdown y ifup para aplicar los cambios


```

TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="dhcp"
DEFROUTE="no"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="ens33"
UUID="3b0c93eb-7c90-4170-8514-e1ac53069254"
DEVICE="ens33"
ONBOOT="yes"

```

Figura 2.11: Cambio defroute=no en ens33

Por otro lado, nos falta cambiar a su vez la configuración de la **interfaz ens37**, a los valores también indicados en la práctica. Accediendo a la dirección indicada en la práctica, se nos muestra en el editor el archivo de configuración de ens37, donde modificamos los campos **IPADDR** y **GATEWAY** (salida a internet), introduciendo 10.23.169.2 y 10.23.169.1 (ip local del router), respectivamente, quedándonos la configuración como se muestra en la figura 2.12.

```

TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="none"
IPV4_FAILURE_FATAL="no"
IPV6INIT="no"
NAME="ens37"
DEVICE="ens37"
ONBOOT="yes"
IPADDR=10.23.169.2
GATEWAY=10.23.169.1
PREFIX=24
UUID="c5b60d3c-f3f5-4e8d-8267-da3c24dfec47"

```

Figura 2.12: Ifconfig ens37 y modificaciones en archivo de configuración

7. Configurar el túnel EoIP

Una vez seguido y comprobado que la conectividad de todos los componentes de nuestra red sea la correcta procederemos a configurar el túnel EoIP.

- a) Comprobación de que rco-X tiene asignada ip correcta por DHCP

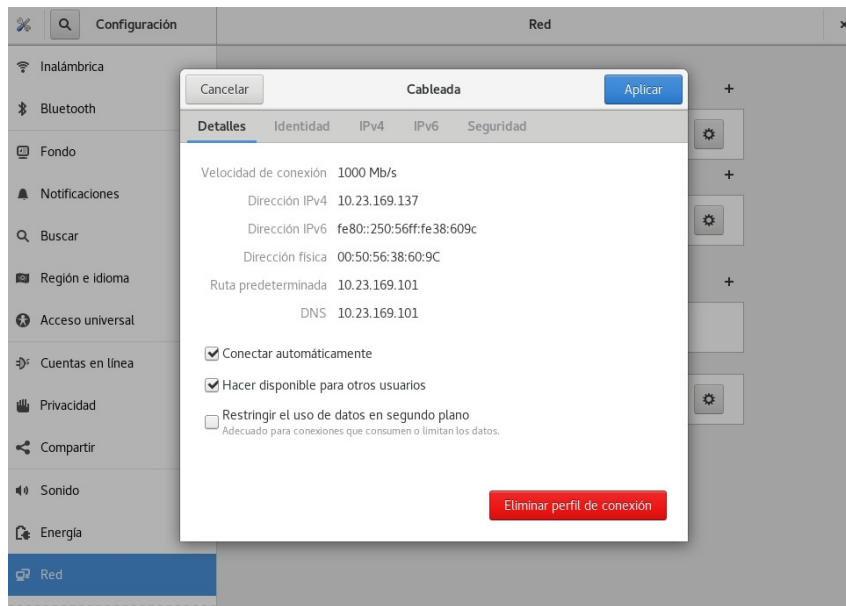


Figura 2.13: IP de rco-X asignada mediante DHCP

En esta figura, la 2.13, podemos ver como la ip asignada corresponde con la deseada, ya que pertenece al rango que otorga el dhcp de ddwrt-X(105-200)

- b) Comprobación de comunicación entre diferentes componentes de las diferentes VMnets via pings.

En este apartado accedimos a varias máquinas para realizar pings desde ellas a otros componentes de la red, a través de las varias VMnets del sistema.

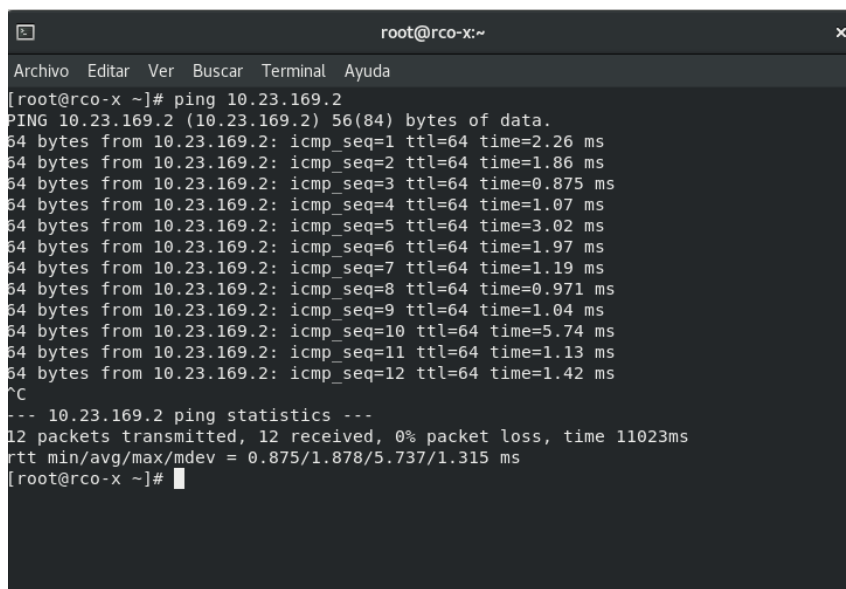


Figura 2.14: Ping desde rco-X hasta RCO-noX Via VMnet1

El ping de la figura 2.14 pasa por la VMnet 2, atraviesa el router ddwrt-X, donde recibe una cabecera EOIP y se encamina a través del tunel EOIP, en la red VMnet 8, llega a ddwrt-noX donde se desencapsula y se envía a través de la VMnet1 paraa llegar a RCO-noX

```
[root@rco-x ~]# ping 192.168.190.130
PING 192.168.190.130 (192.168.190.130) 56(84) bytes of data.
64 bytes from 192.168.190.130: icmp_seq=1 ttl=63 time=5.16 ms
64 bytes from 192.168.190.130: icmp_seq=2 ttl=63 time=1.22 ms
64 bytes from 192.168.190.130: icmp_seq=3 ttl=63 time=1.80 ms
64 bytes from 192.168.190.130: icmp_seq=4 ttl=63 time=1.46 ms
64 bytes from 192.168.190.130: icmp_seq=5 ttl=63 time=0.749 ms
^C
--- 192.168.190.130 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4020ms
rtt min/avg/max/mdev = 0.749/2.080/5.163/1.579 ms
[root@rco-x ~]#
```

Figura 2.15: Ping desde rco-X a rco-noX via VMnet8

El ping de la figura 2.15 pasa por la VMnet 2, atraviesa el router ddwrt-X y se encamina a través de la VMnet 8 directamente hasta RCO-noX, debido a que este también cuenta con una conexión directa con la VMnet 8.

```
root@DD-WRT:~# ping 10.23.169.137
PING 10.23.169.137 (10.23.169.137): 56 data bytes
64 bytes from 10.23.169.137: seq=0 ttl=64 time=3.994 ms
64 bytes from 10.23.169.137: seq=1 ttl=64 time=0.699 ms
64 bytes from 10.23.169.137: seq=2 ttl=64 time=0.739 ms
64 bytes from 10.23.169.137: seq=3 ttl=64 time=0.819 ms
64 bytes from 10.23.169.137: seq=4 ttl=64 time=0.965 ms
64 bytes from 10.23.169.137: seq=5 ttl=64 time=0.809 ms
--- 10.23.169.137 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 0.699/1.337/3.994 ms
root@DD-WRT:~#
```

Figura 2.16: Ping desde ddwrt-noX(VMnet8) a RCO-X(VMnet2)

En este último caso, el de la figura 2.16, el Ping se realiza desde ddwrt-noX. Este encapsula la trama y la envía a través del tunel EOIP(VMnet 8) y es des-encapsulada en ddwrt-X, que la envía a través de la VMnet 2 a RCO-X.

Con esto terminamos de comprobar varios de los enlaces entre los diferentes componentes de las diferentes VMnets

- c) Capturar la interfaz web de los DOS routers con la IP de la Wan correspondiente

A continuación se presentan las dos capturas, figuras 2.17 y 2.18, mostrando la **interfaz oet1**, que aparece al configurar el tunel EOIP. Como podemos observar, encontramos interfaces en modo promiscuo tanto en br0 como en br0:0 en caso de ddwrt-noX, como en br0:0 en ddwrt-X.

En lugar de ignorar los paquetes que no están dirigidos a su dirección MAC, una interfaz en modo promiscuo **escucha y analiza** todos los paquetes que pasan por la red [3]. Esto es especialmente útil para herramientas de análisis de red y diagnóstico, ya que permite examinar el tráfico general de la red y obtener información detallada sobre la comunicación entre diferentes dispositivos.

```

br0      Link encap:Ethernet  HWaddr 00:50:56:2C:2A:18
         inet addr:10.23.169.1  Bcast:10.23.169.255  Mask:255.255.255.0
         UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
         RX packets:1804988  errors:0 dropped:0 overruns:0 frame:0
         TX packets:239  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:2446650696 (2.2 GiB)  TX bytes:18456 (18.0 KiB)

br0:0    Link encap:Ethernet  HWaddr 00:50:56:2C:2A:18
         inet addr:169.254.255.1  Bcast:169.254.255.255  Mask:255.255.0.0
         UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1

eth0     Link encap:Ethernet  HWaddr 00:50:56:21:35:CA
         inet addr:192.168.190.128  Bcast:192.168.190.255  Mask:255.255.255.0
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:3250737  errors:24 dropped:26 overruns:0 frame:0
         TX packets:178098  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:2573045844 (2.3 GiB)  TX bytes:19121472 (18.2 MiB)
         Interrupt:5 Base address:0x2000

eth1     Link encap:Ethernet  HWaddr 00:50:56:2C:2A:18
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:177977  errors:0 dropped:0 overruns:0 frame:0
         TX packets:1627250  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:12712720 (12.1 MiB)  TX bytes:2459226264 (2.2 GiB)
         Interrupt:9 Base address:0x2080

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         UP LOOPBACK RUNNING MULTICAST  MTU:16436  Metric:1
         RX packets:6  errors:0 dropped:0 overruns:0 frame:0
         TX packets:6  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:354 (354.0 B)  TX bytes:354 (354.0 B)

oet1     Link encap:Ethernet  HWaddr 16:2E:18:50:9D:4A
         UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
         RX packets:1627011  errors:0 dropped:0 overruns:0 frame:0
         TX packets:177739  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0

```

Figura 2.17: Interfaz de ddwrt-noX

```

br0      Link encap:Ethernet  HWaddr 00:50:56:3E:AB:C9
         inet addr:10.23.169.101  Bcast:10.23.169.255  Mask:255.255.255.0
         UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
         RX packets:3565472  errors:0 dropped:0 overruns:0 frame:0
         TX packets:2976535  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:600898199 (573.0 MiB)  TX bytes:3513225993 (3.2 GiB)

br0:0    Link encap:Ethernet  HWaddr 00:50:56:3E:AB:C9
         inet addr:169.254.255.1  Bcast:169.254.255.255  Mask:255.255.0.0
         UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1

eth0     Link encap:Ethernet  HWaddr 00:50:56:25:83:BB
         inet addr:192.168.190.129  Bcast:192.168.190.255  Mask:255.255.255.0
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:3154554  errors:171 dropped:300 overruns:0 frame:0
         TX packets:5011129  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:3532351009 (3.2 GiB)  TX bytes:751873566 (717.0 MiB)
         Interrupt:5 Base address:0x2000

eth1     Link encap:Ethernet  HWaddr 00:50:56:3E:AB:C9
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:3387708  errors:1069 dropped:1100 overruns:0 frame:0
         TX packets:3154299  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:638117528 (608.5 MiB)  TX bytes:3525923272 (3.2 GiB)
         Interrupt:9 Base address:0x2080

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         UP LOOPBACK RUNNING MULTICAST  MTU:16436  Metric:1
         RX packets:6  errors:0 dropped:0 overruns:0 frame:0
         TX packets:6  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:270 (270.0 B)  TX bytes:270 (270.0 B)

oet1     Link encap:Ethernet  HWaddr 3A:C0:33:9A:99:BE
         UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
         RX packets:177764  errors:0 dropped:0 overruns:0 frame:0
         TX packets:1627354  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0

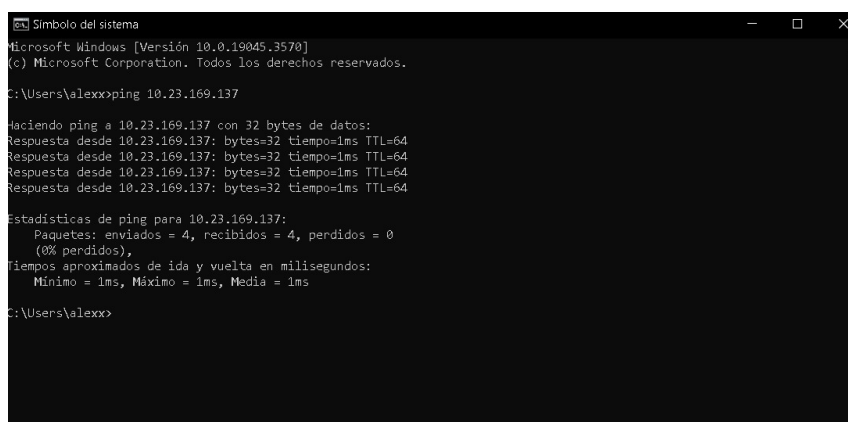
```

Figura 2.18: Interfaz de ddwrt-X

CAPÍTULO 3

Pruebas de funcionamiento

En la **primera prueba** se realizó un ping a la dirección local de rco-X para comprobar que el tunel funcionaba, esta prueba fue exitosa, como demuestra la figura 3.1.



```
Simbolo del sistema
Microsoft Windows [Versión 10.0.19045.3570]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\alexx>ping 10.23.169.137

Haciendo ping a 10.23.169.137 con 32 bytes de datos:
Respuesta desde 10.23.169.137: bytes=32 tiempo=1ms TTL=64
Respuesta desde 10.23.169.137: bytes=32 tiempo=1ms TTL=64
Respuesta desde 10.23.169.137: bytes=32 tiempo=1ms TTL=64
Respuesta desde 10.23.169.137: bytes=32 tiempo=1ms TTL=64

Estadísticas de ping para 10.23.169.137:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 1ms, Máximo = 1ms, Media = 1ms

C:\Users\alexx>
```

Figura 3.1: Ping a RCO-X

Debido al resultado del ping se verificó que el tunel estaba activo, ya que esta era la única conexión de rco-X con el exterior de la red.

Para la **segunda prueba** se realizaron sendas fotografias de las interfaces de los routers, figuras 3.2 y 3.3 :


```

br0      Link encap:Ethernet  HWaddr 00:50:56:2C:2A:18
         inet addr:10.23.169.1  Bcast:10.23.169.255  Mask:255.255.255.0
         UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
         RX packets:1804988  errors:0 dropped:0 overruns:0 frame:0
         TX packets:239  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:2446650696 (2.2 GiB)  TX bytes:18456 (18.0 KiB)

br0:0    Link encap:Ethernet  HWaddr 00:50:56:2C:2A:18
         inet addr:169.254.255.1  Bcast:169.254.255.255  Mask:255.255.0.0
         UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1

eth0     Link encap:Ethernet  HWaddr 00:50:56:21:35:CA
         inet addr:192.168.190.128  Bcast:192.168.190.255  Mask:255.255.255.0
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:3250737  errors:24 dropped:26 overruns:0 frame:0
         TX packets:178098  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:2573045844 (2.3 GiB)  TX bytes:19121472 (18.2 MiB)
         Interrupt:5 Base address:0x2000

eth1     Link encap:Ethernet  HWaddr 00:50:56:2C:2A:18
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:177977  errors:0 dropped:0 overruns:0 frame:0
         TX packets:1627250  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:12712720 (12.1 MiB)  TX bytes:2459226264 (2.2 GiB)
         Interrupt:9 Base address:0x2080

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         UP LOOPBACK RUNNING MULTICAST  MTU:16436  Metric:1
         RX packets:6  errors:0 dropped:0 overruns:0 frame:0
         TX packets:6  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:354 (354.0 B)  TX bytes:354 (354.0 B)

oet1     Link encap:Ethernet  HWaddr 16:2E:18:50:9D:4A
         UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
         RX packets:1627011  errors:0 dropped:0 overruns:0 frame:0
         TX packets:177739  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0

```

Figura 3.2: Interfaz de ddwrt-noX

```

br0      Link encap:Ethernet  HWaddr 00:50:56:3E:AB:C9
         inet addr:10.23.169.101  Bcast:10.23.169.255  Mask:255.255.255.0
         UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
         RX packets:3565472  errors:0 dropped:0 overruns:0 frame:0
         TX packets:2976535  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:600898199 (573.0 MiB)  TX bytes:3513225993 (3.2 GiB)

br0:0    Link encap:Ethernet  HWaddr 00:50:56:3E:AB:C9
         inet addr:169.254.255.1  Bcast:169.254.255.255  Mask:255.255.0.0
         UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1

eth0     Link encap:Ethernet  HWaddr 00:50:56:25:83:BB
         inet addr:192.168.190.129  Bcast:192.168.190.255  Mask:255.255.255.0
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:3154554  errors:171 dropped:300 overruns:0 frame:0
         TX packets:5011129  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:3532351009 (3.2 GiB)  TX bytes:751873566 (717.0 MiB)
         Interrupt:5 Base address:0x2000

eth1     Link encap:Ethernet  HWaddr 00:50:56:3E:AB:C9
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:3387708  errors:1069 dropped:1100 overruns:0 frame:0
         TX packets:3154299  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:638117528 (608.5 MiB)  TX bytes:3525923272 (3.2 GiB)
         Interrupt:9 Base address:0x2080

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         UP LOOPBACK RUNNING MULTICAST  MTU:16436  Metric:1
         RX packets:6  errors:0 dropped:0 overruns:0 frame:0
         TX packets:6  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:270 (270.0 B)  TX bytes:270 (270.0 B)

oet1     Link encap:Ethernet  HWaddr 3A:C0:33:9A:99:BE
         UP BROADCAST RUNNING PROMISC MULTICAST  MTU:1500  Metric:1
         RX packets:177764  errors:0 dropped:0 overruns:0 frame:0
         TX packets:1627354  errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0

```

Figura 3.3: Interfaz de ddwrt-X

Con estas imágenes se comprobó que hay más interfaces a parte del túnel actuando el modo promíscuo.

Desde el punto de vista software que se preguntaba en esta prueba, el túnel EoIP en DD-WRT funciona como una tecnología que permite que el tráfico Ethernet se transporte a través de una infraestructura de red IP. El software en el enrutador es responsable de la encapsulación y desencapsulación de las tramas Ethernet, así como del enrutamiento de los paquetes IP.

En la **la tercera prueba** realizamos la entrada a la página web, como se puede ver a continuación, figura 3.4:



Figura 3.4: Página web de RCO-noX

```

root@rc0-nox ~# tail /var/log/httpd/access_log
10.23.169.3 - - [15/Oct/2023:17:10:10 +0200] "GET /index.asp HTTP/1.1" 404 196 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36 OPR/102.0.0.0"
10.23.169.3 - - [15/Oct/2023:17:10:11 +0200] "GET /favicon.ico HTTP/1.1" 404 196 "http://10.23.169.2/index.asp" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36 OPR/102.0.0.0"
192.168.190.129 - - [15/Oct/2023:17:37:47 +0200] "GET / HTTP/1.1" 200 481 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
192.168.190.129 - - [15/Oct/2023:17:37:48 +0200] "GET /icons/blank.gif HTTP/1.1" 200 148 "http://192.168.190.130/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
192.168.190.129 - - [15/Oct/2023:17:37:48 +0200] "GET /favicon.ico HTTP/1.1" 404 196 "http://192.168.190.130/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
root@rc0-nox ~#

```

Figura 3.5: Log de la Página web de RCO-noX

Vemos, gracias al log de la figura 3.5, como la IP que ha accedido es la IP de la WAN del router ddwrt-X (192.168.190.129 en nuestra configuración) que es lo que cabía esperar ya que este hace NAT y cambia la IP local de RCO-X por la suya de la WAN.

En la **la cuarta prueba** realizamos los cambios pertinentes, cambiando el servidor DHCP del que obtiene la ip y volvimos a obtener el log de la página web.

```

root@rc0-nox ~# tail /var/log/httpd/access_log
10.23.169.3 - - [15/Oct/2023:17:10:10 +0200] "GET /index.asp HTTP/1.1" 404 196 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36 OPR/102.0.0.0"
10.23.169.3 - - [15/Oct/2023:17:10:11 +0200] "GET /favicon.ico HTTP/1.1" 404 196 "http://10.23.169.2/index.asp" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36 OPR/102.0.0.0"
192.168.190.129 - - [15/Oct/2023:17:37:47 +0200] "GET / HTTP/1.1" 200 481 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
192.168.190.129 - - [15/Oct/2023:17:37:48 +0200] "GET /icons/blank.gif HTTP/1.1" 200 148 "http://192.168.190.130/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
192.168.190.129 - - [15/Oct/2023:17:37:48 +0200] "GET /favicon.ico HTTP/1.1" 404 196 "http://192.168.190.130/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
192.168.190.128 - - [16/Oct/2023:20:08:04 +0200] "GET / HTTP/1.1" 200 481 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
192.168.190.128 - - [16/Oct/2023:20:08:09 +0200] "GET /favicon.ico HTTP/1.1" 404 196 "http://192.168.190.130/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
root@rc0-nox ~#

```

Figura 3.6: Log de la Página web de RCO-noX tras el cambio

En la figura 3.6, se puede observar como en este caso la IP usada es la del nuevo router, que, a su vez, hace de nuevo servidor DHCP, esto es debido a que **ahora es este router el que hace NAT**.

Por último, para **la quinta prueba**, realizamos cambios en la tabla de routing para lograr que el router ddwrt-X saliera a la WAN por el router ddwrt-noX, **eliminando la regla default** y añadiendo otras dos adicionales, dejándola como sigue en la figura 3.7:

```
192.168.190.2 dev eth0 scope link
10.23.169.0/24 dev br0 proto kernel scope link src 10.23.169.101
192.168.190.0/24 dev eth0 proto kernel scope link src 192.168.190.129
169.254.0.0/16 dev br0 proto kernel scope link src 169.254.255.1
127.0.0.0/8 dev lo scope link
default via 10.23.169.1 dev br0
```

Figura 3.7: Nueva tabla de routing

Visitamos la página web y obtuvimos un nuevo log, figura 3.8, gracias a la regla adicional, que permite que cualquier tráfico destinado a esta dirección IP se enrute de manera específica a través de la 10.23... del ddwrt-noX. Esto es útil cuando se necesita asegurarse de que un servidor o recurso específico, como un sitio web, **esté disponible y sea accesible** a través del túnel EoIP. Además, en una oficina con varios PCs, la implementación de una regla específica para la dirección IP de rco-noX puede ayudar a evitar el enrutamiento innecesario de tráfico a través del túnel EoIP para otros dispositivos que no necesitan acceder a esa dirección IP en particular. Esto puede ayudar a optimizar el uso de la red y reducir la congestión en el túnel.

```
[root@rco-nox ~]# tail /var/log/httpd/access_log
192.168.190.129 - - [18/Oct/2023:17:57:40 +0200] "GET / HTTP/1.1" 200 481 "-" "Mozilla/5.0 (X11; Lin
ux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
[root@rco-nox ~]# _
```

Figura 3.8: Acceso con la nueva regla

CAPÍTULO 4

Funcionamiento del túnel

Para entender mejor el funcionamiento del túnel EOIP, hemos de identificar primero **qué es lo que se mueve** dentro de este tipo de túneles y cual es su naturaleza. Cabe destacar que en esta modalidad del túnel las tramas no se cifran, en comparación con otros tipos de túneles que se mencionarán posteriormente. Apoyándonos en las figuras posteriores (4.1 y 4.2), vemos cómo un datagrama que pasa por un túnel EOIP acaba teniendo las siguientes cabeceras:

- Ethernet
- IP (donde aparecen las IPs 'exteriores', es decir, las de los routers fuente y destino, en nuestro caso 192.168.190.129 y 192.168.190.128 respectivamente mostradas en la figura 4.4)
- EtherIP, cabecera que indica que el datagrama viaja en un túnel y que precede al frame ethernet.
- Ethernet, donde viajan las MAC de los hosts finales.
- IP (donde aparecen las IPs 'interiores', las ips privadas, en nuestro caso 10.23.169-137 y 10.23.169.2, mostradas en la figura 4.5)
- ICMP (el propio ping que hemos mandado en la orden y que ha sido encapsulado por las cabeceras anteriores)

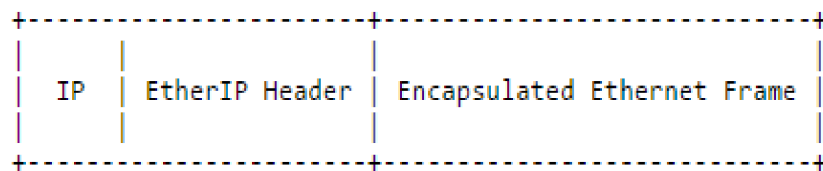


Figura 4.1: Datagrama EtherIP [1]

La encapsulación en el EOIP comienza con el protocolo de Internet (IP) en la **capa 3**. Este se encapsula utilizando la tecnología **Ethernet II en la capa 2**. Los resultados de esta encapsulación luego se encapsulan en el **Protocolo de Encapsulación de Enrutamiento Genérico (GRE)**. [4]

El túnel EoIP añade al menos 42 bytes de sobrecarga (8 bytes de GRE + 14 bytes de Ethernet + 20 bytes de IP). [5]

The GRE packet header has the form:

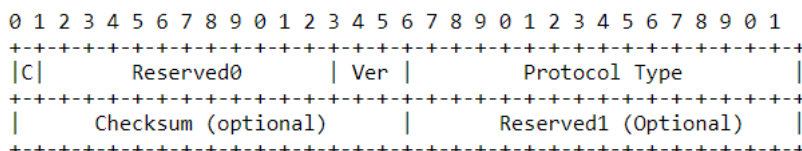


Figura 4.2: Cabecera GRE [2]

En la figura 4.3 observamos tráfico variado al realizar una operación simple de comunicación entre RCO-X (Vmnet 2) y dd-wrtNOx (Vmnet 1) (ping). Vemos la información de diferentes protocolos, a diferencia de EOIP with IPsec, donde no se muestran los protocolos con origen y destino ya que en esta modalidad si que se cifran. [6] Vemos que a raíz de ese momento se manda un ARP Broadcast desde dd-wrtNOx para preguntar por la MAC de RCO-X.

La comunicación ARP se realiza mediante mensajes que contienen la dirección IP del dispositivo de destino y una solicitud del tipo "**Who has (dirección IP)? Tell (dirección IP)**". Cuando un dispositivo emite un mensaje ARP de este tipo, espera recibir una respuesta del dispositivo con la dirección IP solicitada. La respuesta contendrá la dirección MAC correspondiente a la dirección IP buscada. Distinguimos dos casuísticas dentro de ARP:

ARP Broadcast: La máquina virtual emite un mensaje ARP broadcast para preguntar a todas las máquinas virtuales en la misma red quién tiene una determinada dirección IP. Este enfoque se utiliza cuando la máquina virtual no tiene información sobre la dirección MAC correspondiente a una dirección IP específica.

ARP directo entre VMs: Si las máquinas virtuales ya tienen información en su tabla ARP sobre la dirección MAC de la máquina virtual de destino, entonces pueden comunicarse directamente sin necesidad de emitir un ARP broadcast.

Vemos que a raíz de ese momento se manda un ARP Broadcast desde dd-wrtNOx para preguntar por la MAC de RCO-X. En nuestro caso, se le contesta y una vez conoce la MAC y la guarda en la tabla de routing, envía una respuesta del ping (response) a RCO-X.

Observamos en detalle en la figura 4.3 como, al aplicar un sniffer como Wireshark sobre interfaces en modo promiscuo, captan todo tipo de tráfico inicial si no aplicamos ningún filtro específico. Podemos ver desde los **paquetes NTP** de sincronización de relojes intercambiados entre las partes que intervienen en la comunicación, hasta los diferentes datagramas **ICMP** correspondientes a los pings entre 10.23.169. y 10.23.169.137.

No.	Time	Source	Destination	Protocol	Length	Info
14708	1325.845487	192.168.190.129	158.227.98.15	NTP	90	NTP Version 4, client
14709	1325.846125	158.227.98.15	192.168.190.129	NTP	90	NTP Version 4, server
14710	1328.371122	VMware_25:83:bb	VMware_ef:11:ac	ARP	42	Who has 192.168.190.2? Tell 192.168.190.129
14711	1328.371180	VMware_ef:11:ac	VMware_25:83:bb	ARP	42	192.168.190.2 is at 00:50:56:ef:11:ac
14712	1329.847108	192.168.190.129	208.85.20.220	NTP	90	NTP Version 4, client
14713	1329.893548	208.85.20.220	192.168.190.129	NTP	90	NTP Version 4, server
14714	1331.802181	192.168.190.128	158.227.98.15	NTP	90	NTP Version 4, client
14715	1331.823250	158.227.98.15	192.168.190.128	NTP	90	NTP Version 4, server
14716	1336.408011	192.168.190.128	208.85.20.220	NTP	90	NTP Version 4, client
14717	1336.422222	208.85.20.220	192.168.190.128	NTP	90	NTP Version 4, server
14718	1336.799508	VMware_21:35:ca	VMware_ef:11:ac	ARP	42	Who has 192.168.190.2? Tell 192.168.190.128
14719	1336.799559	VMware_ef:11:ac	VMware_21:35:ca	ARP	42	Who has 192.168.190.2? Tell 192.168.190.128
14720	1346.867733	10.23.169.137	10.23.169.1	ICMP	134	Echo (ping) request id=0x0003, seq=1/256, ttl=64 (reply in 14723)
14721	1346.871971	VMware_2c:2a:18	Broadcast	ARP	78	Who has 10.23.169.137? Tell 10.23.169.1
14722	1346.872408	VMware_38:60:9c	VMware_2c:2a:18	ARP	96	10.23.169.137 is at 00:50:56:38:60:9c
14723	1346.872620	10.23.169.1	10.23.169.137	ICMP	134	Echo (ping) reply id=0x0003, seq=1/256, ttl=64 (request in 14720)
14724	1347.869573	10.23.169.137	10.23.169.1	ICMP	134	Echo (ping) request id=0x0003, seq=2/512, ttl=64 (reply in 14725)
14725	1347.869708	10.23.169.1	10.23.169.137	ICMP	134	Echo (ping) reply id=0x0003, seq=2/512, ttl=64 (request in 14724)
14726	1348.882783	10.23.169.137	10.23.169.1	ICMP	134	Echo (ping) request id=0x0003, seq=3/768, ttl=64 (reply in 14727)
> Frame 14728: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface \Device\NPF_{36e...}						
> Ethernet II, Src: VMware_25:83:bb (00:50:56:25:83:bb), Dst: VMware_21:35:ca (00:50:56:21:35:ca)						
> Internet Protocol Version 4, Src: 192.168.190.129, Dst: 192.168.190.128						
> Ethernet II, Src: VMware_38:60:9c (00:50:56:38:60:9c), Dst: VMware_da:79:18 (00:0c:29:da:79:18)						
> Internet Protocol Version 4, Src: 10.23.169.137, Dst: 10.23.169.2						
> Internet Control Message Protocol						

Figura 4.3: Parte de la trama capturada con wireshark perteneciente a la cabecera EOIP

```

> Frame 69: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface \Device\NPF_{36e...}
> Ethernet II, Src: VMware_25:83:bb (00:50:56:25:83:bb), Dst: VMware_21:35:ca (00:50:56:21:35:ca)
> Internet Protocol Version 4, Src: 192.168.190.129, Dst: 192.168.190.128
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 120
    Identification: 0xed28 (60712)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: Ether in IP (97)
    Header Checksum: 0x8ea9 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.190.129
    Destination Address: 192.168.190.128
> Ethernet II, Src: VMware_38:60:9c (00:50:56:38:60:9c), Dst: VMware_da:79:18 (00:0c:29:da:79:18)
> Internet Protocol Version 4, Src: 10.23.169.137, Dst: 10.23.169.2
<

```

Figura 4.4: Cabecera del protocolo IP exterior a la trama EOIP

```

> Ethernet II, Src: VMware_25:83:bb (00:50:56:25:83:bb), Dst: VMware_21:35:ca (00:50:56:21:35:ca)
> Internet Protocol Version 4, Src: 192.168.190.129, Dst: 192.168.190.128
> Ethernet II, Src: VMware_38:60:9c (00:50:56:38:60:9c), Dst: VMware_da:79:18 (00:0c:29:da:79:18)
> Internet Protocol Version 4, Src: 10.23.169.137, Dst: 10.23.169.2
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x3b9e (15262)
  > 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: ICMP (1)
    Header Checksum: 0x9851 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.23.169.137
    Destination Address: 10.23.169.2
> Internet Control Message Protocol

```

Figura 4.5: Cabecera del protocolo IP interior a la trama EOIP

CAPÍTULO 5

Conclusiones

5.0.1. Ampliación grupo 3 alumnos

En nuestro caso, hemos optado por transmitir un fichero lo suficientemente grande (2,2GB) como para poder medir el tiempo de subida en MB/s y así comprobar si es más eficiente en términos de tiempo el uso del túnel o por lo contrario, el envío sin túnel.

Como podemos observar en la Figura 5.1, cuando usamos el acceso directo en vez del túnel obtenemos una velocidad de subida de 24,6 MB/s mientras que cuando usamos el acceso a través del túnel EoIP obtenemos una velocidad de subida menor de 23 MB/s. Esta **diferencia de 1,6 MB** es debida al procesamiento adicional necesario para enviar datos a través del túnel y luego desencapsularlos en el extremo receptor. En otras palabras, la diferencia es debida al añadido de cabeceras complementarias en el túnel EoIP lo cual es una **forma de sobrecarga**.

```
[root@rco-x Descargas]# scp Windows-7SinLicencia.zip root@192.168.190.130:/home
root@192.168.190.130's password:
Windows-7SinLicencia.zip                                100% 2240MB  24.6MB/s   01:31
[root@rco-x Descargas]# scp Windows-7SinLicencia.zip root@10.23.169.2:/home
The authenticity of host '10.23.169.2 (10.23.169.2)' can't be established.
ECDSA key fingerprint is SHA256:jfyZALWwz2RdcHCCQYnb/79IHm//rAk3QzvZ+PlKVW4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.23.169.2' (ECDSA) to the list of known hosts.
root@10.23.169.2's password:
Windows-7SinLicencia.zip                                100% 2240MB  23.0MB/s   01:37
[root@rco-x Descargas]#
```

Figura 5.1: Captura tiempos transmisión fichero

Podemos observar en la figura 5.2 como la trama se guía a través de la red vmnet1 , donde aún no incluye la cabecera EOIP

No.	Time	Source	Destination	Protocol	Length	Info
98	211.539561	10.23.169.2	10.23.169.137	TCP	60	90 Echo (ping) reply Seq=820480, ttl=64 (request in 90)
99	214.540039	10.23.169.137	10.23.169.2	TCP	60	90 Echo (ping) request Seq=92384, ttl=64 (reply in 92)
100	215.540510	10.23.169.137	10.23.169.2	TCP	60	90 Echo (ping) request Seq=92384, ttl=64 (request in 91)
101	215.544100	10.23.169.2	10.23.169.137	TCP	60	90 Echo (ping) request Seq=102560, ttl=64 (reply in 94)
102	216.007095	10.23.169.2	10.23.169.137	TCP	60	90 Echo (ping) request Seq=102560, ttl=64 (request in 93)
103	216.542252	10.23.169.137	10.23.169.2	TCP	60	90 Echo (ping) request Seq=112816, ttl=64 (reply in 97)
104	216.542496	10.23.169.2	10.23.169.137	TCP	60	90 Echo (ping) request Seq=112816, ttl=64 (request in 98)
105	217.546680	10.23.169.137	10.23.169.2	TCP	60	90 Echo (ping) request Seq=123072, ttl=64 (reply in 99)
106	217.547267	10.23.169.2	10.23.169.137	TCP	60	90 Echo (ping) request Seq=123072, ttl=64 (request in 98)
107	217.140372	10.23.169.2	178.215.228.24	UDP	90	90 NTP Version 4, client
108	217.177876	178.215.228.24	10.23.169.2	UDP	90	90 NTP Version 4, server
109	240.000480	10.23.169.2	239.192.152.143	LSO	170	
110	242.000480	10.23.169.2	239.192.152.143	LSO	170	
111	242.140543	10.23.169.2	239.192.152.143	ARP	60	Who has 10.23.169.2? Tell 10.23.169.2

Frame 100: 212 bytes on wire (1696 bits), 212 bytes captured (1696 bits) on interface Vmnet1 (DPI) Ethernet II, Src: VMware, 08:00:27:12:34:56, Dst: VMware, 08:00:27:12:34:56

Type: IPv4 (0x0008)

Internet Protocol Version 4, Src: 10.23.169.2, Dst: 239.255.255.250

0x00 ... = Version: 4

... 0x00 ... = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 100

Identification: 0x57af (22447)

0x00 ... = Flags: 0x0

... 0x0000 0000 ... = Fragment Offset: 0

Time to Live: 1

Protocol: UDP (17)

Header Checksum: 0xb8e3 (validation disabled)

Header checksum status: Unverified

Source Address: 10.23.169.2

Destination Address: 239.255.255.250

User Datagram Protocol, Src Port: 52276, Dst Port: 1900

Simple Sequence Discovery Protocol

Figura 5.2: Captura del paso de los datos por la red vmnet1

Para luego atravesar el túnel en la VMnet 8, ya con las cabeceras necesarias para ejecutar correctamente el EOIP. Se muestra en la figura 5.3 que se muestra a continuación.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.23.169.137	10.23.169.2	ICMP	134	Echo (ping) request: id=0x0000, seq=102/26112, ttl=64 (reply in 2)
2	0.000000	10.23.169.2	10.23.169.137	ICMP	134	Echo (ping) reply: id=0x0000, seq=102/26112, ttl=64 (request in 1)
3	1.002106	10.23.169.137	10.23.169.2	ICMP	134	Echo (ping) request: id=0x0000, seq=103/26168, ttl=64 (reply in 6)
4	1.002019	10.23.169.2	10.23.169.137	ICMP	134	Echo (ping) reply: id=0x0000, seq=103/26168, ttl=64 (request in 3)
5	2.003644	10.23.169.137	10.23.169.2	ICMP	134	Echo (ping) request: id=0x0000, seq=104/26624, ttl=64 (reply in 8)
6	2.004524	10.23.169.2	10.23.169.137	ICMP	134	Echo (ping) reply: id=0x0000, seq=104/26624, ttl=64 (request in 5)
7	3.008869	10.23.169.137	10.23.169.2	ICMP	134	Echo (ping) request: id=0x0000, seq=105/26880, ttl=64 (reply in 12)
8	3.008869	10.23.169.2	10.23.169.137	ICMP	134	Echo (ping) reply: id=0x0000, seq=105/26880, ttl=64 (request in 11)
9	4.007297	10.23.169.137	10.23.169.2	ICMP	134	Echo (ping) request: id=0x0000, seq=106/27136, ttl=64 (reply in 14)
10	4.007362	10.23.169.2	10.23.169.137	ICMP	134	Echo (ping) reply: id=0x0000, seq=106/27136, ttl=64 (request in 13)
11	5.008304	10.23.169.137	10.23.169.2	ICMP	134	Echo (ping) request: id=0x0000, seq=107/27392, ttl=64 (reply in 17)
12	5.009576	10.23.169.2	10.23.169.137	ICMP	134	Echo (ping) reply: id=0x0000, seq=107/27392, ttl=64 (request in 16)
13	6.011109	10.23.169.137	10.23.169.2	ICMP	134	Echo (ping) request: id=0x0000, seq=108/27648, ttl=64 (reply in 20)
14	6.012539	10.23.169.2	10.23.169.137	ICMP	134	Echo (ping) reply: id=0x0000, seq=108/27648, ttl=64 (request in 19)
15	7.013569	10.23.169.137	10.23.169.2	ICMP	134	Echo (ping) request: id=0x0000, seq=109/27904, ttl=64 (reply in 23)

Figura 5.3: captura de la trama desde rco-X a rco-noX pasando por el túnel

Este último paso añade peso a la ejecución, lo que provoca la diferencia en tiempos y velocidades entre el uso del túnel y su no uso.

5.0.2. Conclusiones finales

A lo largo de este trabajo hemos aprendido los diferentes aspectos de la utilización de túneles EOIP en redes locales pequeñas, su potencial uso y sus posibles configuraciones. Hemos a su vez entendido que, para **grandes redes**, mucho más complejas puede ser altamente ineficiente debido al gran nivel de overhead que se añadirían a los paquetes, siendo otras soluciones mucho más acertadas.

Creemos que entre los objetivos cumplidos caben destacar:

- Utilización y configuración de máquinas virtuales VMware
- Configuración de routers dd-wrt
- Detección y filtrado de paquetes mediante Wireshark
- Conocimiento en profundidad de los túneles EOIP y las cabeceras implicadas

Bibliografía

- [1] Julio Pons. Práctica/trabajo 1: Túnel eoip con routers dd-wrt, 2023. Disponible en https://poliformat.upv.es/access/content/group/GRA_11609_2023/_2_%20PR%C3%81CTICAS/Pr%C3%A1ctica-Trabajo%201/_S1__2023_%20Seminario%20EoIP.pdf, consultada el 19/10/2023].
- [2] RFC. Generic routing encapsulation (gre), 2000. Disponible en <https://www.rfc-editor.org/rfc/rfc2784.html>, consultada el 21/10/2023].
- [3] Wikipedia. Modo promiscuo, -. Disponible en https://es.wikipedia.org/wiki/Modo_promiscuo#Modo_promiscuo_y_redes_wi-fi, consultada el 23/10/2023].
- [4] Ahmad Purwana. Analysis of ethernet over internet protocol (eoip) vpn performance. Technical report, 2021. Disponible en <https://jcsitech-upiyptk.org/ojs/index.php/jcsitech/article/view/11/15>, consultada el 19/10/2023.
- [5] MikroTik. Eoip -routers, 2020. Disponible en <https://help.mikrotik.com/docs/display/ROS/EoIP>, consultada el 15/10/2023].
- [6] Thandar Thein Si Thu Aung. Comparative analysis of site-to-site layer 2 virtual private networks. Technical report, 2020. Disponible en <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9022848>, consultada el 21/10/2023.