



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

## **Trabajo 2: PPTP**

**TRABAJO RCO**

Grado en Ingeniería Informática

*Autor:* Ismael Fernández Herreruela  
Rafael Belenguer Arcas  
Francisco Ramos Guardiola  
*Grupo:* 161

Curso 2024-2025



# Resumen

Este estudio analiza en profundidad el comportamiento del túnel PPTP en dos escenarios principales: conexión site-to-site y acceso remoto. En el caso del modo site-to-site, se conecta de forma directa dos routers (ddwrt-noX y ddwrt-X), permitiendo la integración entre redes corporativa. Por su parte, el modo remote-access transforma al servidor PPTP en un punto centralizado que asegura el acceso de usuarios remotos a la red. Se llevan a cabo pruebas operativas, se examinan los paquetes encapsulados por PPTP y se revisan las tablas de enrutamiento. También se explica cómo se configura el túnel, se mantiene la conexión y se gestionan los dispositivos y las tablas de reenvío.

**Palabras clave:** PPTP, túnel, site-to-site, remote-access, encapsulación, enrutamiento, remoto

---

# Abstract

This study analyzes in depth the behavior of the PPTP tunnel in two scenarios. main parameters: site-to-site connection and remote access. In the case of site-to-site mode, directly connects two routers (ddwrt-noX and ddwrt-X), allowing integration between corporate networks. For its part, remote-access mode transforms the server PPTP at a centralized point that ensures remote user access to the network. HE operational tests are carried out, packets encapsulated by PPTP are examined and routing tables are reviewed. It also explains how to configure the tunnel, how the connection is maintained and how devices and forwarding tables are managed.

**Key words:** PPTP, tunnel, site-to-site, remote-access, encapsulation, routing, remote

---



# Índice general

<b>Índice general</b>	<b>V</b>
<b>Índice de figuras</b>	<b>VII</b>
<b>Índice de tablas</b>	<b>VIII</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Objetivos . . . . .	1
<b>2 Configuración</b>	<b>3</b>
2.1 Desmontaje del tunel EoIP . . . . .	3
2.2 Cambio de la IP de red local de ddwrt-X . . . . .	4
2.3 Desconexión de red NAT en RCO-noX . . . . .	5
2.4 Configuración de PPTP site-to-site . . . . .	6
2.4.1 Configuración ddwrt-X (Servidor PPTP) . . . . .	6
2.4.2 Configuración ddwrt-noX (Cliente PPTP) . . . . .	8
2.4.3 Verificación de la configuración del tunel PPTP site-to-site . . . . .	10
2.5 Configuración para remote-access . . . . .	11
2.5.1 Desactivación de la regla de enrutamiento PC anfitrión . . . . .	11
2.5.2 Desactivación cliente PPTP router ddwrt-noX . . . . .	12
2.5.3 Creación VPN PPTP en PC anfitrión . . . . .	12
2.5.4 Verificación de la configuración del túnel PPTP remote-access . . . . .	15
<b>3 Funcionamiento del túnel PPTP site-to-site</b>	<b>19</b>
3.1 Explicación teórica . . . . .	19
3.2 Fases en PPP . . . . .	19
3.3 Prueba 1 pings entre RCO-noX y RCO-X . . . . .	20
3.4 Prueba 2 pings de anfitrión a RCO-X no funcionan . . . . .	25
3.5 Prueba 3 tracert de PC anfitrión a RCO-X . . . . .	27
3.6 Prueba 4 Comprobación de conexión entre las máquinas RCO . . . . .	28
3.7 Prueba 5 Tablas de routing . . . . .	30
<b>4 Funcionamiento del túnel PPTP remote-access</b>	<b>33</b>
4.1 Prueba 1. Pings entre PC anfitrión y RCO-X . . . . .	33
4.2 Prueba 2. Análisis del datagrama original . . . . .	35
4.3 Prueba 4. Análisis de los dispositivos creados por el túnel en el cliente . . . . .	36
4.4 Prueba 5. Estudio de las tablas de forwarding . . . . .	37
4.5 Prueba 6. Comparación de prestaciones y redirección de puertos . . . . .	38
4.5.1 Comparación de prestaciones . . . . .	38
4.5.2 Redirección de Puertos . . . . .	44
4.5.3 Comparación prestaciones con y sin redirección . . . . .	47
<b>5 Conclusiones</b>	<b>49</b>
<b>Bibliografía</b>	<b>51</b>



# Índice de figuras

---

2.1	Configuración inicial . . . . .	3
2.2	Deshabilitación túnel EoIP ddwrt-noX . . . . .	4
2.3	Deshabilitación túnel EoIP ddwrt-X . . . . .	4
2.4	Cambios en la IP de ddwrt-X . . . . .	5
2.5	Cambios a realizar en RCO-noX . . . . .	5
2.6	ifconfig RCO-noX . . . . .	6
2.7	iproute RCO-noX . . . . .	6
2.8	Ping a google.es desde RCO-noX . . . . .	6
2.9	Configuración PPTP ddwrt-X . . . . .	7
2.10	Redirección puerto 1723 . . . . .	7
2.11	Configuración nuevo routing ddwrt-X . . . . .	8
2.12	Configuración PPTP Cliente router ddwrt-noX . . . . .	9
2.13	PPTP Passthrough activado . . . . .	9
2.14	ifconfig ddwrt-X . . . . .	10
2.15	iproute ddwrt-X . . . . .	10
2.16	ifconfig ddwrt-noX . . . . .	10
2.17	iproute ddwrt-noX . . . . .	11
2.18	iproute ddwrt-noX . . . . .	11
2.19	Eliminación regla de enrutamiento . . . . .	11
2.20	Desactivación cliente PPTP . . . . .	12
2.21	Configuración Windows . . . . .	12
2.22	Ajustes VPN PPTP . . . . .	13
2.23	Ajustes VPN PPTP . . . . .	14
2.24	Ajustes VPN PPTP . . . . .	15
2.25	Ajustes VPN PPTP . . . . .	15
2.26	route print PC anfitrión . . . . .	16
2.27	interfaz PC anfitrión . . . . .	16
2.28	interfaces ddwrt-X . . . . .	17
2.29	tablas routing ddwrt-X . . . . .	17
2.30	tablas routing ddwrt-X . . . . .	17
3.1	Ping desde RCO-noX a RCO-X en site-to-site . . . . .	20
3.2	Paquete ICMP en VMnet1 . . . . .	21
3.3	Paquete LCP en VMnet8 . . . . .	22
3.4	Paquete PPP Comp en VMnet8 . . . . .	23
3.5	Paquete GRE modificado en VMnet8 . . . . .	24
3.6	Primeros paquetes al realizar el ping . . . . .	24
3.7	VMnet1 y VMnet8 capturadas de forma simultánea al realizar el ping . . . . .	25
3.8	Ping entre PC anfitrión y RCO-X . . . . .	25
3.9	Comando de modificación de las tablas de enrutamiento . . . . .	26
3.10	Comando de imprimir las tablas de enrutamiento, lista de interfaces . . . . .	26
3.11	Comando de imprimir las tablas de enrutamiento, tabla de enrutamiento . . . . .	26
3.12	Ping correcto entre PC anfitrión y RCO-X . . . . .	27

3.13	Tracert entre PC anfitrión y RCO-X pre regla . . . . .	27
3.14	Tracert entre PC anfitrión y RCO-X . . . . .	28
3.15	Instalación de traceroute en RCO-noX . . . . .	28
3.16	Instalación de traceroute en RCO-X . . . . .	29
3.17	Uso de traceroute en RCO-noX . . . . .	29
3.18	Uso de traceroute en RCO-X . . . . .	29
3.19	Traceroute en Wireshark . . . . .	30
3.20	Traceroute en Wireshark: cabeceras . . . . .	30
3.21	Tabla de ruta de DDWRT-noX . . . . .	31
3.22	Tabla de ruta de DDWRT-X . . . . .	31
3.23	Tabla de ruta de RCO-noX . . . . .	32
3.24	Tabla de ruta de RCO-X . . . . .	32
4.1	Ping correcto entre PC anfitrión y RCO-X . . . . .	34
4.2	Captura Wireshark VMnet8 . . . . .	34
4.3	Informacion detallada Datagrama Original . . . . .	35
4.4	Captura Wiresharck Señalando Echo Request extra . . . . .	36
4.5	ipconfig en el PC anfitrión . . . . .	36
4.6	routeprint en el PC anfitrión . . . . .	37
4.7	iproute en ddwrt . . . . .	38
4.8	iproute en RCO-X . . . . .	38
4.9	tracert desde PC anfitrión a RCO-X . . . . .	38
4.10	Activación de la encriptación del túnel desde la interfaz gráfica del router ddwrt-X . . . . .	39
4.11	Propiedades VPN para permitir conexión encriptada . . . . .	40
4.12	Ping desde PC anfitrión a RCO-X sin encriptar . . . . .	41
4.13	Ping desde PC anfitrión a RCO-X encriptado . . . . .	41
4.14	iperf3 desde RCO-X sin encriptar . . . . .	42
4.15	iperf3 desde PC anfitrión sin encriptar . . . . .	42
4.16	iperf3 desde RCO-X encriptado . . . . .	43
4.17	iperf3 desde PC anfitrión encriptado . . . . .	43
4.18	Redirección en la Interfaz Gráfica de ddwrt-X . . . . .	44
4.19	Ping desde PC anfitrión a RCO-X usando el puerto 22222 . . . . .	44
4.20	Ping desde PC anfitrión a RCO-X sin encriptar . . . . .	45
4.21	Ping desde PC anfitrión a RCO-X encriptado . . . . .	45
4.22	iperf3 desde RCO-X sin encriptar . . . . .	46
4.23	iperf3 desde PC anfitrión sin encriptar . . . . .	46
4.24	iperf3 desde RCO-X encriptado . . . . .	47
4.25	iperf3 desde PC anfitrión encriptado . . . . .	47

## Índice de tablas

---



---

# CAPÍTULO 1

## Introducción

---

El objetivo principal de este trabajo es analizar el protocolo de tunelización PPTP en modo remoto y acceso remoto. En el modo de sitio a sitio, se establece una conexión entre dos enrutadores (ddwrt-noX y ddwrt-X) permitiendo una comunicación eficiente entre redes corporativas. Por otro lado, el modo de acceso remoto utiliza un servidor PPTP como concentrador, lo que facilita conexiones seguras para los usuarios remotos. Además, se proporciona un diagrama de red virtual detallado, que incluye todas las direcciones IP relevantes.

### 1.1 Objetivos

---

1. Analizar y entender el funcionamiento del túnel PPTP en los modos site-to-site y remote-access.
2. Realizar pruebas de funcionamiento y capturas de paquetes para validar la configuración del túnel.
3. Examinar las tablas de enrutamiento de los dispositivos involucrados.
4. Ofrecer una perspectiva general sobre las ventajas y desventajas del protocolo PPTP, así como su relevancia en el contexto actual.



---

## CAPÍTULO 2

# Configuración

---

Partiremos de la configuración realizada en el Trabajo 1 la cual podemos observar en la figura 2.1 y haremos cambios a partir de ella.

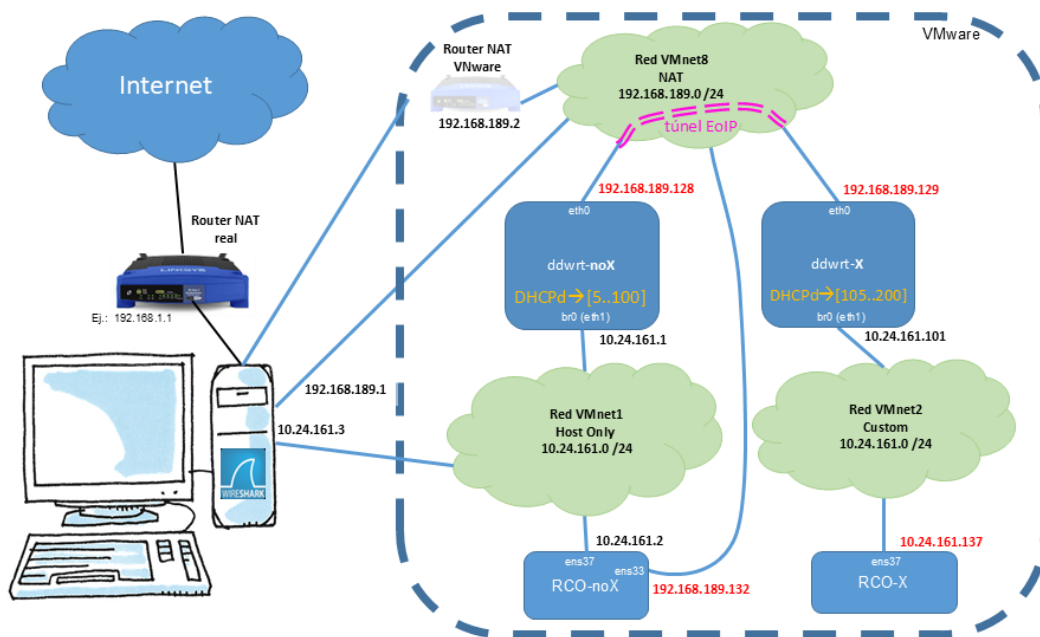


Figura 2.1: Configuración inicial

### 2.1 Desmontaje del tunel EoIP

---

Para comenzar a desmontar el tunel primero debemos acceder a ambos routers a través de sus respectivas IPs ('192.168.161.133' para noX en la figura 2.2 Y '192.168.161.134' para X en la figura 2.3). Una vez dentro de los routers debemos de seguir los siguientes pasos:

1. Click en Setup

2. Buscamos EoIP Tunnel
3. Le damos a disable, save y apply settings.

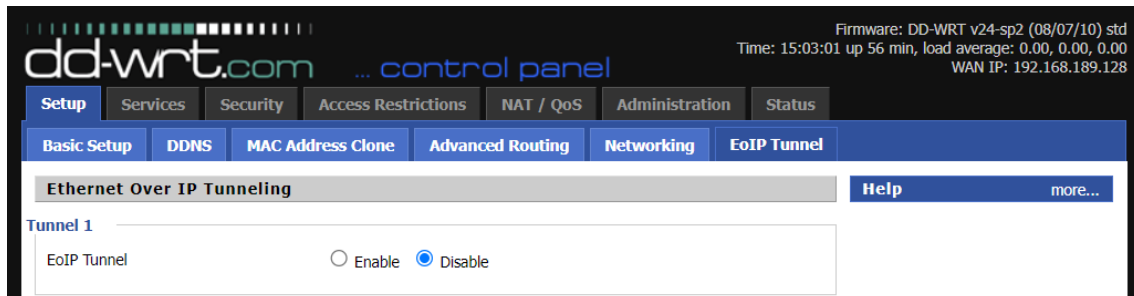


Figura 2.2: Deshabilitación túnel EoIP ddwrt-noX

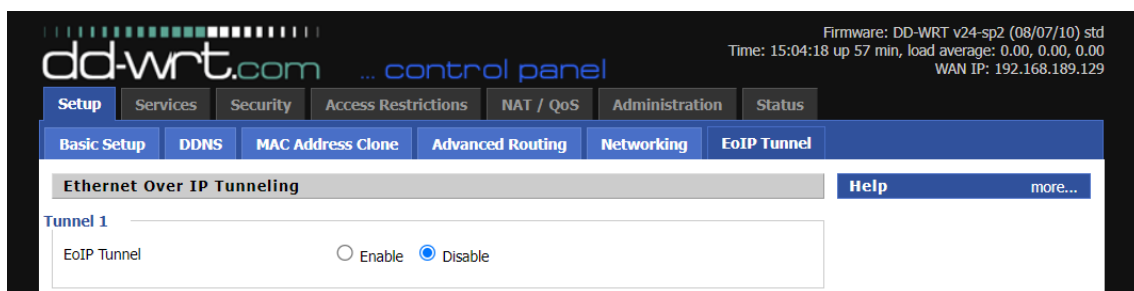


Figura 2.3: Deshabilitación túnel EoIP ddwrt-X

## 2.2 Cambio de la IP de red local de ddwrt-X

Dado que la red Vmnet2 tiene IP "10.54.xxx.xxx" hay que hacer un ajuste. Para conseguir esto hay que hacer los siguientes pasos:

1. Entrar en la IP del router
2. Click en Setup
3. Click en Basic Setup
4. Buscamos Local IP Address
5. Cambiar los parámetros por los de la figura 2.4.

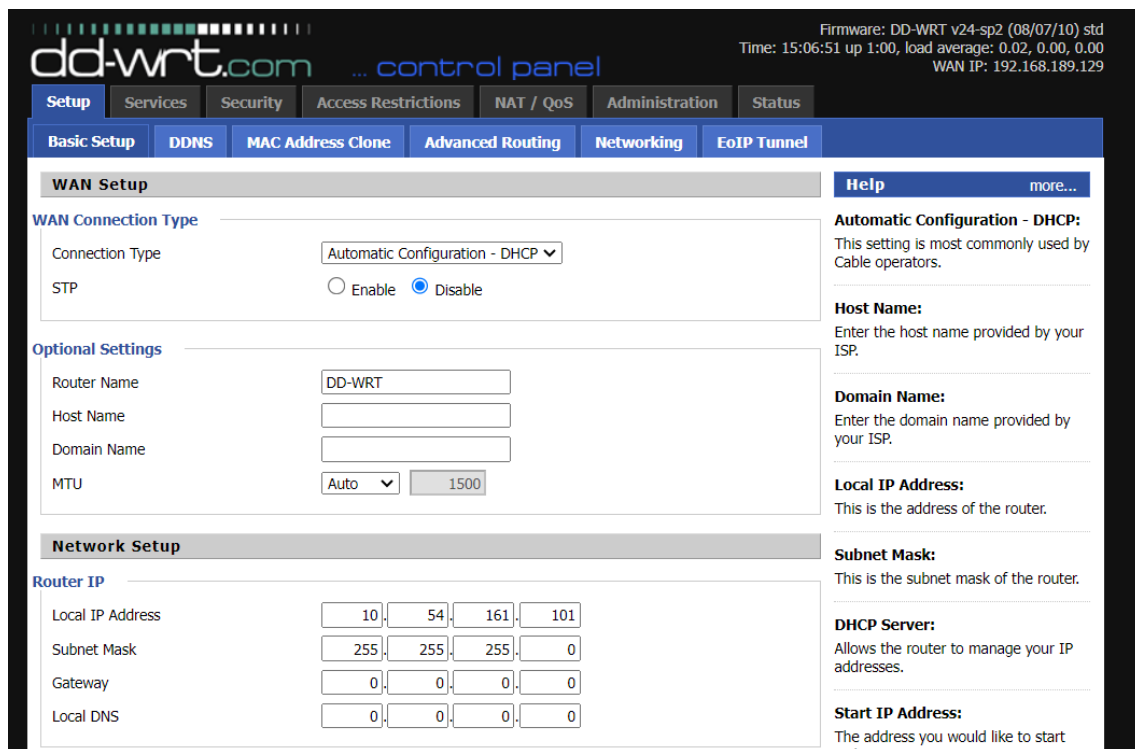


Figura 2.4: Cambios en la IP de ddwrt-X

## 2.3 Desconexión de red NAT en RCO-noX

En este punto, en RCO-noX solo necesitamos quedarnos con la red VMnet1, por lo que vamos a proceder a desconectarla con los siguientes pasos(figura 2.5):

1. Ir a los ajustes de la VM
2. Click en Network Adapter
3. Deseleccionamos las casillas 'Connected' y 'Connect at power on'

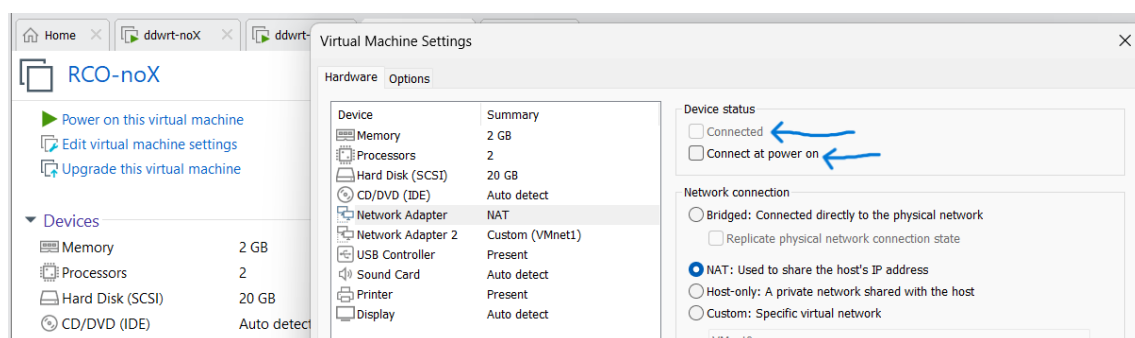


Figura 2.5: Cambios a realizar en RCO-noX

Después de esto, podemos comprobar en la figura 2.6 y la figura 2.7 los resultados obtenidos.

```
[root@rco-nox ~]# ifconfig
ens33: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 00:0c:29:69:62:cf txqueuelen 1000 (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 0 overruns 0 carrier 0 collisions 0

ens37: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.24.161.2 netmask 255.255.255.0 broadcast 10.24.161.255
```

Figura 2.6: ifconfig RCO-noX

```
[root@rco-nox ~]# ip route show
default via 10.24.161.1 dev ens37 proto static metric 100
10.24.161.0/24 dev ens37 proto kernel scope link src 10.24.161.2 metric 100
```

Figura 2.7: iproute RCO-noX

En la figura 2.8 podemos ver el correcto funcionamiento después de realizar la desconexión de la NAT.

```
[root@rco-nox ~]# ping google.es
PING google.es (142.250.200.131) 56(84) bytes of data.
64 bytes from mad41s14-in-f3.1e100.net (142.250.200.131): icmp_seq=1 ttl=127 time=18.8 ms
64 bytes from mad41s14-in-f3.1e100.net (142.250.200.131): icmp_seq=2 ttl=127 time=23.5 ms
64 bytes from mad41s14-in-f3.1e100.net (142.250.200.131): icmp_seq=3 ttl=127 time=19.5 ms
64 bytes from mad41s14-in-f3.1e100.net (142.250.200.131): icmp_seq=4 ttl=127 time=22.9 ms
```

Figura 2.8: Ping a google.es desde RCO-noX

## 2.4 Configuración de PPTP site-to-site

### 2.4.1. Configuración ddwrt-X (Servidor PPTP)

En primer lugar, se deshabilita el cifrado MPPE para que cuando se realicen las pruebas correspondientes se pueda ver la información en un formato claro. Acto seguido, en 'Server IP' tenemos que poner como parámetro la red local interna (10.54.161.101) y en 'Client IP' hay que poner el rango 10.54.161.201-209. A continuación en el apartado 'CHAP-Secrets' hay que poner 'admin \* admin \*' incluyendo los espacios y los asteriscos. Se puede ver toda esta configuración de manera mas visual en la figura 2.9.

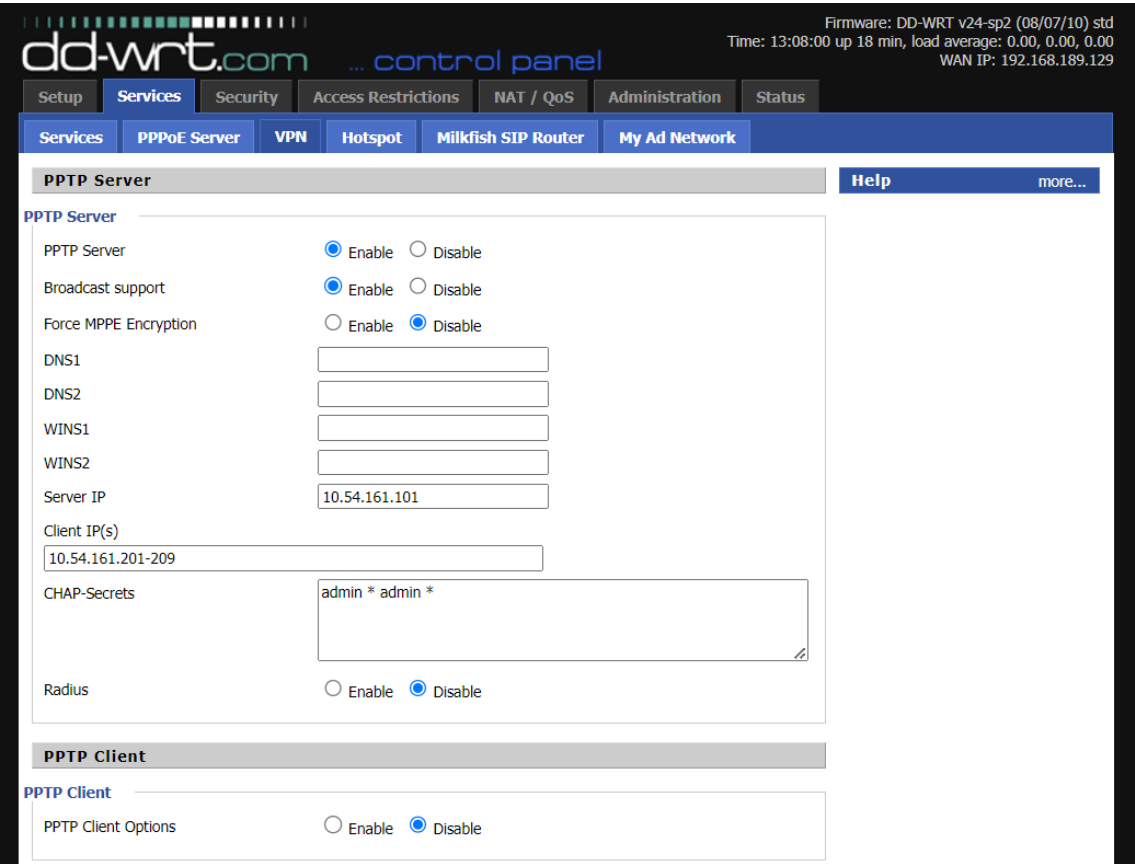


Figura 2.9: Configuración PPTP ddwrt-X

Dado que el protocolo PPTP utiliza el puerto TCP 1723, configuraremos una redirección de este puerto público hacia el servidor PPTP. Aunque esto podría no ser estrictamente necesario ya que el servidor PPTP está integrado en el propio router. De todos modos es recomendable hacerlo porque el servidor está asociado a una dirección IP local. Esta configuración se realiza accediendo a la pestaña “NAT/QoS” y luego a “Port Forwarding”. Esta configuración la podemos observar en la figura 2.10.

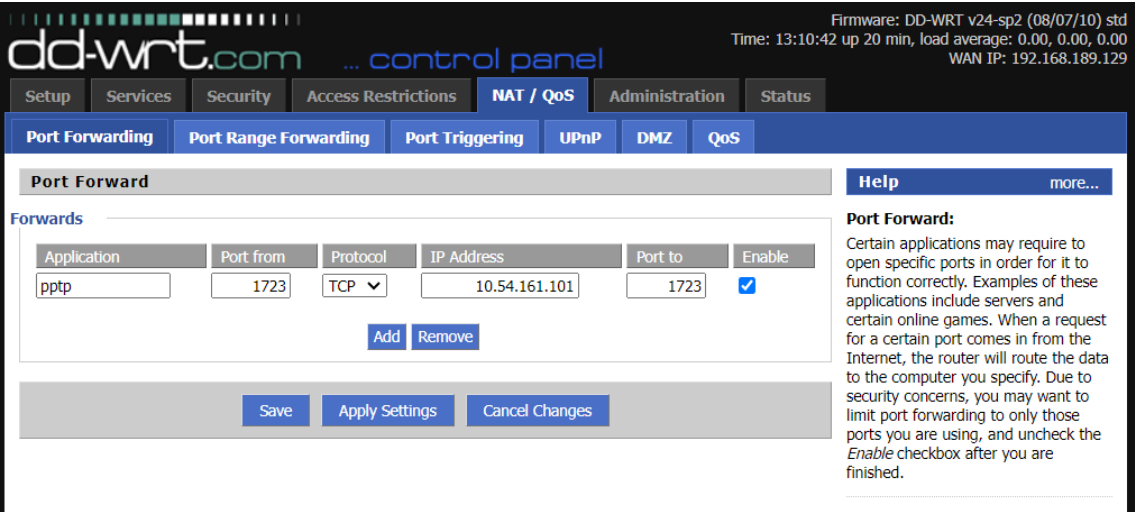


Figura 2.10: Redirección puerto 1723

Para finalizar la configuración site-to-site, es necesario indicarle a nuestro router cómo manejar los datagramas dirigidos a la red '10.24.x.0/24'. Esto le permitirá saber que debe enviarlos a través del otro extremo del túnel. Para lograrlo, añadimos una nueva regla en la tabla de enrutamiento.

Podemos realizar este paso desde la interfaz gráfica:

1. Click en Setup
2. Click en Advanced Routing

Dejamos el Operating Mode configurado como 'Gateway', asignamos un nombre personalizado en 'Route Name', y configuramos la 'Metric' en 0. Como red de destino (LAN), ingresamos '10.23.161.0' con máscara '255.255.255.0' y establecemos el Gateway en '10.23.161.1'. Por último, es crucial seleccionar la interfaz como 'ANY'.

Esta configuración se representa de manera más clara en la figura 2.11.

The screenshot shows the dd-wrt control panel interface. At the top, it says 'dd-wrt.com ... control panel' and 'Firmware: DD-WRT v24-sp2 (08/07/10) std'. Below this are tabs for Setup, Services, Security, Access Restrictions, NAT / QoS, Administration, and Status. The 'Advanced Routing' tab is selected. Under 'Operating Mode', 'Gateway' is selected. Under 'Static Routing', a new route is being added with set number 1, route name 'a-ddwrt-noX-via-PPTP', metric 0, destination LAN NET 10.24.161.0/255.255.255.0, gateway 10.24.161.1, and interface ANY. A 'Show Routing Table' button is visible. On the right, there is a 'Help' section with more information about the Operating Mode and other fields.

**Figura 2.11:** Configuración nuevo routing ddwrt-X

## 2.4.2. Configuración ddwrt-noX (Cliente PPTP)

Debemos de configurar este router para que se conecte como Cliente PPTP al Servidor PPTP, por lo que su configuración sería la siguiente:

1. Entrar a la IP del router
2. Ir al apartado 'Services'
3. Dentro de 'Services' hay que entrar en 'VPN'
4. Cambiar a 'Enable' el 'PPTP Client'
5. Una vez activado el cliente PPTP ponemos su IP pública, en nuestro caso es la mencionada anteriormente '192.168.161.129'



Hay que aclarar que en estos momentos el router ya sabe que su conexión va a ser del tipo site-to-site y en este caso hay que aclarar cual es la subred a la que se va a conectar. Para conseguir esto, lo que haremos será poner en 'Remote Subnet' la dirección que comentábamos en el anterior router como 'Red de destino', en este caso es la '10.54.161.0'. Como 'Remote Subnet Mask' utilizaremos la '255.255.255.0'. Todas estas configuraciones se pueden apreciar en la figura 2.12.

The screenshot shows the 'dd-wrt.com ... control panel' interface. The top navigation bar includes 'Setup', 'Services', 'Security', 'Access Restrictions', 'NAT / QoS', 'Administration', and 'Status'. The 'Services' tab is active, and the 'VPN' sub-tab is selected. The 'PPTP Server' section has 'PPTP Server' disabled. The 'PPTP Client' section is expanded, showing 'PPTP Client Options' with 'Enable' selected. The 'Server IP or DNS Name' is '192.168.189.129'. The 'Remote Subnet' is '10.54.161.0' and the 'Remote Subnet Mask' is '255.255.255.0'. 'MPPE Encryption' is set to 'mppe required'. 'MTU' and 'MRU' are both '1450'. 'NAT' is disabled. 'User Name' is 'admin' and 'Password' is 'admin', with the 'Unmask' checkbox checked.

Figura 2.12: Configuración PPTP Cliente router ddwrt-noX

En último lugar, antes de realizar las verificaciones, hay que comprobar que el NAT sea compatible con clientes PPTP de la red local. Para hacer esto iremos al apartado 'Security' y dentro de este buscaremos 'VPN Passthrough'. Debemos de comprobar que la opción 'PPTP Passthrough' esté activada. Como se muestra en la figura 2.13

The screenshot shows the 'dd-wrt.com ... control panel' interface. The top navigation bar includes 'Setup', 'Services', 'Security', 'Access Restrictions', 'NAT / QoS', 'Administration', and 'Status'. The 'Security' tab is active, and the 'VPN Passthrough' sub-tab is selected. The 'Virtual Private Network (VPN)' section is expanded, showing 'VPN Passthrough' options. 'IPSec Passthrough', 'PPTP Passthrough', and 'L2TP Passthrough' are all enabled. A blue arrow points to the 'PPTP Passthrough' 'Enable' radio button. A help message on the right states: 'You may choose to enable IPSec, PPTP and/or L2TP passthrough to allow your network devices to communicate via VPN.' At the bottom, there are 'Save', 'Apply Settings', and 'Cancel Changes' buttons.

Figura 2.13: PPTP Passthrough activado

Tras todas estas configuraciones ya tendremos el servidor y el cliente configurados. Solo faltaria realizar un 'reboot' para que se establezca la conexión entre ambos. Para hacer esto habria que ir al apartado de 'Administration', dentro de este vamos a 'Management' y ahi le damos a 'Reboot Router' al fondo de la página. Primero el router y después el cliente. Si se tiene la suficiente confianza en el uso de la terminal, esto seria posible hacerlo de una manera mas rápida simplemente abriendo un terminal y escribiendo 'reboot'.

### 2.4.3. Verificación de la configuración del tunel PPTP site-to-site

Para saber que todo se ha configurado correctamente, debemos de comprobar que aparecen nuevas interfaces de red 'pppX' y nuevas entradas en las tablas de routing que estén vinculadas a estas interfaces de red. Podemos observarlo en las figuras 2.14, 2.15, 2.16 y 2.17.

```
root@DD-WRT:~# ifconfig ppp0
ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.54.161.101  P-t-P:10.24.161.1  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1442  Metric:1
          RX packets:10  errors:0  dropped:0  overruns:0  frame:0
          TX packets:10  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:3
          RX bytes:151 (151.0 B)  TX bytes:157 (157.0 B)
```

Figura 2.14: ifconfig ddwrt-X

```
root@DD-WRT:~# ip route
10.24.161.1 dev ppp0 proto kernel scope link src 10.54.161.101
192.168.189.2 dev eth0 scope link
10.24.161.0/24 via 10.24.161.1 dev ppp0
10.54.161.0/24 dev br0 proto kernel scope link src 10.54.161.101
192.168.189.0/24 dev eth0 proto kernel scope link src 192.168.189.129
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 192.168.189.2 dev eth0
```

Figura 2.15: iproute ddwrt-X

```
root@DD-WRT:~# ifconfig ppp0
ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.24.161.1  P-t-P:10.54.161.101  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1438  Metric:1
          RX packets:10  errors:0  dropped:0  overruns:0  frame:0
          TX packets:10  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:3
          RX bytes:157 (157.0 B)  TX bytes:151 (151.0 B)
```

Figura 2.16: ifconfig ddwrt-noX

```

root@DD-WRT:~# ip route
10.24.161.1 dev ppp0 proto kernel scope link src 10.54.161.101
192.168.189.2 dev eth0 scope link
10.24.161.0/24 via 10.24.161.1 dev ppp0
10.54.161.0/24 dev br0 proto kernel scope link src 10.54.161.101
192.168.189.0/24 dev eth0 proto kernel scope link src 192.168.189.129
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 192.168.189.2 dev eth0

```

Figura 2.17: iproute ddwrt-noX

Después de haber realizado la configuración requerida, el esquema de red resultante con el túnel site-to-site se muestra en la figura 2.18.

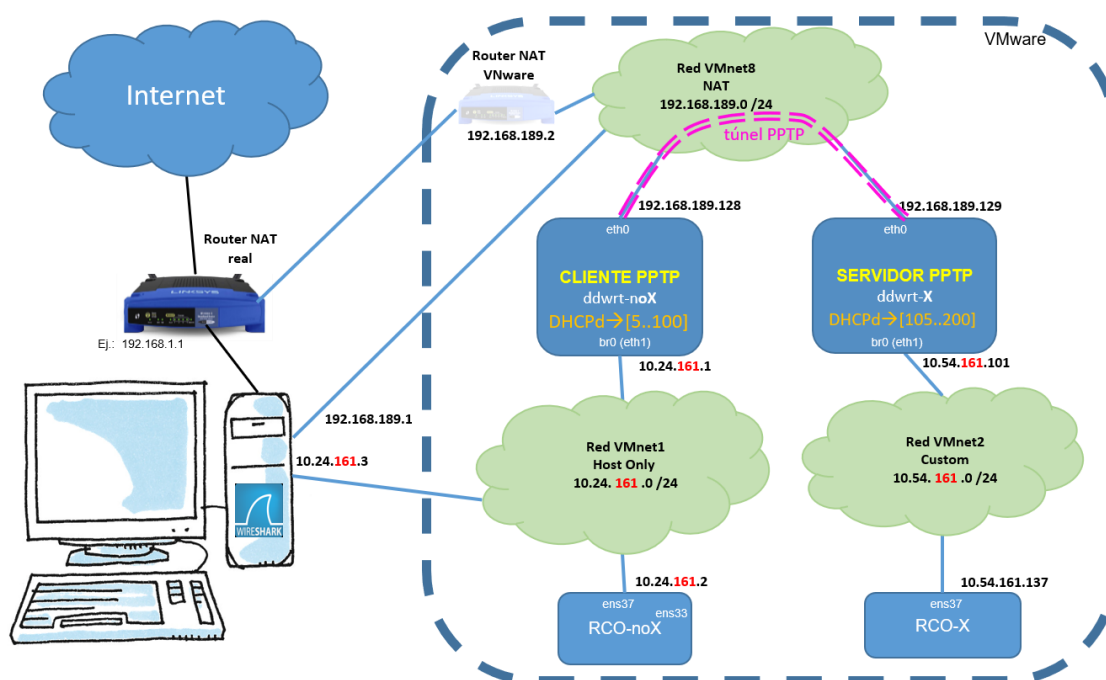


Figura 2.18: iproute ddwrt-noX

## 2.5 Configuración para remote-access

### 2.5.1. Desactivación de la regla de enrutamiento PC anfitrión

Dado que previamente habíamos establecido esta regla para que el tráfico se redirigiese hacia '10.54.161.0'. Esto se puede cambiar de una manera bastante sencilla abriendo una terminal en nuestro PC anfitrión y realizando el comando `route DELETE 10.54.161.0`. Sabremos que se ha ejecutado correctamente si vemos como respuesta en la terminal un 'Correcto' tal y como se muestra en la figura 2.19.

```

PS C:\Users\Work> route DELETE 10.54.161.0
Correcto

```

Figura 2.19: Eliminación regla de enrutamiento

### 2.5.2. Desactivación cliente PPTP router ddwrt-noX

Para hacer este cambio es tan sencillo como lo hemos activado antes, solo que en vez de activar y poner ciertos parámetros, en esta situación únicamente hay que darle al botón de 'Disable'. Para llegar a este botón se siguen los mismos pasos que antes, vamos a 'Services', dentro buscamos 'VPN' y ya dentro de VPN nos debería de salir la opción de desactivar el cliente PPTP tal y como se muestra en la figura 2.20.

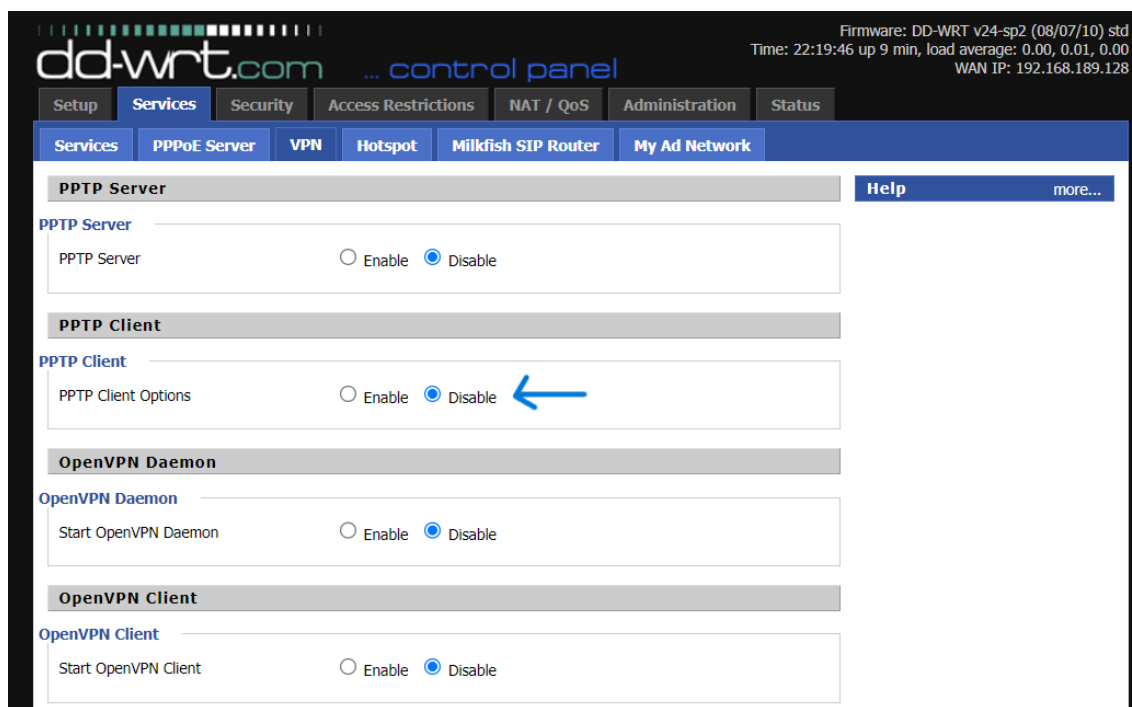


Figura 2.20: Desactivación cliente PPTP

Una vez cambiado, daremos al botón de 'Apply Settings' y procederemos a reiniciar el router. Se puede tanto desde la página web como desde la consola.

### 2.5.3. Creación VPN PPTP en PC anfitrión

A partir de aquí será necesaria la creación de una VPN que no utilice ni cifrado ni compresión para poder ver los resultados posteriormente con el Wireshark. Para hacer esto nos iremos al programa 'Configuración' de Windows 11 (figura 2.21). Una vez dentro nos iremos al apartado de 'Red e Internet' y aquí buscamos la opción de 'VPN'. Ahora será el momento de crear la nueva VPN, por lo que le daremos al botón 'Agregar VPN' y utilizaremos exactamente los ajustes de la figura 2.22 (la contraseña utilizada en la figura es 'admin').

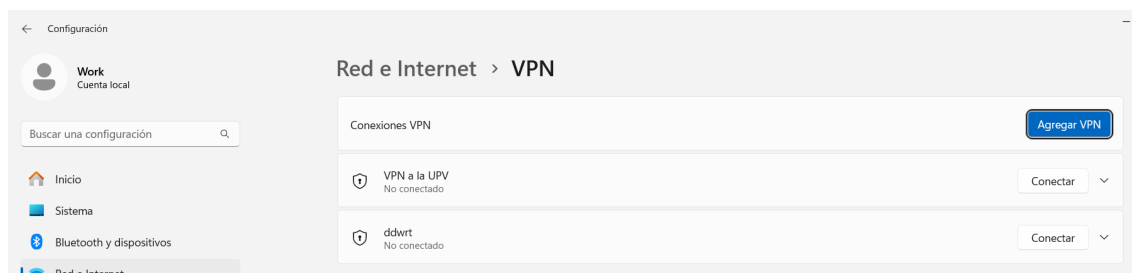
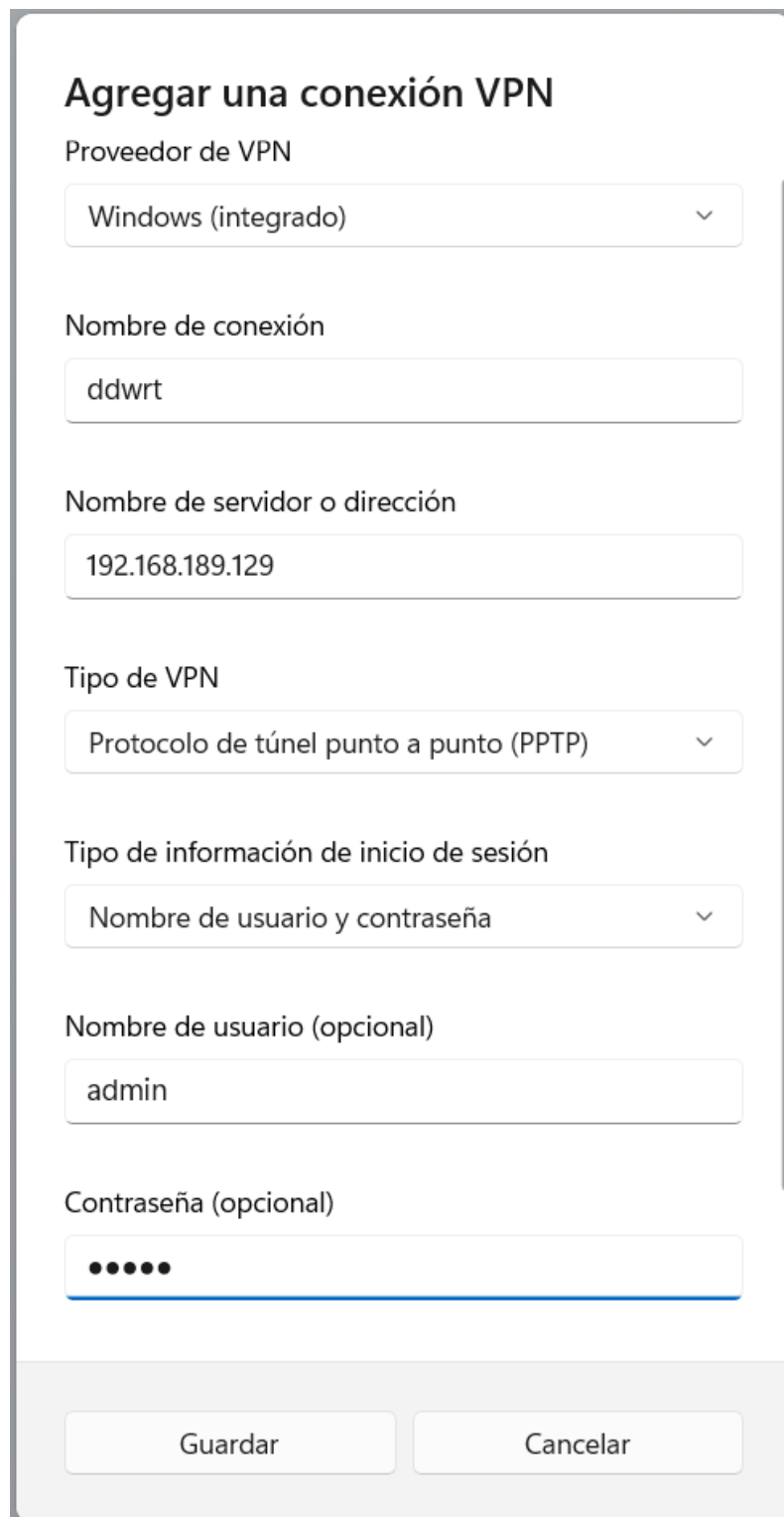


Figura 2.21: Configuración Windows



**Agregar una conexión VPN**

Proveedor de VPN

Windows (integrado) ▾

Nombre de conexión

ddwrt

Nombre de servidor o dirección

192.168.189.129

Tipo de VPN

Protocolo de túnel punto a punto (PPTP) ▾

Tipo de información de inicio de sesión

Nombre de usuario y contraseña ▾

Nombre de usuario (opcional)

admin

Contraseña (opcional)

•••••

Guardar Cancelar

**Figura 2.22:** Ajustes VPN PPTP

Tras realizar estos ajustes ya sería posible conectarnos al servidor, pero habría un problema, y se trata de que entraríamos en un bucle ya que todo el tráfico del PC iría por la VPN y nos quedaríamos sin internet porque la propia NAT usa nuestro PC anfitrión para salir a Internet. Para que esto no ocurra son necesarios unos ajustes que podremos realizar desde la pestaña de 'Propiedades' del router. Para llegar a esta pestaña debemos de ir al 'Panel de Control' y una vez dentro ir a 'Redes e internet' y acto seguido 'Conexión de red'. Ahí nos saldrán las conexiones, por lo que lo único que hara falta será

hacer click derecho en 'ddwrt' y darle a 'Propiedades'. A partir de aquí aplicaremos las configuraciones de las figuras 2.23, 2.24 y 2.25.

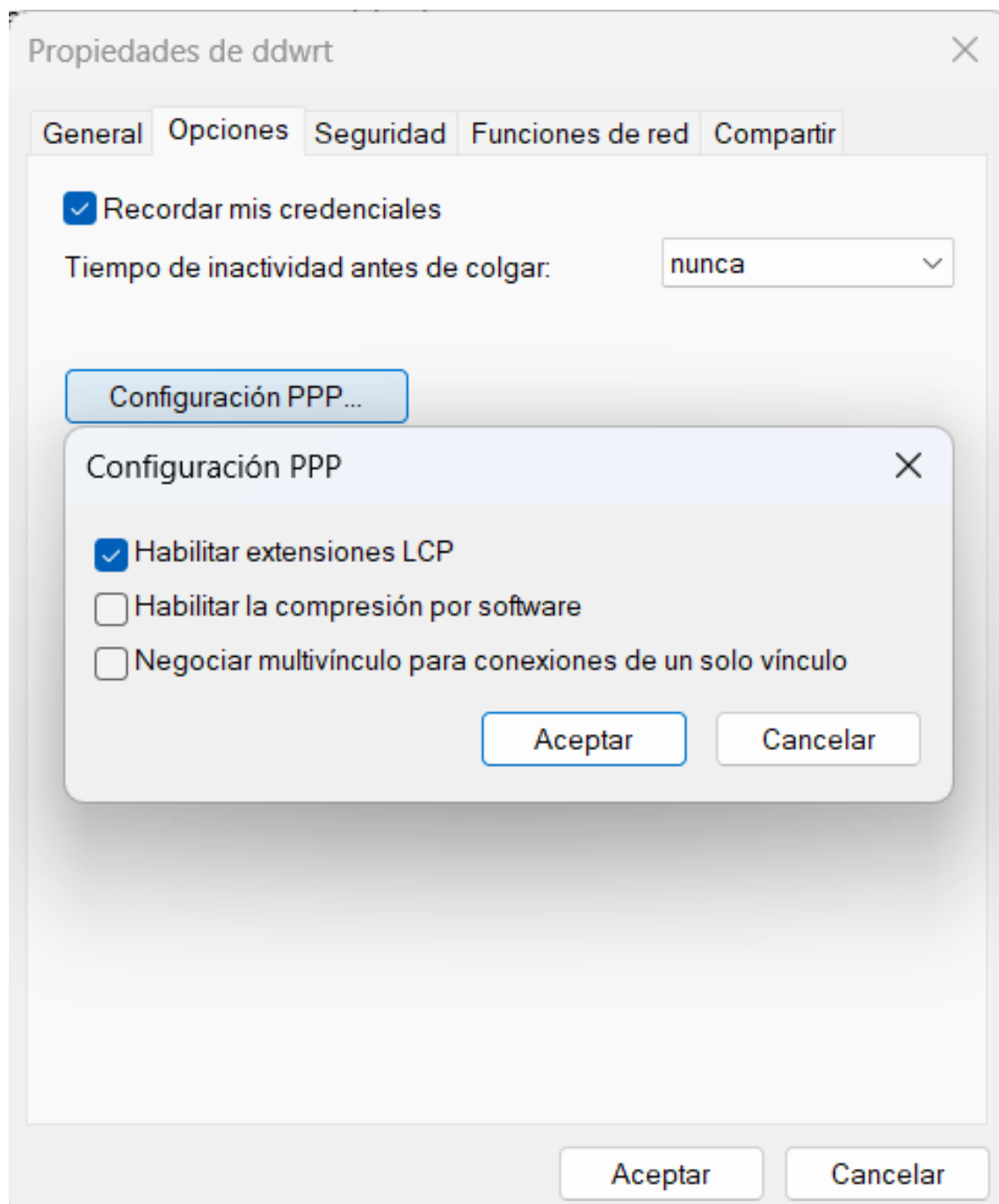


Figura 2.23: Ajustes VPN PPTP

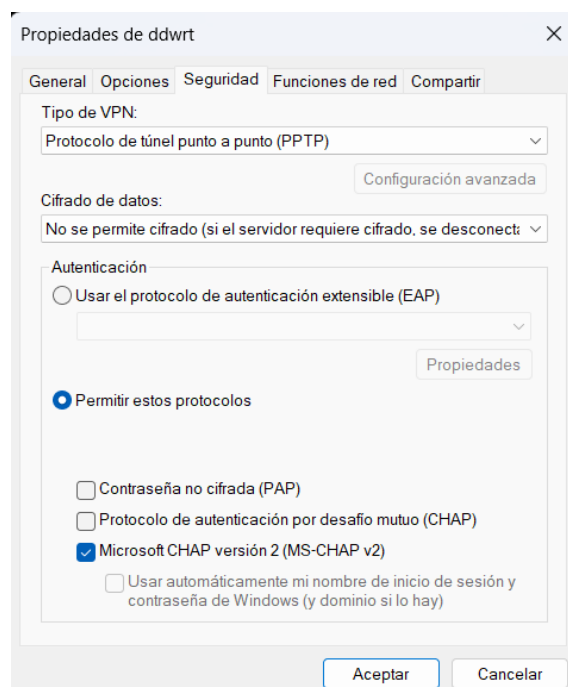


Figura 2.24: Ajustes VPN PPTP

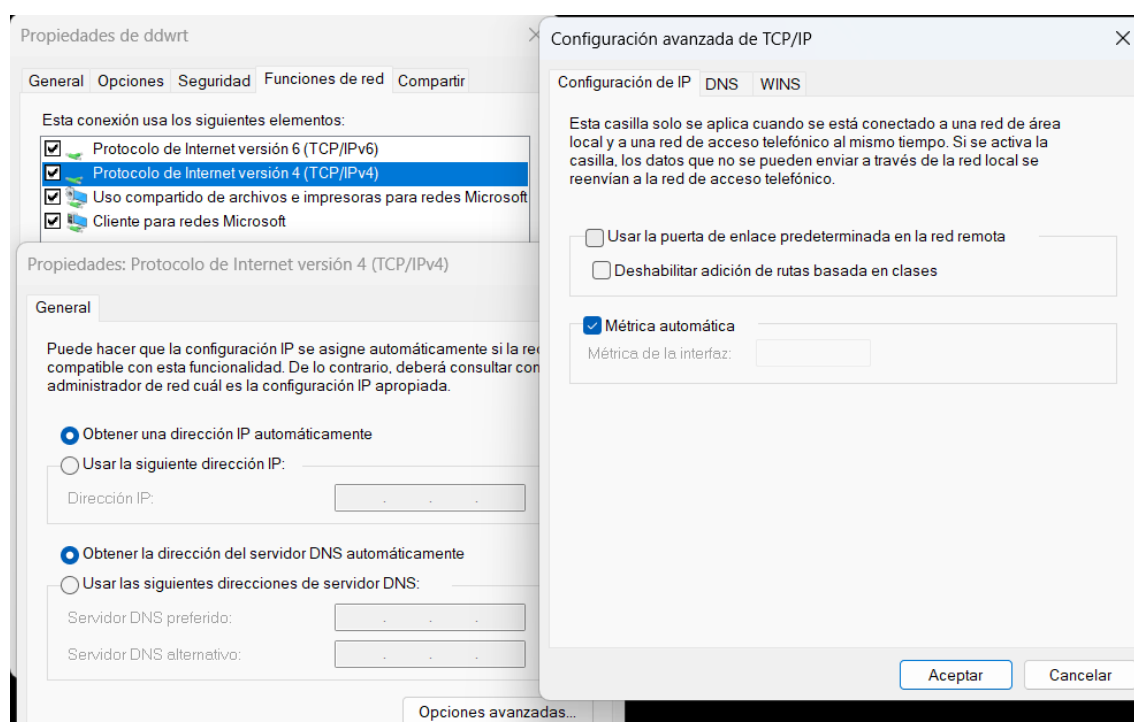


Figura 2.25: Ajustes VPN PPTP

#### 2.5.4. Verificación de la configuración del túnel PPTP remote-access

Para comprobar que la configuración ha sido correcta deberemos de comprobar que en el router ddwrt-X aparecen nuevas interfaces de red 'pppX' y nuevas entradas en la tabla de enrutamiento vinculadas a esas interfaces de red. Podemos ver los resultados a obtener en las figuras 2.26 y 2.27 para el PC anfitrión y para el router ddwrt-X las figuras 2.28 y 2.30.

```

PS C:\Users\Work> route print -4
=====
Ilista de interfaces
21.....Kaspersky VPN
13...16 13 33 c6 3e 2f .....Microsoft Wi-Fi Direct Virtual Adapter
16...16 13 33 c6 3e 3f .....Microsoft Wi-Fi Direct Virtual Adapter #2
10...5c 60 ba 58 1a 0f .....Realtek Gaming GbE Family Controller
18...00 50 56 c0 00 01 .....VMware Virtual Ethernet Adapter for VMnet1
15...00 50 56 c0 00 08 .....VMware Virtual Ethernet Adapter for VMnet8
66.....ddwrt
17...14 13 33 c6 3e 3f .....MediaTek MT7921 Wi-Fi 6 802.11ax PCIe Adapter
12...14 13 33 c6 3e 3e .....Bluetooth Device (Personal Area Network)
1.....Software Loopback Interface 1
=====

IPv4 Tabla de enrutamiento
=====
Rutas activas:
Destino de red      Máscara de red      Puerta de enlace    Interfaz  Métrica
0.0.0.0             0.0.0.0             192.168.1.1         192.168.1.183    35
10.0.0.0            255.0.0.0           10.54.161.101       10.54.161.201    36
10.24.161.0         255.255.255.0       En vínculo          10.24.161.3      291
10.24.161.3         255.255.255.255     En vínculo          10.24.161.3      291
10.24.161.255       255.255.255.255     En vínculo          10.24.161.3      291
10.54.161.0         255.255.255.0       En vínculo          10.54.161.201    36
10.54.161.201       255.255.255.255     En vínculo          10.54.161.201    291
10.54.161.255       255.255.255.255     En vínculo          10.54.161.201    291
127.0.0.0           255.0.0.0           En vínculo          127.0.0.1        331
127.0.0.1           255.255.255.255     En vínculo          127.0.0.1        331
127.255.255.255     255.255.255.255     En vínculo          127.0.0.1        331
192.168.1.0         255.255.255.0       En vínculo          192.168.1.183    291
192.168.1.183       255.255.255.255     En vínculo          192.168.1.183    291
192.168.1.255       255.255.255.255     En vínculo          192.168.1.183    291
192.168.189.0       255.255.255.0       En vínculo          192.168.189.1    291
192.168.189.1       255.255.255.255     En vínculo          192.168.189.1    291
192.168.189.129     255.255.255.255     En vínculo          192.168.189.1    36
192.168.189.255     255.255.255.255     En vínculo          192.168.189.1    291
224.0.0.0           240.0.0.0           En vínculo          127.0.0.1        331
224.0.0.0           240.0.0.0           En vínculo          192.168.1.183    291
224.0.0.0           240.0.0.0           En vínculo          192.168.189.1    291
224.0.0.0           240.0.0.0           En vínculo          10.24.161.3      291
224.0.0.0           240.0.0.0           En vínculo          10.54.161.201    291
255.255.255.255     255.255.255.255     En vínculo          127.0.0.1        331
255.255.255.255     255.255.255.255     En vínculo          192.168.1.183    291
255.255.255.255     255.255.255.255     En vínculo          192.168.189.1    291
255.255.255.255     255.255.255.255     En vínculo          10.24.161.3      291
255.255.255.255     255.255.255.255     En vínculo          10.54.161.201    291
=====
Rutas persistentes:
Ninguno

```

Figura 2.26: route print PC anfitrión

```

Adaptador PPP ddwrt:

Sufijo DNS específico para la conexión. . . :
Dirección IPv4. . . . . : 10.54.161.201
Máscara de subred . . . . . : 255.255.255.255
Puerta de enlace predeterminada . . . . . :

```

Figura 2.27: interfaz PC anfitrión



```

ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.54.161.101 P-t-P:10.54.161.201  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1396  Metric:1
          RX packets:2671  errors:1 dropped:0 overruns:0 frame:0
          TX packets:2500  errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:216171 (211.1 KiB)  TX bytes:432898 (422.7 KiB)

```

Figura 2.28: interfaces ddwrt-X

```

root@DD-WRT:~# ip route
10.54.161.201 dev ppp0  proto kernel  scope link  src 10.54.161.101
192.168.189.2 dev eth0  scope link
10.54.161.0/24 dev br0  proto kernel  scope link  src 10.54.161.101
192.168.189.0/24 dev eth0 proto kernel  scope link  src 192.168.189.129
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 192.168.189.2 dev eth0

```

Figura 2.29: tablas routing ddwrt-X

Después de haber realizado la configuración requerida, el esquema de red resultante con el túnel site-to-site se muestra en la figura

```

root@DD-WRT:~# ip route
10.54.161.201 dev ppp0  proto kernel  scope link  src 10.54.161.101
192.168.189.2 dev eth0  scope link
10.54.161.0/24 dev br0  proto kernel  scope link  src 10.54.161.101
192.168.189.0/24 dev eth0 proto kernel  scope link  src 192.168.189.129
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 192.168.189.2 dev eth0

```

Figura 2.30: tablas routing ddwrt-X



---

## CAPÍTULO 3

# Funcionamiento del túnel PPTP site-to-site

---

### 3.1 Explicación teórica

---

Desde el punto de vista teórico, el PPTP site-to-site o Protocolo de Tunnelización Punto a Punto nos permite poder conectar dos redes remotas a través de internet como si estuviesen en una misma red privada. Se considera un tipo de VPN[1] que utiliza PPTP para encapsular los paquetes y cifrar los datos que se transmiten entre los dos puntos, haciéndose servir del protocolo PPP original[2]. Este tipo de VPN se sitúa en el N2 del modelo OSI ya que está en el nivel de enlace de datos.

En este caso, nuestro servidor ddwrt-X va a estar conectado a través de un túnel PPTP a nuestro cliente ddwrt-noX, a través de la red VMNet8.

### 3.2 Fases en PPP

---

En un PPP hay tres fases principalmente[3]:

1. Fase de Control del Enlace (LCP): establece los parámetros básicos necesarios en la transmisión de datos por la capa física y se negocian el MRU y el CRC a utilizar, así como la autenticación y los requisitos de padding.
2. Fase de Autenticación: los peers se identifican mutuamente con lo acordado en el LCP (fase anterior), se incluye el PAP, CHAP y EAP.
3. Fase de Control del Protocolo de Red (NCP): se acuerdan los parámetros de red como IP que va a transportar el PPP.

Por esto mismo, hay dos tipos de paquetes, los que llamaremos de control, para establecer los parámetros de la primera fase (los paquetes PPP LCP) y luego los que contendrán los datos, en los que se empleará una versión extendida de los paquetes GRE, encargados de encapsular las siguientes cabeceras proporcionadas por el protocolo y así codificar la información.

Esto lo veremos a continuación en más detalle en las capturas que añadiremos del Wireshark.

### 3.3 Prueba 1 pings entre RCO-noX y RCO-X

Para ello hemos realizado el ping desde RCO-noX hacia RCO-X (que tiene la IP 10.54.161.137) y comprobar si pueden comunicarse el cliente de PPTP con el servidor.

Este comando no es más que una utilidad contenida en prácticamente todos los equipos, que nos ayuda a comprobar la conectividad entre dos máquinas. Lo que realiza es un envío de paquetes ICMP, con un número de identificación y de tipo request. El servidor, si recibe estos paquetes, contesta a la misma máquina que se los ha enviado con paquete ICMP, con la misma información pero en este caso cambia el tipo a reply.

Sí se consigue realizar la conexión, tendremos una traza de paquetes que consistirán en pares de request-reply y con lo que podremos concluir que ambas máquinas tienen conectividad.

Procedemos a realizar el ping, desde RCO-noX con el formato: ping <ip destino>

```
[root@rc0-nox ~]# ping 10.54.161.137
PING 10.54.161.137 (10.54.161.137) 56(84) bytes of data.
64 bytes from 10.54.161.137: icmp_seq=1 ttl=62 time=4.03 ms
64 bytes from 10.54.161.137: icmp_seq=2 ttl=62 time=4.67 ms
64 bytes from 10.54.161.137: icmp_seq=3 ttl=62 time=3.97 ms
64 bytes from 10.54.161.137: icmp_seq=4 ttl=62 time=3.10 ms
64 bytes from 10.54.161.137: icmp_seq=5 ttl=62 time=4.23 ms
64 bytes from 10.54.161.137: icmp_seq=6 ttl=62 time=4.45 ms
64 bytes from 10.54.161.137: icmp_seq=7 ttl=62 time=5.30 ms
64 bytes from 10.54.161.137: icmp_seq=8 ttl=62 time=4.97 ms
64 bytes from 10.54.161.137: icmp_seq=9 ttl=62 time=4.22 ms
64 bytes from 10.54.161.137: icmp_seq=10 ttl=62 time=3.14 ms
64 bytes from 10.54.161.137: icmp_seq=11 ttl=62 time=5.31 ms
^C
--- 10.54.161.137 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10017ms
rtt min/avg/max/mdev = 3.103/4.306/5.310/0.717 ms
```

Figura 3.1: Ping desde RCO-noX a RCO-X en site-to-site

Como se puede observar en la figura 3.1, el resultado del ping es correcto ya que de 11 paquetes que hemos transmitido, hemos recibido la respuesta de 11 más. Para poder entenderlo con más profundidad, hemos activado el Wireshark mientras realizábamos la prueba, con las redes VMnet1 y VMnet8 seleccionadas, y así poder capturar el tráfico y poder analizarlo.

Primero vamos a mostrar los paquetes de forma separada por comodidad.

En la figura 3.2 podemos observar varios tipos de paquetes:

1. Paquetes ICMP: son los paquetes del ping (con Echo reply y request) que podemos ver de forma clara, ya que en VMnet1 vemos el envío antes de pasar por el túnel y después al contestar. A demás se puede observar como las IP son las esperadas tanto de origen como de destino.

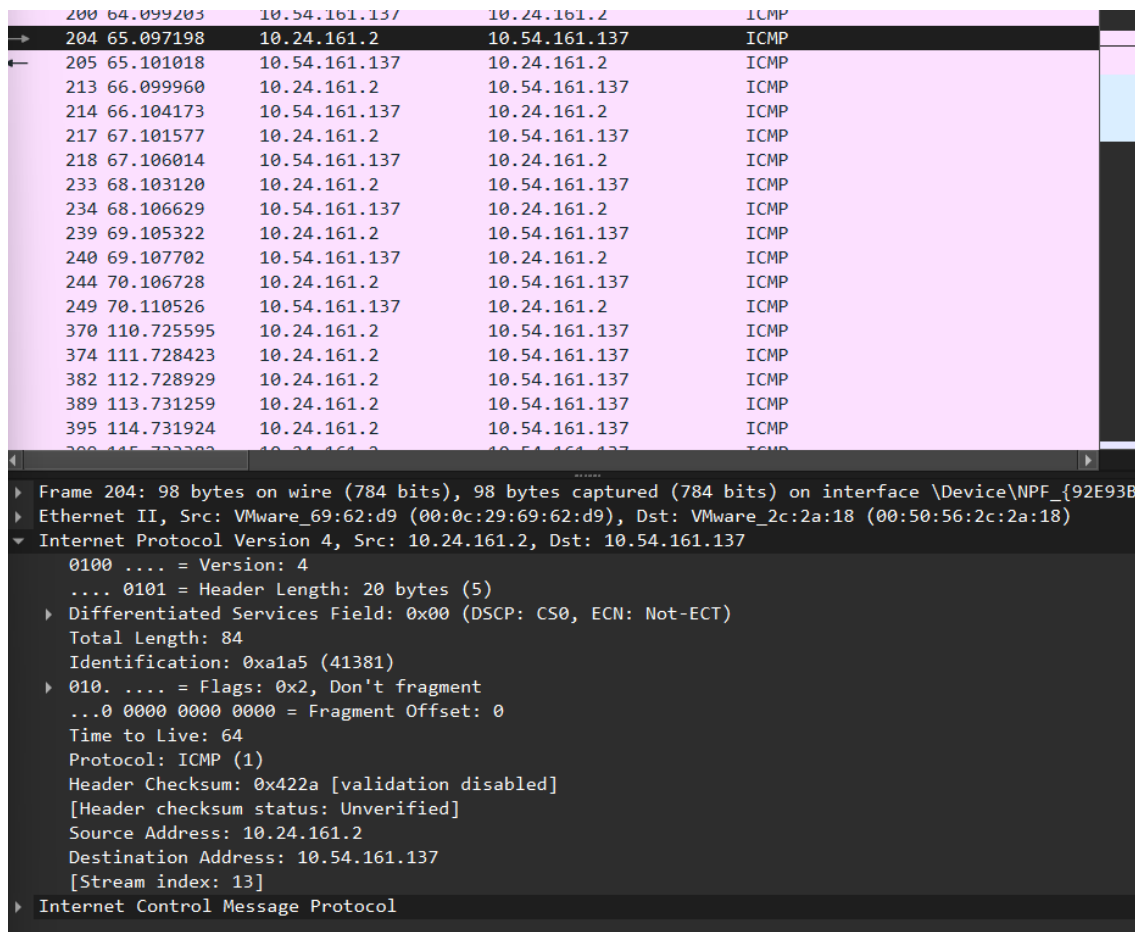


Figura 3.2: Paquete ICMP en VMnet1

Como se puede observar dentro de las cabeceras, se incluye la IP desde donde se ha mandado el paquete, y la IP de destino que es nuestra máquina RCO-X, por tanto podemos comprobar que es una traza correcta y que tienen conectividad, como habíamos aclarado en el comando ping.

2. Paquetes ARP: En este caso no aplican, pero están preguntando a quien les corresponden las direcciones que desconocemos pidiéndoselo a la dirección broadcast. Estos paquetes, tambien son capturados por la VMnet1.

3. Paquetes de Control PPP LCP: como hemos comentado en la explicación teórica, en la fase de control se envían estos paquetes para acordar entre cliente servidor del point-to-point la información que se va a emplear para realizar la conexión tal y como se muestra en la figura 3.3

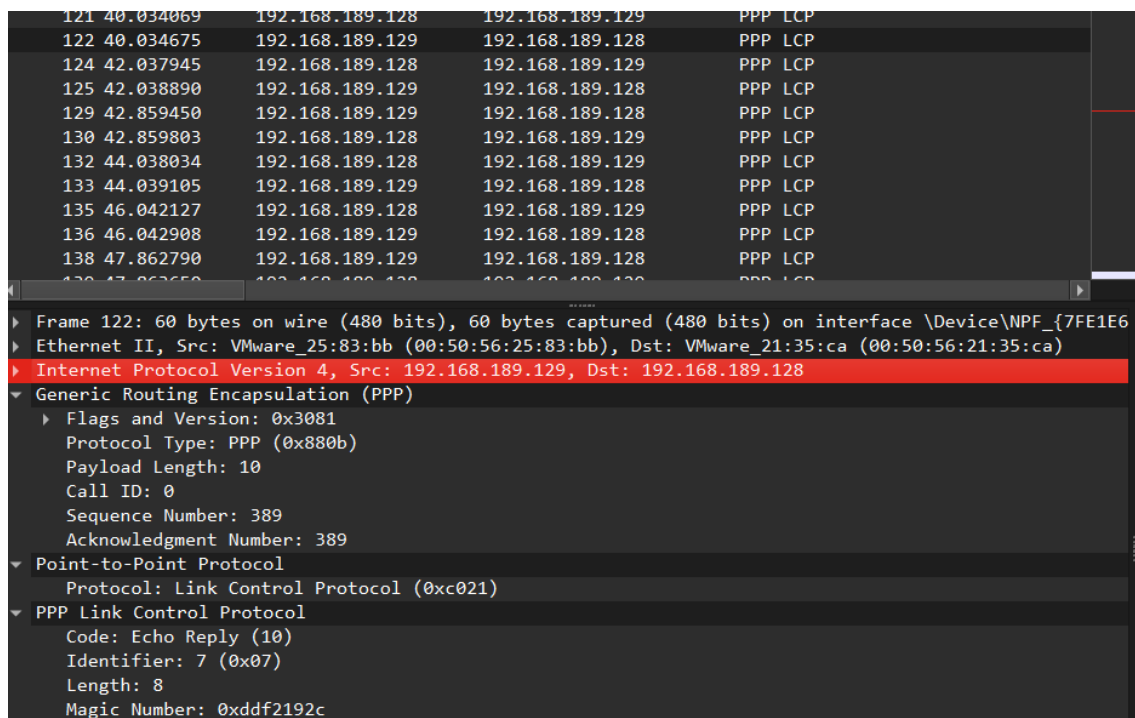


Figura 3.3: Paquete LCP en VMnet8

En este caso, se sabe que son este tipo de paquetes gracias a la cabecera PTPP con el campo Protocol con valor 0xC021, que indica que es un LCP y como podemos observar en la cabecera de IPv4 se realiza desde la IP de ddwrt-noX a ddwrt-X.

4. Paquetes PPP Comp: En estos paquetes, tenemos la traza ICMP (es decir, lo generado por el ping) pero de una forma encapsulada por el GRE modificado mencionado anteriormente y como se puede observar en la figura 3.4, antes de la información que se va a transmitir. Como tal se puede observar en la cabecera PTPP que el protocolo es de tipo datagrama comprimido (en hexadecimal 0x00FD) y justo después de esto, tenemos el PPP que contiene la información.

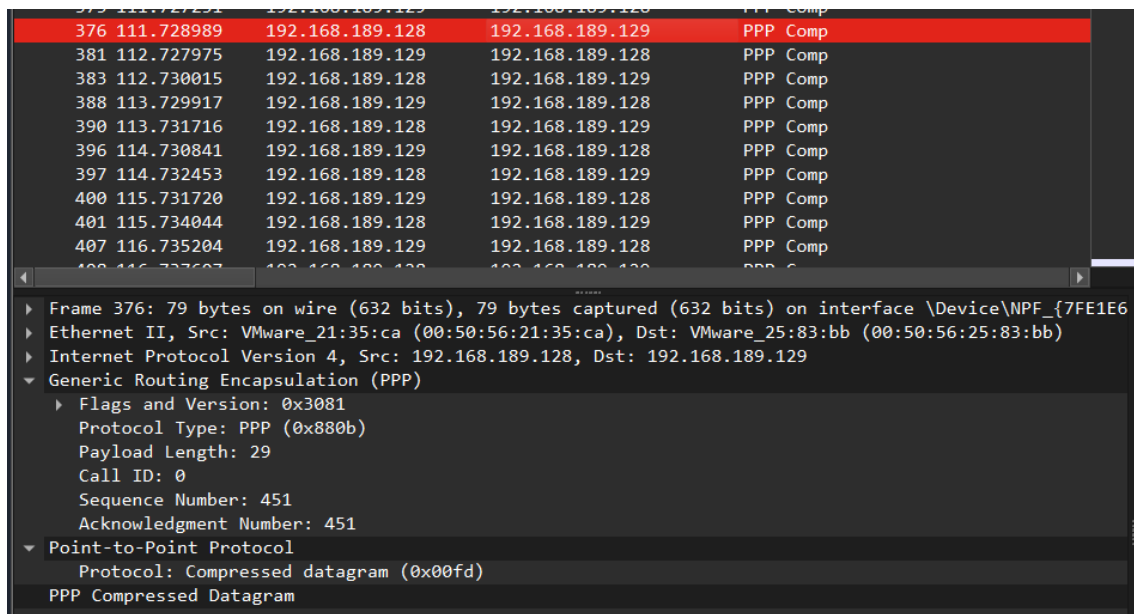


Figura 3.4: Paquete PPP Comp en VMnet8

5. Paquetes TCP: se encargan de que la información que se ha enviado llegue a su destino y sean entregadas, a demás de una forma ordenada.

6. Paquetes de tipo GRE: como se ha mencionado anteriormente para encapsular los paquetes de IP sobre el túnel. Estos paquetes se encargan de encapsular PPP como se puede observar en la información de los paquetes, y como hemos dicho, siendo modificados, añadiendo el bit A que si esta activo, indica que se ha realizado otra modificación, añadiendo un id justo detrás y modificando tamaños de diferentes campos tal y como se muestra en la figura 3.5

357	100.583370	192.168.189.128	192.168.189.128	GRE	46 Encapsulated PPP
360	107.930497	192.168.189.129	192.168.189.128	GRE	46 Encapsulated PPP
363	108.589809	192.168.189.128	192.168.189.129	GRE	46 Encapsulated PPP
369	110.594104	192.168.189.128	192.168.189.129	GRE	46 Encapsulated PPP
373	110.778630	192.168.189.129	192.168.189.128	GRE	46 Encapsulated PPP
377	111.779030	192.168.189.129	192.168.189.128	GRE	46 Encapsulated PPP
380	112.598335	192.168.189.128	192.168.189.129	GRE	46 Encapsulated PPP
384	112.782595	192.168.189.129	192.168.189.128	GRE	46 Encapsulated PPP
387	112.930589	192.168.189.129	192.168.189.128	GRE	46 Encapsulated PPP

```

Frame 360: 46 bytes on wire (368 bits), 46 bytes captured (368 bits) on interface \Device\NPF_{7FE1E660-70EC-47F2-B589-89DB454C80BD}, id 0
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: 192.168.189.129, Dst: 192.168.189.128
Generic Routing Encapsulation (PPP)
  Flags and Version: 0x2081
  Protocol Type: PPP (0x880b)
  Payload Length: 0
  Call ID: 0
  Acknowledgment Number: 447

```

Figura 3.5: Paquete GRE modificado en VMnet8

7. Paquetes NTP[4]: Basados en UDP, se utilizan para sincronizar el reloj de cliente-servidor y que no haya conflictos con los TTLs.

Ahora hemos mostrado los paquetes por separado, para poder reconocerlos de una forma más cómoda y poder explicarlos mejor, pero realmente, si los ordenamos por tiempo se verían como la figura 3.6:

1	0.000000	192.168.189.128	192.168.189.129	PPP LCP	56 Echo Request
2	0.000865	192.168.189.129	192.168.189.128	PPP LCP	60 Echo Reply
3	0.499873	192.168.189.128	192.168.189.129	GRE	46 Encapsulated PPP
4	2.004185	192.168.189.128	192.168.189.129	PPP LCP	56 Echo Request
5	2.005014	192.168.189.129	192.168.189.128	PPP LCP	60 Echo Reply
6	2.503461	192.168.189.128	192.168.189.129	GRE	46 Encapsulated PPP
7	2.545376	192.168.189.129	193.149.0.221	NTP	90 NTP Version 4, client
8	2.577638	193.149.0.221	192.168.189.129	NTP	90 NTP Version 4, server
9	2.691456	192.168.189.129	94.143.139.219	NTP	90 NTP Version 4, client
10	2.716322	94.143.139.219	192.168.189.129	NTP	90 NTP Version 4, server
11	2.852570	192.168.189.129	192.168.189.128	PPP LCP	56 Echo Request
12	2.853382	192.168.189.128	192.168.189.129	PPP LCP	60 Echo Reply
13	2.904578	192.168.189.129	192.168.189.128	GRE	46 Encapsulated PPP

Figura 3.6: Primeros paquetes al realizar el ping

Aquí, podemos observar como primero, se envían los paquetes de configuración LCP, obteniendo respuesta desde el servidor, y después se procede a enviar de forma encapsulada la información con el paquete GRE, a demás de estar los paquetes NTP que se van enviando de forma periódica para sincronizar los relojes.



En la figura 3.7, podemos ver como capturando simultáneamente VMnet1 y VMnet8, están los paquetes que se envían a con la traza ICMP (VMnet8) desde RCO-noX a RCO-X y cómo en la red VMnet1, esto se transforma primero a paquetes de configuración, los paquetes PPP LCP, de ahí se realiza el ping desde el cliente, y se observa en la VMnet8 como aparecen las trazas ICMP. Siguiendo a esto, el tunel nos comprime esa información en paquetes de tipo PPP Comp, donde se contiene la información a transmitir y de ahí se crean los paquetes de tipo GRE (modificado) que contienen la información de dicho paquete encapsulado, es decir el PPP pero en forma encapsulada. Cuando el servidor contesta, es lo mismo pero a la inversa y con paquetes de reply en vez de request (como se puede observar a la derecha de la imagen).

170	60.055366	192.168.189.128	192.168.189.129	PPP LCP	56 Echo Request
171	60.056175	192.168.189.129	192.168.189.128	PPP LCP	60 Echo Reply
168	60.089317	10.24.161.2	10.54.161.137	ICMP	98 Echo (ping) request id=0x0001, seq=1/256, ttl=64 (reply in 169)
172	60.090147	192.168.189.128	192.168.189.129	PPP Comp	137 Compressed data
173	60.092204	192.168.189.129	192.168.189.128	PPP Comp	130 Compressed data
169	60.092736	10.54.161.137	10.24.161.2	ICMP	98 Echo (ping) reply id=0x0001, seq=1/256, ttl=62 (request in 168)
174	60.591082	192.168.189.128	192.168.189.129	GRE	46 Encapsulated PPP
175	61.091209	10.24.161.2	10.54.161.137	ICMP	98 Echo (ping) request id=0x0001, seq=2/512, ttl=64 (reply in 176)
177	61.092100	192.168.189.128	192.168.189.129	PPP Comp	75 Compressed data
178	61.094345	192.168.189.129	192.168.189.128	PPP Comp	78 Compressed data
176	61.095017	10.54.161.137	10.24.161.2	ICMP	98 Echo (ping) reply id=0x0001, seq=2/512, ttl=62 (request in 175)
179	61.595661	192.168.189.128	192.168.189.129	GRE	46 Encapsulated PPP
180	62.055660	192.168.189.128	192.168.189.129	PPP LCP	56 Echo Request
181	62.056629	192.168.189.129	192.168.189.128	PPP LCP	60 Echo Reply

Figura 3.7: VMnet1 y VMnet8 capturadas de forma simultánea al realizar el ping

### 3.4 Prueba 2 pings de anfitrión a RCO-X no funcionan

En esta segunda prueba se nos pide que realicemos pings entre el PC anfitrión y RCO-X, para comprobar que entre estos dos no hay comunicación, a pesar de estar en la misma red VMnet1 a través del túnel.

Para comprobar esto, como hemos mencionado anteriormente, realizaremos el ping, pero en este caso no deberíamos recibir los paquetes de vuelta, es decir, debe salirnos el paquete de tipo request pero no recibir los de tipo reply, como podemos observar en la figura 3.8, esto se cumple:

```
PS C:\Users\Work> ping 10.54.161.137

Haciendo ping a 10.54.161.137 con 32 bytes de datos:
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 10.54.161.137:
    Paquetes: enviados = 4, recibidos = 0, perdidos = 4
              (100% perdidos),
```

Figura 3.8: Ping entre PC anfitrión y RCO-X

Lo que podemos observar, viene dado por culpa de las tablas de enrutamiento, ya que nuestro equipo anfitrión no conoce la red 10.54.161.0/18. Por ello, lo que se puede realizar, es configurar nuestro equipo para que todas las peticiones que vayan dirigidas a dicha red, sean enviadas a través de la red 10.24.161.0/18, es decir, la red VMnet1.

Para poder obtener esta funcionalidad, lo que realizaremos será añadir una regla usando el comando `route ADD 10.54.161.0 MASK 255.255.255.0 10.24.161.1 METRIC 2 IF 18`. Como se puede muestra en la figura 3.9

```
PS C:\Users\Work> route ADD 10.54.161.0 MASK 255.255.255.0 10.24.161.1 METRIC 2 IF 18
Correcto
```

Figura 3.9: Comando de modificación de las tablas de enrutamiento

Con esto lo que hacemos es añadir una nueva ruta a la tabla de enrutamiento, donde la red destino será 10.54.161.0 con una máscara que fija las posibles IP desde 10.54.161.0 hasta 10.54.161.255, y donde la puerta de enlace por la cual se enviarán todos los paquetes serán a través de la 10.24.161.1, es decir la IP del router cliente para el túnel PPTP. Con METRIC 2, añadimos la prioridad de la ruta, donde cuanto menor sea este valor, más prioridad tienen, por si acaso existen varias rutas para un mismo destino, se usará la que tenga la métrica más baja. En cuanto al IF 18, lo que define es el identificador de red que se usará para esa ruta, que en nuestro caso es el 18.

Como se puede observar, se ha añadido de forma correcta ya que el comando nos responde con un Correcto al finalizar la tarea.

Para comprobar de todas formas, que esto se ha hecho de una forma correcta, vamos a correr el comando `route print -4`, que nos mostrará las tablas de enrutamiento para únicamente las rutas de IPv4, (por eso el -4) tal y como se muestran en las figuras 3.10 y 3.11

```
PS C:\Users\Work> route print -4
=====
Ilista de interfaces
21.....Kaspersky VPN
13...16 13 33 c6 3e 2f .....Microsoft Wi-Fi Direct Virtual Adapter
16...16 13 33 c6 3e 3f .....Microsoft Wi-Fi Direct Virtual Adapter #2
18...00 50 56 c0 00 01 .....VMware Virtual Ethernet Adapter for VMnet1
15...00 50 56 c0 00 08 .....VMware virtual Ethernet Adapter for VMnet8
17...14 13 33 c6 3e 3f .....MediaTek MT7921 Wi-Fi 6 802.11ax PCIe Adapter
12...14 13 33 c6 3e 3e .....Bluetooth Device (Personal Area Network)
10...5c 60 ba 58 1a 0f .....Realtek Gaming GbE Family Controller
1.....Software Loopback Interface 1
=====
```

Figura 3.10: Comando de imprimir las tablas de enrutamiento, lista de interfaces

```
IPv4 Tabla de enrutamiento
=====
Rutas activas:
Destino de red      Máscara de red      Puerta de enlace      Interfaz      Métrica
10.0.0.0            0.0.0.0            192.168.1.1          192.168.1.134    35
10.24.161.0         255.255.255.0      En vínculo           10.24.161.3      291
10.24.161.3         255.255.255.255    En vínculo           10.24.161.3      291
10.24.161.255       255.255.255.255    En vínculo           10.24.161.3      291
10.54.161.0         255.255.255.0      10.24.161.1          10.24.161.3      37
```

Figura 3.11: Comando de imprimir las tablas de enrutamiento, tabla de enrutamiento

Se puede observar como el destino de red 10.54.161.0 irá encaminado por la puerta de enlace que hemos indicado en el comando anterior, es decir el gateway 10.24.161.1. y por tanto, debemos tener conectividad entre el PC anfitrión y RCO-X, ya que ahora si que puede resolver correctamente su ruta. Esto, lo realizaremos repitiendo el comando ping

del principio como se ve en la figura 3.12, pero en este caso, vamos a observar que si que se devuelven los pares request-reply:

```
PS C:\Users\Work> ping 10.54.161.137

Haciendo ping a 10.54.161.137 con 32 bytes de datos:
Respuesta desde 10.54.161.137: bytes=32 tiempo=4ms TTL=62
Respuesta desde 10.54.161.137: bytes=32 tiempo=4ms TTL=62
Respuesta desde 10.54.161.137: bytes=32 tiempo=4ms TTL=62
Respuesta desde 10.54.161.137: bytes=32 tiempo=4ms TTL=62

Estadísticas de ping para 10.54.161.137:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 4ms, Máximo = 4ms, Media = 4ms
```

Figura 3.12: Ping correcto entre PC anfitrión y RCO-X

No hemos añadido imágenes del Wireshark ya que la traza que íbamos a observar, es la misma que hay en la primera prueba, solo que cambiando la IP de origen, por la interfaz del equipo con la red VMnet1.

### 3.5 Prueba 3 tracert de PC anfitrión a RCO-X

En esta prueba, se nos pide realizar un tracert desde el PC anfitrión (ya que es Windows, si no, usaríamos traceroute, para Linux o Mac), y con ello, mostrar el recorrido que hace la conexión desde el PC anfitrión hasta RCO-X.

Para poder observar la diferencia, hemos aplicado el comando antes de aplicar la regla añadida en la prueba anterior, y con ello hemos obtenido el resultado de la figura 3.13:

```
PS C:\Windows> tracert 10.54.161.137

Traza a 10.54.161.137 sobre caminos de 30 saltos como máximo.

 1      1 ms      1 ms      1 ms    HOME [192.168.1.1]
 2      4 ms      3 ms      4 ms    10.248.0.1
 3     10 ms      8 ms      9 ms    10.14.7.101
 4     15 ms     14 ms     14 ms    10.0.12.33
 5      *        *        *      Tiempo de espera agotado para esta solicitud.
 6      *        *        *      Tiempo de espera agotado para esta solicitud.
 7      *        *        *      Tiempo de espera agotado para esta solicitud.
 8
```

Figura 3.13: Tracert entre PC anfitrión y RCO-X pre regla

Como era de esperar, y como se puede observar, no es capaz de trazar una ruta, ya que el elemento no es accesible. En cambio, una vez aplicada la regla anterior, si volvemos a repetir el mismo comando obtenemos el resultado de la figura 3.14

```
PS C:\Users\Work> tracert 10.54.161.137

Traza a 10.54.161.137 sobre caminos de 30 saltos como máximo.

  1      <1 ms      <1 ms      <1 ms  10.24.161.1
  2      14 ms      2 ms       2 ms  10.54.161.101
  3       3 ms      3 ms       3 ms  10.54.161.137

Traza completa.
```

Figura 3.14: Tracert entre PC anfitrión y RCO-X

En la figura 3.14, se puede observar que directamente, y de forma muy rápida (menos de 1ms) obtenemos la dirección de la puerta de enlace que hemos añadido en la regla anterior. Justo después, obtenemos que el siguiente salto lo hace a la dirección 10.54.161.101, que es la que corresponde con ddwrt-x y a su vez es el servidor del túnel PPTP. Finalmente, llegamos a la IP que antes era inalcanzable por el anfitrión y dirección de RCO-X, 10.54.161.137.

### 3.6 Prueba 4 Comprobación de conexión entre las máquinas RCO

En esta prueba vamos a trazar la ruta que sigue un paquete desde las máquinas RCO. Para ello, primero que nada vamos a instalar en ambas esta utilidad, utilizando el comando mostrado en las figuras 3.16 y 3.15.

```
[root@rc0-nox ~]# yum install traceroute
Última comprobación de caducidad de metadatos hecha hace 0:10:01, el mar 12 nov 2024 21:26:22 CET.
Dependencias resueltas.
=====
Paquete                Arquitectura      Versión           Repositorio       Tam.
-----
Instalando:
traceroute              x86_64            3:2.1.0-8.el8    baseos             66 k
=====
Resumen de la transacción
=====
Instalar 1 Paquete

Tamaño total de la descarga: 66 k
Tamaño instalado: 101 k
¿Está de acuerdo [s/N]? s
Descargando paquetes:
traceroute-3:2.1.0-8.el8.x86_64.rpm                                667 kB/s | 66 kB  00:00
-----
Total
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
Preparando :
Instalando : traceroute-3:2.1.0-8.el8.x86_64                      1/1
Ejecutando scriptlet: traceroute-3:2.1.0-8.el8.x86_64             1/1
Verificando : traceroute-3:2.1.0-8.el8.x86_64                     1/1
-----
Instalado:
traceroute-3:2.1.0-8.el8.x86_64
¡Listo!
```

Figura 3.15: Instalación de traceroute en RCO-noX

```
[root@rco-x ~]# yum install traceroute
Última comprobación de caducidad de metadatos hecha hace 0:08:50, el mar 12 nov 2024 21:27:05 CET.
Dependencias resueltas.
=====
Paquete                Arquitectura  Versión          Repositorio      Tam.
=====
Instalando:
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| traceroute | x86_64 | 3:2.1.0-8.el8 | baseos | 66 k |

Resumen de la transacción
=====
Instalar 1 Paquete

Tamaño total de la descarga: 66 k
Tamaño instalado: 101 k
¿Está de acuerdo [s/N]?: s
Descargando paquetes:
traceroute-2.1.0-8.el8.x86_64.rpm          327 kB/s | 66 kB      00:00
-----
Total                                     24 kB/s | 66 kB      00:02
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
  Preparando      :                               1/1
  Instalando      : traceroute-3:2.1.0-8.el8.x86_64 1/1
  Ejecutando scriptlet: traceroute-3:2.1.0-8.el8.x86_64 1/1
  Verificando     : traceroute-3:2.1.0-8.el8.x86_64 1/1
Productos instalados actualizados.

Instalado:
  traceroute-3:2.1.0-8.el8.x86_64

¡Listo!
```

Figura 3.16: Instalación de traceroute en RCO-X

Ahora, tenemos instalada en ambas máquinas esta utilidad.

Principalmente, el funcionamiento de este comando, lo que hace es a través de paquetes ICMP (los mismos que usa el comando ping) coger el campo TTL (Time to Live) de la cabecera IP, siendo este un número entero que va disminuyendo por cada uno de los nodos que pasa y que cuando llega su valor a 0, es descartado[5]. El comando, envía un mensaje por nodo que vaya encontrando, aumentando el TTL para poder ir haciendo un mapeo ya que lo irá haciendo por orden de los que vaya encontrando.

Ahora, sabiendo lo que hace más en profundidad este comando, vamos a comprobar que sucede y explicarlo:

```
[root@rco-nox ~]# traceroute 10.54.161.137
traceroute to 10.54.161.137 (10.54.161.137), 30 hops max, 60 byte packets
 1  _gateway (10.24.161.1)  1.129 ms  0.955 ms  0.882 ms
 2  10.54.161.101 (10.54.161.101)  8.301 ms  8.833 ms  9.532 ms
 3  10.54.161.137 (10.54.161.137)  9.490 ms !X  10.055 ms !X  10.518 ms !X
```

Figura 3.17: Uso de traceroute en RCO-noX

```
[root@rco-x ~]# traceroute 10.24.161.2
traceroute to 10.24.161.2 (10.24.161.2), 30 hops max, 60 byte packets
 1  DD-WRT (10.54.161.101)  0.969 ms  0.824 ms  0.758 ms
 2  10.24.161.1 (10.24.161.1)  10.290 ms  12.371 ms  13.668 ms
 3  10.24.161.2 (10.24.161.2)  13.579 ms !X  13.527 ms !X  13.557 ms !X
```

Figura 3.18: Uso de traceroute en RCO-X

Como se puede observar, en la figura 3.17 realizamos el traceroute desde rco-nox, hacia la IP del RCO-X (10.54.161.137) y al igual que desde el equipo anfitrión, son tres saltos,

primero, encuentra el gateway predeterminado, que es el ddwrt-noX, con IP 10.24.161.1. Justo después, redirecciona a ddwrt-x es decir, el servidor del túnel, y finalmente desde ahí llega el paquete a RCO-X.

En cambio, observando la figura 3.18 desde RCO-X hacia RCO-noX, lo que obtenemos es que primero busca el gateway que hay definido en la VMNet2, es decir, el router ddwrt-x y servidor del túnel, que encamina el tráfico a ddwrt-noX y de ahí directamente se lo hace llegar a nuestra máquina RCO-noX.

En la figura 3.19 un ejemplo del tráfico generado por este comando, pudiendo observar como comentabamos el funcionamiento de traceroute, enviando un fin del TTL y por tanto obteniendo la información de los distintos nodos:

214	15.098865	10.24.161.1	10.24.161.2	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
219	15.099364	10.24.161.1	10.24.161.2	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
220	15.099427	10.24.161.1	10.24.161.2	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
236	15.108555	10.54.161.101	10.24.161.2	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
237	15.109470	10.54.161.101	10.24.161.2	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
238	15.109513	10.54.161.101	10.24.161.2	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
239	15.109558	10.54.161.137	10.24.161.2	ICMP	102 Destination unreachable (Communication administratively filtered)
240	15.109615	10.54.161.137	10.24.161.2	ICMP	102 Destination unreachable (Communication administratively filtered)
241	15.110303	10.54.161.137	10.24.161.2	ICMP	102 Destination unreachable (Communication administratively filtered)
242	15.110572	10.54.161.137	10.24.161.2	ICMP	102 Destination unreachable (Communication administratively filtered)
243	15.110616	10.54.161.137	10.24.161.2	ICMP	102 Destination unreachable (Communication administratively filtered)
244	15.110663	10.54.161.137	10.24.161.2	ICMP	102 Destination unreachable (Communication administratively filtered)

Figura 3.19: Traceroute en Wireshark

Y en la figura 3.20 la información que obtiene de las cabeceras, donde podemos observar que son paquetes ICMP con (el seleccionado por ejemplo) tiene un TTL de 1, y se ha excedido, por tanto va a devolver un time-out, con la información del nodo al que ha llegado. A demás como se puede observar en la línea de color rojo, nos indica que esto puede ser debido a un traceroute:

238	15.109513	10.54.161.101	10.24.161.2	ICMP	102 Time-to-live exceeded (Time to live exceeded in transit)
239	15.109558	10.54.161.137	10.24.161.2	ICMP	102 Destination unreachable (Communication administratively filtered)
240	15.109615	10.54.161.137	10.24.161.2	ICMP	102 Destination unreachable (Communication administratively filtered)
241	15.110303	10.54.161.137	10.24.161.2	ICMP	102 Destination unreachable (Communication administratively filtered)
242	15.110572	10.54.161.137	10.24.161.2	ICMP	102 Destination unreachable (Communication administratively filtered)
243	15.110616	10.54.161.137	10.24.161.2	ICMP	102 Destination unreachable (Communication administratively filtered)
244	15.110663	10.54.161.137	10.24.161.2	ICMP	102 Destination unreachable (Communication administratively filtered)

.... 0101 = Header Length: 20 bytes (5)
► Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 60
Identification: 0x9dcc (40396)
► 000. .... = Flags: 0x0
...0 0000 0000 0000 = Fragment Offset: 0
▼ Time to Live: 1
▼ [Expert Info (Note/Sequence): "Time To Live" only 1]
["Time To Live" only 1]
[Severity level: Note]
[Group: Sequence]
Protocol: UDP (17)
Header Checksum: 0xc50b [validation disabled]
[Header checksum status: Unverified]
Source Address: 10.24.161.2
Destination Address: 10.54.161.137
[Stream index: 12]
User Datagram Protocol, Src Port: 37255, Dst Port: 33439
Source Port: 37255
Destination Port: 33439
▼ [Expert Info (Chat/Sequence): Possible traceroute: hop #2, attempt #2]
[Possible traceroute: hop #2, attempt #2]
[Severity level: Chat]
[Group: Sequence]

Figura 3.20: Traceroute en Wireshark: cabeceras

## 3.7 Prueba 5 Tablas de routing

Para esta prueba, se nos pide mostrar las tablas de ruta que tienen todas las máquinas. Para ello vamos a emplear el comando `ip route` en todos los equipos.

Primero vamos a mostrar las de los routers ddwrt:



```

root@DD-WRT:~# ip route
10.54.161.101 dev ppp0 proto kernel scope link src 10.24.161.1
192.168.189.2 dev eth0 scope link
10.24.161.0/24 dev br0 proto kernel scope link src 10.24.161.1
10.54.161.0/24 dev ppp0 scope link
192.168.189.0/24 dev eth0 proto kernel scope link src 192.168.189.128
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 192.168.189.2 dev eth0

```

**Figura 3.21:** Tabla de ruta de DDWRT-noX

Esta primera tabla, visible en la figura 3.21 muestra las rutas de ddwrt-noX y contiene ocho entradas:

1. 10.54.161.101: esta ruta indica que los paquetes enviados a la dirección IP 10.54.161.101 van a ser enviados a través de la interfaz de red ppp0, siendo la interfaz de enlace PPP que aparece al configurar el túnel PPTP, e indica con src que la dirección de origen será 10.24.161.1.
2. 192.168.189.2: la ruta apunta directamente a dicho host a través de eth0.
3. 10.24.161.0/24: esta indica que por la interfaz de red br0 (dado ese nombre por el bridge, al crear el puente) define esa red, con direcciones de 10.24.161.0 a 10.24.161.255.
4. 10.54.161.0/24: similar a la explicada anteriormente pero con la red 10.54.161.0/24, y a través de la interfaz ppp0, es decir, accesible a través del enlace PPP (la vpn).
5. 192.168.189.0/24: define la red donde los paquetes serán enviados por eth0.
6. 169.254.0.0/16: define la red de tipo APIPA (Automatic Private IP Addressing), para cuando el dispositivo no pueda obtener una IP válida, asignarle una de esta red, a través de la interfaz de red br0 donde la dirección de origen será 169.254.255.1.
7. 127.0.0.0/8: simplemente define el loopback, es decir el localhost y similares a través de la interfaz local lo.
8. default via: Esta entrada define la ruta por defecto (default gateway), donde todo el tráfico que no coincida con ninguna de las definidas anteriormente, será enviado a través de esta ruta por la interfaz eth0 hacia la dirección 192.168.189.2.

Ahora, vamos a mostrar la tabla de ddwrt-x, que se muestra en la figura 3.22:

```

root@DD-WRT:~# ip route
10.24.161.1 dev ppp0 proto kernel scope link src 10.54.161.101
192.168.189.2 dev eth0 scope link
10.24.161.0/24 via 10.24.161.1 dev ppp0
10.54.161.0/24 dev br0 proto kernel scope link src 10.54.161.101
192.168.189.0/24 dev eth0 proto kernel scope link src 192.168.189.128
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 192.168.189.2 dev eth0

```

**Figura 3.22:** Tabla de ruta de DDWRT-X

1. 10.24.161.101: esta ruta indica que los paquetes enviados a la dirección IP 10.24.161.101 van a ser enviados a través de la interfaz de red ppp0, siendo la interfaz de enlace PPP que aparece al configurar el túnel PPTP, e indica con src que la dirección de

origen será 10.54.161.1. Con esto, y el router anterior, vemos el camino completo del túnel.

2. 192.168.189.2: la ruta apunta directamente a dicho host a través de eth0.
3. 10.24.161.0/24: esta indica que por la interfaz de red br0 (dado ese nombre por el bridge, al crear el puente) define esa red, con direcciones de 10.24.161.0 a 10.24.161.255.
4. 10.54.161.0/24: similar a la explicada anteriormente pero con la red 10.54.161.0/24, y a través de la interfaz ppp0, es decir, accesible a través del enlace PPP (la vpn).
5. 192.168.189.0/24: define la red donde los paquetes serán enviados por eth0.
6. 169.254.0.0/16: define la red de tipo APIPA (Automatic Private IP Addressing), para cuando el dispositivo no pueda obtener una IP válida, asignarle una de esta red, a través de la interfaz de red br0 donde la dirección de origen será 169.254.255.1.
7. 127.0.0.0/8: simplemente define el loopback, es decir el localhost y similares a través de la interfaz local lo.
8. default via: Esta entrada define la ruta por defecto (default gateway), donde todo el tráfico que no coincida con ninguna de las definidas anteriormente, será enviado a través de esta ruta por la interfaz eth0 hacia la dirección 192.168.189.2.

Como se puede observar, entre un router y otro lo único que cambia es el primer enlace ya que cada uno apunta al contrario por así decirlo y completa el túnel usando al interfaz ppp0 creada al crear el túnel.

Ahora vamos mostrar las tablas de las dos máquinas RCO, primero RCO-noX, en la figura 3.23:

```
[root@rco-nox ~]# ip route
default via 10.24.161.1 dev ens37 proto static metric 100
10.24.161.0/24 dev ens37 proto kernel scope link src 10.24.161.2 metric 100
```

**Figura 3.23:** Tabla de ruta de RCO-noX

En este caso, la tabla de rutas es muy corta, ya que simplemente tiene en la interfaz ens37 para enviar todo el tráfico hacia redes externas por la puerta de enlace por defecto 10.24.161.1 (es decir, ddwrt-noX) y luego tiene definida la red 10.24.161.0/24 para que todo el tráfico de dicha red se enrute directamente por la interfaz ens37 y no haga falta que pase por la puerta de enlace, es decir, dos máquinas en la red se conecten directamente.

Por último vamos a ejecutar el traceroute en RCO-X, en la figura 3.24:

```
[root@rco-x ~]# ip route
default via 10.54.161.101 dev ens37 proto dhcp src 10.54.161.137 metric 100
10.54.161.0/24 dev ens37 proto kernel scope link src 10.54.161.137 metric 100
```

**Figura 3.24:** Tabla de ruta de RCO-X

Al igual que en RCO-noX, en la interfaz ens37 se define como puerta de enlace la 10.54.161.101 pero en este caso, se ha definido la IP de la máquina por dhcp. En cuanto a la red definida, en este caso es la 10.54.161.0/24 para la misma casuística que en la anterior, pero con esta red.



---

## CAPÍTULO 4

# Funcionamiento del túnel PPTP remote-access

---

El PPTP (Point-to-Point Tunneling Protocol), definido en la RFC [6], utiliza un modelo cliente-servidor para establecer túneles seguros sobre redes IP. En este modelo, el PAC (PPTP Access Concentrator) es el componente que actúa como el extremo del túnel en el lado de la red privada.

Rol de PAC:

1. Gestión del túnel: El PAC es responsable de establecer y administrar los túneles PPTP. Recibe conexiones de los clientes remotos (PPTP Network Servers o PNS) y facilita el acceso a la red privada.
2. Intermediario de datos: Encapsula y desencapsula los paquetes PPP dentro de GRE para transportarlos a través del túnel.
3. Autenticación: Colabora en los procesos de autenticación para validar la identidad del cliente antes de otorgar acceso.

En el contexto de una VPN, el PAC se encuentra en la red privada y sirve como el punto de acceso que gestiona los túneles PPTP creados por clientes remotos. Este dispositivo asegura que los datos encapsulados lleguen al destino correcto dentro de la red privada, permitiendo a los usuarios externos interactuar como si estuvieran localmente conectados.

### 4.1 Prueba 1. Pings entre PC anfitrión y RCO-X

---

Tras haber realizado la configuración necesaria explicada en el apartado (escribir apartado de configuración remote access) es posible establecer una conexión entre nuestro PC anfitrión y RCO-X gracias a la conexión VPN creada como se muestra en la figura 4.1.

```

PS C:\Users\Work> ping 10.54.161.137

Haciendo ping a 10.54.161.137 con 32 bytes de datos:
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=3ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=1ms TTL=63

Estadísticas de ping para 10.54.161.137:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 1ms, Máximo = 3ms, Media = 2ms

```

Figura 4.1: Ping correcto entre PC anfitrión y RCO-X

Usaremos Wireshark para capturar el tráfico que atraviesa la interfaz VMnet8 y comprender mejor lo que sucede durante el ping. La figura 4.2 muestra el resultado de la captura, donde es posible identificar varios tipos de tráfico al examinarla.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.189.129	192.168.189.1	PPP LCP	56	Echo Request
2	0.000014	192.168.189.1	192.168.189.129	PPP LCP	60	Echo Reply
3	0.051819	192.168.189.129	192.168.189.1	GRE	40	Encapsulated PPP
4	0.050677	192.168.189.128	217.71.200.210	NTP	90	NTP Version 4, client
5	0.122758	217.71.200.210	192.168.189.128	NTP	90	NTP Version 4, server
6	0.741988	192.168.189.128	90.68.206.60	NTP	90	NTP Version 4, client
7	0.760985	90.68.206.60	192.168.189.128	NTP	90	NTP Version 4, server
8	4.777411	VMware_C8:00:08	VMware_25:83:bb	ARP	42	Who has 192.168.189.129? Tell 192.168.189.1
9	4.777957	VMware_25:83:bb	VMware_C8:00:08	ARP	42	192.168.189.129 is at 08:50:56:25:83:bb
10	4.891857	VMware_21:35:ca	VMware_C9:1a:00	ARP	42	Who has 192.168.189.2? Tell 192.168.189.128
11	4.892835	VMware_C9:1a:00	VMware_21:35:ca	ARP	42	192.168.189.2 is at 08:50:56:e9:1a:00
12	5.000402	192.168.189.129	192.168.189.1	PPP LCP	56	Echo Request
13	5.001067	192.168.189.1	192.168.189.129	PPP LCP	60	Echo Reply
14	5.052140	192.168.189.129	192.168.189.1	GRE	40	Encapsulated PPP
15	5.242495	10.54.161.201	10.54.161.137	ICMP	107	Echo (ping) request id=0x0001, seq=52/13312, ttl=128 (reply in 16)
16	5.242924	10.54.161.137	10.54.161.201	ICMP	111	Echo (ping) reply id=0x0001, seq=52/13312, ttl=63 (request in 15)
17	5.429608	192.168.189.1	192.168.189.129	GRE	40	Encapsulated PPP
18	6.537341	10.54.161.201	10.54.161.137	ICMP	107	Echo (ping) request id=0x0001, seq=53/13568, ttl=128 (reply in 19)
19	6.539084	10.54.161.137	10.54.161.201	ICMP	111	Echo (ping) reply id=0x0001, seq=53/13568, ttl=63 (request in 18)
20	6.640222	192.168.189.1	192.168.189.129	GRE	40	Encapsulated PPP
21	7.555827	10.54.161.201	10.54.161.137	ICMP	107	Echo (ping) request id=0x0001, seq=54/13824, ttl=128 (reply in 22)
22	7.557362	10.54.161.137	10.54.161.201	ICMP	111	Echo (ping) reply id=0x0001, seq=54/13824, ttl=63 (request in 21)
23	7.562372	192.168.189.1	192.168.189.129	GRE	40	Encapsulated PPP
24	8.562972	10.54.161.201	10.54.161.137	ICMP	107	Echo (ping) request id=0x0001, seq=55/14080, ttl=128 (reply in 25)
25	8.564136	10.54.161.137	10.54.161.201	ICMP	111	Echo (ping) reply id=0x0001, seq=55/14080, ttl=63 (request in 24)
26	8.670540	192.168.189.1	192.168.189.129	GRE	40	Encapsulated PPP
27	8.656093	10.54.161.201	10.54.161.137	DNS	110	Standard query 0x429b A sync-v2.brave.com
28	8.656585	192.168.189.129	192.168.189.2	DNS	77	Standard query 0x429b A sync-v2.brave.com

Figura 4.2: Captura Wireshark VMnet8

Como podemos observar los diferentes tipos de tráfico encontrados son:

1. Paquetes PPP LCP: Como se explica en [7], es un protocolo que se encarga de probar el enlace, realiza un control continuo mediante paquetes de mantenimiento, detectando errores o fallos en la conexión mientras el túnel está activo.
2. Paquetes GRE: Estos paquetes se utilizan para encapsular los datos del protocolo PPP, permitiendo su transporte seguro a través de redes IP. El PAC recibe y procesa estos paquetes GRE, gestionando la comunicación entre el cliente remoto y la red privada, y asegurando la correcta transmisión de datos a través del túnel.
3. Paquetes NTP: Este es un paquete del Protocolo de Tiempo de Red (NTP). El cliente solicita sincronización horaria al servidor NTP, indicando que está configurando su reloj en base al servidor especificado.
4. Paquetes ARP: Este paquete se utiliza para resolver direcciones IP en direcciones MAC dentro de una red local. Cuando un dispositivo quiere comunicarse con otro en la misma red, pero solo conoce la dirección IP de destino, envía un mensaje ARP Who has para preguntar quién posee esa IP. El dispositivo correspondiente

responde con un mensaje ARP Reply que incluye su dirección MAC. Esto permite que los dispositivos puedan construir una tabla ARP y direccionar los paquetes correctamente a nivel de enlace.

5. Paquetes ICMP: Estos paquetes son esenciales para funciones como el comando ping, que envía solicitudes de echo (echo Request) para comprobar la conectividad entre dispositivos y recibe respuestas (echo Reply). También notifican errores, como cuando un host o red es inaccesible, o mensajes relacionados con el tiempo de vida (TTL) de los paquetes.

## 4.2 Prueba 2. Análisis del datagrama original

Basándonos en la captura de Wireshark (Figura 4.2), el datagrama original se encuentra en la línea número 15, correspondiente al primer paquete ICMP de la conexión ping. En la Figura 4.3 se muestra la información en detalle de este datagrama. De esta captura, se puede concluir lo siguiente:

1. El paquete utiliza el protocolo GRE para encapsular un datagrama PPP, lo que indica que la comunicación ICMP se está llevando a cabo a través de un túnel configurado en la red.
2. El datagrama ICMP encapsulado contiene un mensaje Echo Request, parte del comando ping. La dirección de origen es 10.54.161.201 (asignada al PC anfitrión en el túnel PPTP) y la dirección de destino es 10.54.161.137 (asignada a RCO-X).
3. Existen múltiples capas de encapsulación en este paquete: Ethernet, IPv4, GRE, PPP, y finalmente el datagrama ICMP. Esto sugiere un diseño de red donde la comunicación ICMP atraviesa una red interna virtualizada.
4. Dentro del paquete ICMP, se observan campos como el TTL (Time to Live), que tiene un valor de 128, común en sistemas Windows como el PC anfitrión, y el Checksum, un valor calculado para verificar la integridad del paquete. Estos valores confirman que el paquete transmitido busca diagnosticar conectividad y no presenta errores en los datos.

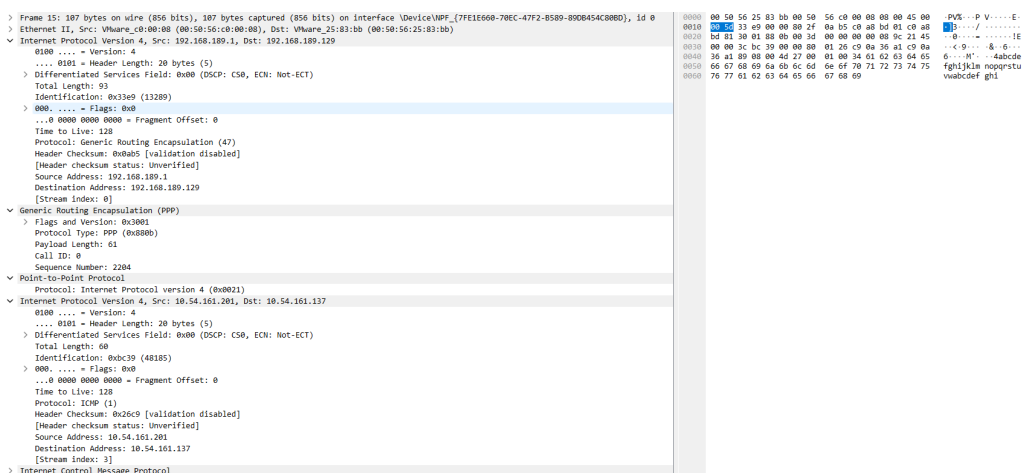


Figura 4.3: Información detallada Datagrama Original

Como se mencionó en el apartado (prueba 1), la figura 4.4 destaca la presencia de echo request asociados al protocolo LCP (Link Control Protocol), los cuales no están relacionados directamente con el comando ping. Según lo especificado en el [7], estos mensajes cumplen funciones clave para mantener la estabilidad de las conexiones PPP.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.189.129	192.168.189.1	PPP LCP	56	Echo Request
2	0.000014	192.168.189.1	192.168.189.129	PPP LCP	46	Version Reply
3	0.051819	192.168.189.129	192.168.189.1	GRE	46	Encapsulated PPP
4	0.096077	192.168.189.128	217.71.200.210	NTP	90	NTP Version 4, client
5	0.122758	217.71.200.210	192.168.189.128	NTP	90	NTP Version 4, server
6	0.741980	192.168.189.128	90.68.206.60	NTP	90	NTP Version 4, client
7	0.769885	90.68.206.60	192.168.189.128	NTP	90	NTP Version 4, server
8	4.777411	Vhware_08:00:00	Vhware_25:83:00	ARP	42	Who has 192.168.189.129? Tell 192.168.189.1
9	4.777957	Vhware_25:83:00	Vhware_08:00:00	ARP	42	192.168.189.129 is at 00:50:56:09:1e:00
10	4.891857	Vhware_21:35:ca	Vhware_e9:1e:00	ARP	42	Who has 192.168.189.2? Tell 192.168.189.128
11	4.892035	Vhware_e9:1e:00	Vhware_21:35:ca	ARP	42	192.168.189.2 is at 00:50:56:09:1e:00
12	5.000492	192.168.189.129	192.168.189.1	PPP LCP	56	Echo Request
13	5.001067	192.168.189.1	192.168.189.129	PPP LCP	60	Echo Reply
14	5.052146	192.168.189.129	192.168.189.1	GRE	46	Encapsulated PPP
15	5.524295	10.54.161.201	10.54.161.137	ICMP	107	Echo (ping) request id=0x0001, seq=52/13312, ttl=128 (reply in 16)
16	5.525024	10.54.161.137	10.54.161.201	ICMP	111	Echo (ping) reply id=0x0001, seq=52/13312, ttl=63 (request in 15)

Figura 4.4: Captura Wireshark Señalando Echo Request extra

En este caso, los echo requests son enviados periódicamente para mantener el túnel activo. Esto evita que el servidor cierre la conexión tras un período de inactividad. Además de su función para evitar desconexiones, estos paquetes también se generan durante las fases iniciales de creación del túnel y en su proceso de cierre, según lo descrito en el RFC. Esto garantiza una gestión eficiente y continua de las conexiones a través del túnel.

### 4.3 Prueba 4. Análisis de los dispositivos creados por el túnel en el cliente

En esta prueba, ejecutamos el comando ipconfig en la máquina anfitriona y verificamos que, después de aplicar la configuración previamente explicada, se añadió un nuevo dispositivo: el Adaptador PPP ddwrt, como se muestra en la Figura 4.5

```
Adaptador PPP ddwrt:

Sufijo DNS específico para la conexión. . . :
Dirección IPv4. . . . . : 10.54.161.201
Máscara de subred . . . . . : 255.255.255.255
Puerta de enlace predeterminada . . . . . :
```

Figura 4.5: ipconfig en el PC anfitrión

Este dispositivo está compuesto por:

1. Sufijo DNS específico para la conexión: En este caso, no se asigna ningún valor, ya que no hemos utilizado un servidor DNS durante la configuración del túnel.
2. Dirección IPv4: 10.54.161.201. Esta es la dirección IP asignada a nuestro dispositivo por el servidor PPP, lo que le permite participar en la red configurada previamente en el router DD-WRT-X, cuya asignación de direcciones abarca el rango 10.54.161.201-209. (asignada a RCO-X).
3. Máscara de subred: 255.255.255.0. Esta máscara corresponde a la subred a la que nuestro dispositivo ha accedido a través del túnel, permitiendo que se comuniquen correctamente los dispositivos dentro de esta red.

4. Puerta de enlace predeterminada: En este caso, no se asigna ningún valor, ya que el equipo anfitrión actúa directamente como cliente del servidor PPP, por lo que no requiere una puerta de enlace adicional.

## 4.4 Prueba 5. Estudio de las tablas de forwarding

Durante la ejecución de esta prueba, analizaremos las tablas de enrutamiento de las máquinas involucradas, que son las siguientes:

1. Máquina anfitriona: En la figura 4.6, la línea marcada en rojo corresponde a la que se utiliza para enrutar el paquete de la orden.

```
PS C:\Users\Work> route print -4
=====
ILista de interfaces
21.....Kaspersky VPN
13...16 13 33 c6 3e 2f .....Microsoft Wi-Fi Direct Virtual Adapter
16...16 13 33 c6 3e 3f .....Microsoft Wi-Fi Direct Virtual Adapter #2
18...00 50 56 c0 00 01 .....VMware Virtual Ethernet Adapter for VMnet1
15...00 50 56 c0 00 08 .....VMware Virtual Ethernet Adapter for VMnet8
66.....ddwrt
17...14 13 33 c6 3e 3f .....MediaTek MT7921 Wi-Fi 6 802.11ax PCIe Adapter
12...14 13 33 c6 3e 3e .....Bluetooth Device (Personal Area Network)
10...5c 60 ba 58 1a 0f .....Realtek Gaming GbE Family Controller
1.....Software Loopback Interface 1
=====

IPv4 Tabla de enrutamiento
=====
Rutas activas:
Destino de red      Máscara de red      Puerta de enlace      Interfaz      Métrica
0.0.0.0             0.0.0.0             192.168.1.1           192.168.1.135 35
10.0.0.0             255.0.0.0           10.54.161.101         10.54.161.201 36
10.24.161.0          255.255.255.0       En vínculo            10.24.161.3    291
10.24.161.3          255.255.255.255     En vínculo            10.24.161.3    291
10.24.161.255        255.255.255.255     En vínculo            10.24.161.3    291
10.54.161.0          255.255.255.0       En vínculo            10.54.161.201 36
10.54.161.201        255.255.255.255     En vínculo            10.54.161.201 291
10.54.161.255        255.255.255.255     En vínculo            10.54.161.201 291
127.0.0.0            255.0.0.0           En vínculo            127.0.0.1      331
127.0.0.1            255.255.255.255     En vínculo            127.0.0.1      331
127.255.255.255      255.255.255.255     En vínculo            127.0.0.1      331
192.168.1.0          255.255.255.0       En vínculo            192.168.1.135 291
192.168.1.135        255.255.255.255     En vínculo            192.168.1.135 291
192.168.1.255        255.255.255.255     En vínculo            192.168.1.135 291
192.168.189.0        255.255.255.0       En vínculo            192.168.189.1 291
192.168.189.1        255.255.255.255     En vínculo            192.168.189.1 291
192.168.189.129      255.255.255.255     En vínculo            192.168.189.1 36
192.168.189.255      255.255.255.255     En vínculo            192.168.189.1 291
224.0.0.0            240.0.0.0           En vínculo            127.0.0.1      331
224.0.0.0            240.0.0.0           En vínculo            10.24.161.3    291
224.0.0.0            240.0.0.0           En vínculo            192.168.189.1 291
224.0.0.0            240.0.0.0           En vínculo            192.168.1.135 291
224.0.0.0            240.0.0.0           En vínculo            10.54.161.201 291
255.255.255.255      255.255.255.255     En vínculo            127.0.0.1      331
255.255.255.255      255.255.255.255     En vínculo            10.24.161.3    291
255.255.255.255      255.255.255.255     En vínculo            192.168.189.1 291
255.255.255.255      255.255.255.255     En vínculo            192.168.1.135 291
255.255.255.255      255.255.255.255     En vínculo            10.54.161.201 291
=====
Rutas persistentes:
Ninguno
```

Figura 4.6: routepint en el PC anfitrión

2. DD-WRT-X: En la figura 4.7, se muestra la tabla de enrutamiento de DD-WRT-X. En este caso, la línea utilizada para enrutar el paquete es la que comienza con la dirección 10.54.161.201, que corresponde a la IP asignada a la máquina anfitriona en la red 10.54...

```
root@DD-WRT:~# ip route
10.54.161.201 dev ppp0 proto kernel scope link src 10.54.161.101
192.168.189.2 dev eth0 scope link
10.54.161.0/24 dev br0 proto kernel scope link src 10.54.161.101
192.168.189.0/24 dev eth0 proto kernel scope link src 192.168.189.129
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 192.168.189.2 dev eth0
```

Figura 4.7: iproute en ddwrt

3. RCO-X: En la figura 4.8, se puede observar la tabla de enrutamiento de la máquina RCO-X, que, en este caso, no tiene ninguna relevancia para el análisis.

```
[root@rco-x ~]# ip route
default via 10.54.161.101 dev ens37 proto dhcp src 10.54.161.137 metric 100
10.54.161.0/24 dev ens37 proto kernel scope link src 10.54.161.137 metric 100
```

Figura 4.8: iproute en RCO-X

Finalmente, ejecutamos el comando 'tracert 10.54.161.137' para observar la ruta que sigue el paquete para alcanzar a RCO-X. La ruta seguida se puede ver en la figura 4.9, y se puede ver como a través de la VPN conecta directamente con el router ddwrt-X y después con RCO-X.

```
PS C:\Users\Work> tracert 10.54.161.137

Traza a la dirección rco-x [10.54.161.137]
sobre un máximo de 30 saltos:

  1      1 ms      1 ms      1 ms  DD-WRT [10.54.161.101]
  2      4 ms      2 ms      2 ms  rco-x [10.54.161.137]

Traza completa.
```

Figura 4.9: tracert desde PC anfitrión a RCO-X

## 4.5 Prueba 6. Comparación de prestaciones y redirección de puertos

### 4.5.1. Comparación de prestaciones

Para este análisis de rendimiento, se llevaron a cabo pruebas de conectividad y rendimiento de red entre el PC anfitrión y un equipo RCO-X, tanto en un entorno sin cifrado como a través de un túnel cifrado.

Para habilitar el cifrado del túnel, es necesario ingresar a la interfaz gráfica del router, navegar a Services >VPN y activar la opción Force MPPE Encryption, como se muestra en la Figura 4.10

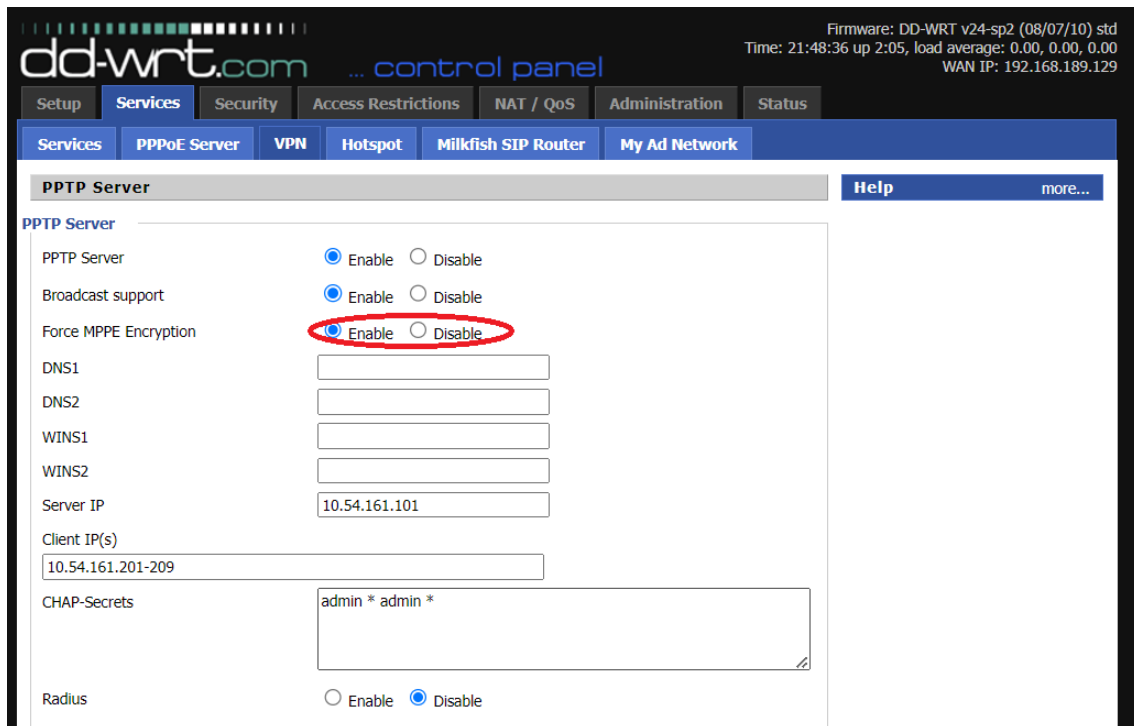


Figura 4.10: Activación de la encriptación del túnel desde la interfaz gráfica del router ddwrt-X

Para garantizar una conexión exitosa, es necesario actualizar las propiedades de seguridad del VPN configurando el cifrado de datos en Requiere cifrado. Esto evita que la conexión sea rechazada al activar la VPN, como se muestra en la Figura 4.11.



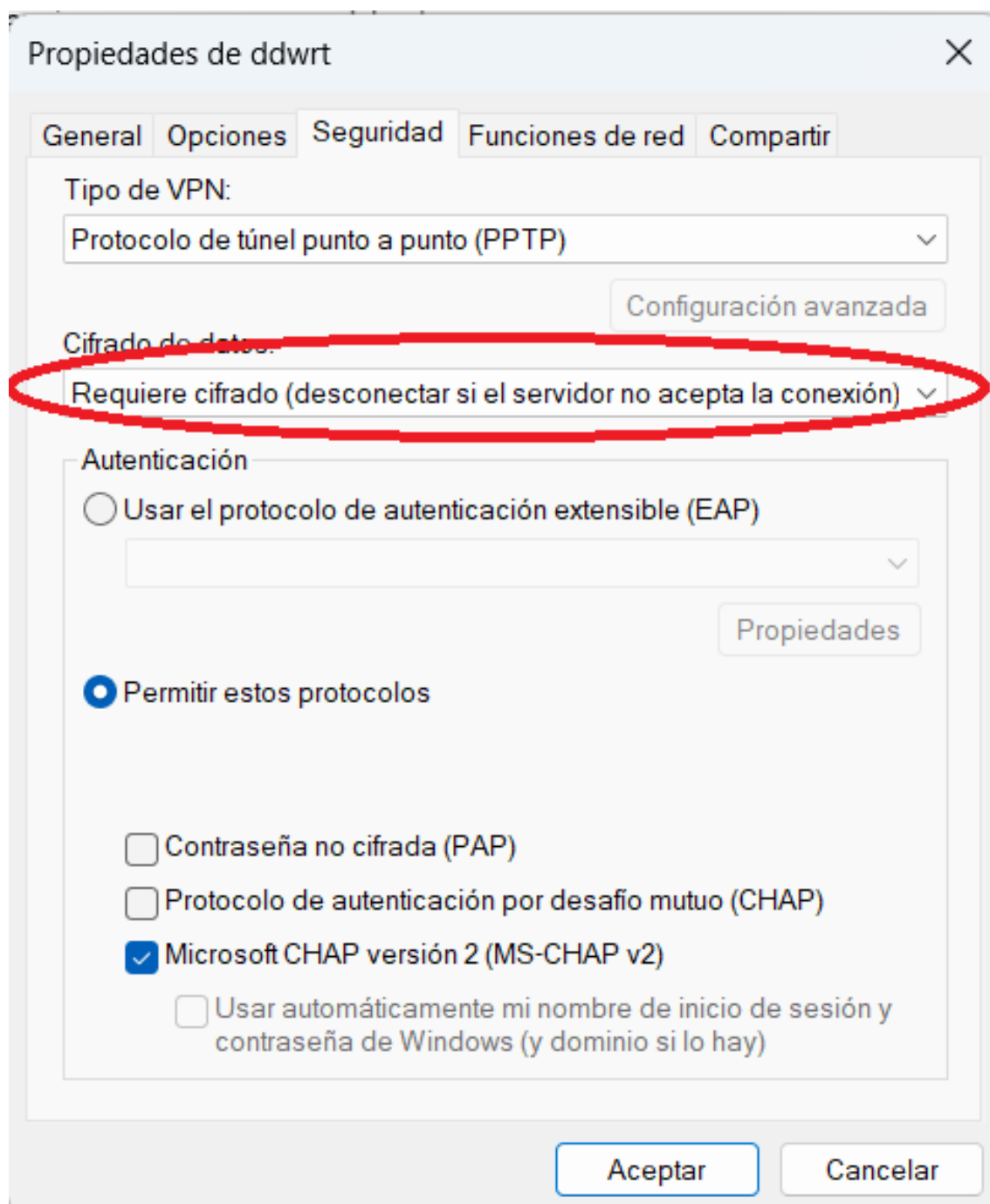


Figura 4.11: Propiedades VPN para permitir conexión encriptada

Las métricas se recopilaron siguiendo las recomendaciones del artículo [8], específicamente en el apartado 1, "Métricas de Rendimiento de Red". Además, se utilizó la herramienta 'iperf3' para medir el ancho de banda, de acuerdo con las indicaciones de [9]. En las figuras 4.12 y 4.13 se presentan los resultados obtenidos al realizar pruebas de conectividad hacia el RCO-X, tanto sin cifrado como con cifrado, respectivamente. Para ello, se utilizó el comando 'ping 10.54.161.137 -n 10', donde '-n 10' especifica el número de paquetes enviados.



```
PS C:\Users\Work> ping 10.54.161.137 -n 10

Haciendo ping a 10.54.161.137 con 32 bytes de datos:
Respuesta desde 10.54.161.137: bytes=32 tiempo=3ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=3ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=3ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=3ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=3ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63

Estadísticas de ping para 10.54.161.137:
    Paquetes: enviados = 10, recibidos = 10, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 2ms, Máximo = 3ms, Media = 2ms
```

Figura 4.12: Ping desde PC anfitrión a RCO-X sin encriptar

```
PS C:\Users\Work> ping 10.54.161.137 -n 10

Haciendo ping a 10.54.161.137 con 32 bytes de datos:
Respuesta desde 10.54.161.137: bytes=32 tiempo=7ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=3ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=1ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63

Estadísticas de ping para 10.54.161.137:
    Paquetes: enviados = 10, recibidos = 10, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 1ms, Máximo = 7ms, Media = 2ms
```

Figura 4.13: Ping desde PC anfitrión a RCO-X encriptado

Además, se evaluó el rendimiento desde dos perspectivas, la transferencia de datos desde el PC hacia RCO-X y viceversa utilizando 'iperf3', mostrando en las figuras 4.14 y 4.15 los resultados del comando sin encriptar y las figuras 4.16 y 4.17 los resultados estando encriptado:

```
[root@rco-x ~]# iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 10.54.161.101, port 60621
[ 5] local 10.54.161.137 port 5201 connected to 10.54.161.101 port 60622
[ ID] Interval            Transfer        Bitrate
[ 5] 0.00-1.32      sec    1.25 MBytes    7.94 Mbits/sec
[ 5] 1.32-2.36      sec    1.62 MBytes    13.1 Mbits/sec
[ 5] 2.36-3.35      sec    1.50 MBytes    12.8 Mbits/sec
[ 5] 3.35-4.09      sec    896 KBytes     9.90 Mbits/sec
[ 5] 4.09-5.12      sec    1.38 MBytes    11.2 Mbits/sec
[ 5] 5.12-6.08      sec    1.50 MBytes    13.1 Mbits/sec
[ 5] 6.08-7.08      sec    1.62 MBytes    13.6 Mbits/sec
[ 5] 7.08-8.11      sec    1.00 MBytes     8.13 Mbits/sec
[ 5] 8.11-9.25      sec    1.25 MBytes     9.18 Mbits/sec
[ 5] 9.25-10.10     sec    2.00 MBytes    19.8 Mbits/sec
-----
[ ID] Interval            Transfer        Bitrate
[ 5] 0.00-10.10     sec    14.0 MBytes    11.6 Mbits/sec
receiver
```

Figura 4.14: iperf3 desde RCO-X sin encriptar

```
PS C:\Users\Work\Desktop\iperf3.17.1_64_updatedcygwin\iperf3.17.1_updatedcygwin> .\iperf3.exe -c 10.54.161.137
Connecting to host 10.54.161.137, port 5201
[ 5] local 10.54.161.201 port 60622 connected to 10.54.161.137 port 5201
[ ID] Interval            Transfer        Bitrate
[ 5] 0.00-1.00      sec    1.25 MBytes    10.5 Mbits/sec
[ 5] 1.00-2.00      sec    1.62 MBytes    13.6 Mbits/sec
[ 5] 2.00-3.01      sec    1.50 MBytes    12.6 Mbits/sec
[ 5] 3.01-4.00      sec    1.00 MBytes     8.43 Mbits/sec
[ 5] 4.00-5.00      sec    1.25 MBytes    10.4 Mbits/sec
[ 5] 5.00-6.01      sec    1.62 MBytes    13.5 Mbits/sec
[ 5] 6.01-7.01      sec    1.62 MBytes    13.6 Mbits/sec
[ 5] 7.01-8.01      sec    896 KBytes     7.31 Mbits/sec
[ 5] 8.01-9.00      sec    1.25 MBytes    10.6 Mbits/sec
[ 5] 9.00-10.00     sec    2.12 MBytes    17.9 Mbits/sec
-----
[ ID] Interval            Transfer        Bitrate
[ 5] 0.00-10.00     sec    14.1 MBytes    11.8 Mbits/sec
[ 5] 0.00-10.10     sec    14.0 MBytes    11.6 Mbits/sec
sender
receiver
iperf Done.
```

Figura 4.15: iperf3 desde PC anfitrión sin encriptar

```
[root@rco-x ~]# iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 10.54.161.101, port 65447
[ 5] local 10.54.161.137 port 5201 connected to 10.54.161.101 port 65448
[ ID] Interval            Transfer        Bitrate
[ 5]  0.00-1.41      sec    1.50 MBytes    8.90 Mbits/sec
[ 5]  1.41-2.10      sec     768 KBytes    9.14 Mbits/sec
[ 5]  2.10-3.28      sec    1.12 MBytes    8.02 Mbits/sec
[ 5]  3.28-4.09      sec    1.12 MBytes   11.6 Mbits/sec
[ 5]  4.09-5.42      sec    2.00 MBytes   12.7 Mbits/sec
[ 5]  5.42-6.11      sec     896 KBytes   10.7 Mbits/sec
[ 5]  6.11-7.30      sec    1.88 MBytes   13.2 Mbits/sec
[ 5]  7.30-8.38      sec    1.00 MBytes    7.77 Mbits/sec
[ 5]  8.38-9.08      sec    1.62 MBytes   19.4 Mbits/sec
[ 5]  9.08-10.07     sec     768 KBytes    6.33 Mbits/sec
-----
[ ID] Interval            Transfer        Bitrate
[ 5]  0.00-10.07     sec   12.6 MBytes   10.5 Mbits/sec
receiver
```

Figura 4.16: iperf3 desde RCO-X encriptado

```
PS C:\Users\Work\Desktop\iperf3.17.1_64_updatedcygwin\iperf3.17.1_updatedcygwin> .\iperf3.exe -c 10.54.161.137
Connecting to host 10.54.161.137, port 5201
[ 5] local 10.54.161.201 port 65448 connected to 10.54.161.137 port 5201
[ ID] Interval            Transfer        Bitrate
[ 5]  0.00-1.00      sec    1.50 MBytes   12.6 Mbits/sec
[ 5]  1.00-2.01      sec     768 KBytes    6.27 Mbits/sec
[ 5]  2.01-3.01      sec    1.12 MBytes    9.39 Mbits/sec
[ 5]  3.01-4.01      sec    1.25 MBytes   10.4 Mbits/sec
[ 5]  4.01-5.01      sec    1.88 MBytes   15.8 Mbits/sec
[ 5]  5.01-6.01      sec     896 KBytes    7.34 Mbits/sec
[ 5]  6.01-7.01      sec    1.88 MBytes   15.8 Mbits/sec
[ 5]  7.01-8.02      sec    1.12 MBytes    9.40 Mbits/sec
[ 5]  8.02-9.00      sec    1.50 MBytes   12.8 Mbits/sec
[ 5]  9.00-10.00     sec     896 KBytes    7.32 Mbits/sec
-----
[ ID] Interval            Transfer        Bitrate
[ 5]  0.00-10.00     sec   12.8 MBytes   10.7 Mbits/sec
[ 5]  0.00-10.07     sec   12.6 MBytes   10.5 Mbits/sec
sender
receiver
iperf Done.
```

Figura 4.17: iperf3 desde PC anfitrión encriptado

Mostrando el siguiente impacto en las Prestaciones:

- Latencia: Aunque la media del tiempo de respuesta no cambió significativamente, el túnel cifrado introdujo picos ocasionales más altos.
- Ancho de Banda: Se observó una disminución en el 'bitrate' promedio del 9-10 por ciento cuando se activó el cifrado, lo que es coherente con la sobrecarga inherente al cifrado/descifrado.
- Estabilidad: El entorno sin cifrado presentó un rendimiento más estable, mientras que el cifrado introdujo mayor variabilidad en ciertos intervalos.

Las pruebas muestran que la activación del cifrado tiene un impacto moderado en el rendimiento de la red, reduciendo ligeramente el ancho de banda promedio y aumentan-

do ocasionalmente la latencia máxima. Sin embargo, estas diferencias son esperables y justificables en entornos donde la seguridad es prioritaria.

Para escenarios críticos donde el rendimiento óptimo sea más relevante que la seguridad, podría ser adecuado considerar el uso de túneles sin cifrado. Sin embargo, en contextos donde la confidencialidad de los datos es esencial, el cifrado proporciona un balance razonable entre seguridad y rendimiento.

#### 4.5.2. Redirección de Puertos

Para aplicar la redirección de puertos hay que acceder a la interfaz gráfica del router ddwrt-C y acceder a NAT/QoS>Port Forwarding. Como se muestra en la Figura 4.18, se hizo lo siguiente:

1. Añadir nueva regla: que redirige el puerto 22222 del protocolo TCP hacia el puerto 22 de la dirección IP interna '10.54.161.101'. Esta configuración permite que las conexiones externas realizadas al puerto 22222 en las IP públicas del router sean dirigidas al puerto 22 (generalmente utilizado por ssh).
2. Desactivar la primera regla. Esto evita que se pueda acceder por el puerto 1723, es decir, deshabilita el acceso al servicio PPTP desde el exterior.



Figura 4.18: Redirección en la Interfaz Gráfica de ddwrt-X

Pudiendo probar mediante SSH la conexión a RCO a través del puerto público 22222 como se muestra en la Figura 4.19

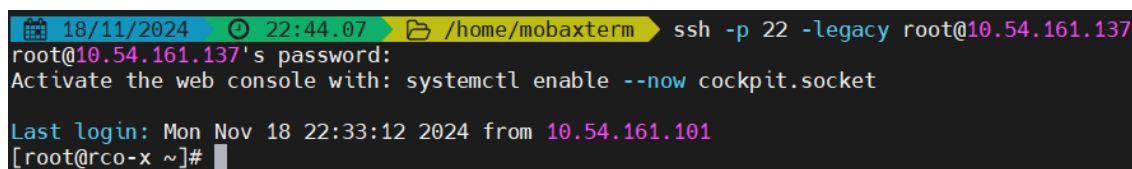


Figura 4.19: Ping desde PC anfitrión a RCO-X usando el puerto 22222

Una vez realizada la redirección de puertos, se presenta un análisis comparativo de las métricas de rendimiento de red para los datos encriptados y sin encriptar utilizando las herramientas 'iperf3' y 'ping' al igual que se hizo en el apartado anterior.

En las figuras 4.20 y 4.21 se presentan los resultados obtenidos al realizar pruebas de conectividad hacia el RCO-X, tanto sin cifrado como con cifrado, respectivamente.

```
PS C:\Users\Work> ping 10.54.161.137 -n 10

Haciendo ping a 10.54.161.137 con 32 bytes de datos:
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=1ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=1ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=4ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63

Estadísticas de ping para 10.54.161.137:
    Paquetes: enviados = 10, recibidos = 10, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 1ms, Máximo = 4ms, Media = 2ms
```

Figura 4.20: Ping desde PC anfitrión a RCO-X sin encriptar

```
PS C:\Users\Work> ping 10.54.161.137 -n 10

Haciendo ping a 10.54.161.137 con 32 bytes de datos:
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=3ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=1ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=1ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63
Respuesta desde 10.54.161.137: bytes=32 tiempo=2ms TTL=63

Estadísticas de ping para 10.54.161.137:
    Paquetes: enviados = 10, recibidos = 10, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 1ms, Máximo = 3ms, Media = 1ms
```

Figura 4.21: Ping desde PC anfitrión a RCO-X encriptado

Además, se evaluó el rendimiento desde dos perspectivas, la transferencia de datos desde el PC hacia RCO-X y viceversa utilizando 'iperf3', mostrando en las figuras 4.22 y 4.23 los resultados del comando sin encriptar y las figuras 4.24 y 4.25 los resultados estando encriptado:

```
[root@rco-x ~]# iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 10.54.161.101, port 54380
[ 5] local 10.54.161.137 port 5201 connected to 10.54.161.101 port 54381
[ ID] Interval           Transfer     Bitrate
[ 5]  0.00-1.43       sec   1.62 MBytes  9.54 Mbits/sec
[ 5]  1.43-2.71       sec   1.38 MBytes  8.97 Mbits/sec
[ 5]  2.71-3.11       sec   896 KBytes  18.5 Mbits/sec
[ 5]  3.11-4.07       sec   1.25 MBytes  10.9 Mbits/sec
[ 5]  4.07-5.07       sec   1.50 MBytes  12.6 Mbits/sec
[ 5]  5.07-6.10       sec   1.62 MBytes  13.2 Mbits/sec
[ 5]  6.10-7.32       sec   1.25 MBytes  8.58 Mbits/sec
[ 5]  7.32-8.26       sec   1.38 MBytes  12.2 Mbits/sec
[ 5]  8.26-9.43       sec   1.00 MBytes  7.20 Mbits/sec
[ 5]  9.43-10.30      sec   1.25 MBytes  12.0 Mbits/sec
-----
[ ID] Interval           Transfer     Bitrate
[ 5]  0.00-10.30      sec  13.1 MBytes  10.7 Mbits/sec
receiver
```

Figura 4.22: iperf3 desde RCO-X sin encriptar

```
PS C:\Users\Work\Desktop\iperf3.17.1_64_updatedcygwin\iperf3.17.1_updatedcygwin> .\iperf3.exe -c 10.54.161.137
Connecting to host 10.54.161.137, port 5201
[ 5] local 10.54.161.201 port 54381 connected to 10.54.161.137 port 5201
[ ID] Interval           Transfer     Bitrate
[ 5]  0.00-1.01       sec   1.62 MBytes  13.6 Mbits/sec
[ 5]  1.01-2.01       sec   1.38 MBytes  11.5 Mbits/sec
[ 5]  2.01-3.00       sec   896 KBytes  7.40 Mbits/sec
[ 5]  3.00-4.01       sec   1.38 MBytes  11.4 Mbits/sec
[ 5]  4.01-5.01       sec   1.50 MBytes  12.6 Mbits/sec
[ 5]  5.01-6.01       sec   1.50 MBytes  12.6 Mbits/sec
[ 5]  6.01-7.00       sec   1.25 MBytes  10.5 Mbits/sec
[ 5]  7.00-8.00       sec   1.38 MBytes  11.5 Mbits/sec
[ 5]  8.00-9.01       sec   1.12 MBytes  9.33 Mbits/sec
[ 5]  9.01-10.00      sec   1.25 MBytes  10.6 Mbits/sec
-----
[ ID] Interval           Transfer     Bitrate
[ 5]  0.00-10.00      sec  13.2 MBytes  11.1 Mbits/sec
[ 5]  0.00-10.30      sec  13.1 MBytes  10.7 Mbits/sec
sender
receiver
iperf Done.
```

Figura 4.23: iperf3 desde PC anfitrión sin encriptar



```
[root@rco-x ~]# iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 10.54.161.101, port 52500
[ 5] local 10.54.161.137 port 5201 connected to 10.54.161.101 port 52501
[ ID] Interval            Transfer           Bitrate
[ 5]  0.00-1.24          sec    512 KBytes      3.38 Mbits/sec
[ 5]  1.24-2.07          sec    1.75 MBytes     17.7 Mbits/sec
[ 5]  2.07-3.40          sec    1.62 MBytes     10.2 Mbits/sec
[ 5]  3.40-4.08          sec    1.50 MBytes     18.6 Mbits/sec
[ 5]  4.08-5.07          sec    1.50 MBytes     12.7 Mbits/sec
[ 5]  5.07-6.06          sec    1.50 MBytes     12.7 Mbits/sec
[ 5]  6.06-7.14          sec    1.62 MBytes     12.6 Mbits/sec
[ 5]  7.14-8.09          sec    768 KBytes      6.61 Mbits/sec
[ 5]  8.09-9.33          sec    2.00 MBytes     13.6 Mbits/sec
[ 5]  9.33-10.38         sec    896 KBytes      6.95 Mbits/sec
-----
[ ID] Interval            Transfer           Bitrate
[ 5]  0.00-10.38         sec   13.6 MBytes     11.0 Mbits/sec
receiver
```

Figura 4.24: iperf3 desde RCO-X encriptado

```
PS C:\Users\Work\Desktop\iperf3.17.1_64_updatedcygwin\iperf3.17.1_updatedcygwin> .\iperf3.exe -c 10.54.161.137
Connecting to host 10.54.161.137, port 5201
[ 5] local 10.54.161.201 port 52501 connected to 10.54.161.137 port 5201
[ ID] Interval            Transfer           Bitrate
[ 5]  0.00-1.01          sec    512 KBytes      4.15 Mbits/sec
[ 5]  1.01-2.01          sec    1.88 MBytes     15.7 Mbits/sec
[ 5]  2.01-3.01          sec    1.50 MBytes     12.6 Mbits/sec
[ 5]  3.01-4.01          sec    1.62 MBytes     13.6 Mbits/sec
[ 5]  4.01-5.00          sec    1.50 MBytes     12.7 Mbits/sec
[ 5]  5.00-6.01          sec    1.50 MBytes     12.5 Mbits/sec
[ 5]  6.01-7.01          sec    1.50 MBytes     12.6 Mbits/sec
[ 5]  7.01-8.00          sec    896 KBytes      7.39 Mbits/sec
[ 5]  8.00-9.00          sec    1.88 MBytes     15.8 Mbits/sec
[ 5]  9.00-10.01         sec    1.00 MBytes      8.31 Mbits/sec
-----
[ ID] Interval            Transfer           Bitrate
[ 5]  0.00-10.01         sec    13.8 MBytes     11.5 Mbits/sec
[ 5]  0.00-10.38         sec    13.6 MBytes     11.0 Mbits/sec
sender
receiver
iperf Done.
```

Figura 4.25: iperf3 desde PC anfitrión encriptado

El análisis muestra que el impacto del cifrado en el rendimiento de la red es mínimo:

- Ancho de banda: Las velocidades promedio entre ambas configuraciones son casi idénticas (alrededor de 11 Mbps).
- Latencia: Los datos encriptados presentaron un tiempo de respuesta promedio más bajo (1 ms), lo que podría ser circunstancial.

Ambas configuraciones ofrecen un desempeño adecuado para tareas regulares, pero el uso de cifrado añade una capa de seguridad sin comprometer significativamente el rendimiento

#### 4.5.3. Comparación prestaciones con y sin redirección

- Ancho de Banda
  - Las velocidades promedio han mostrado una consistencia notable en todas las pruebas, con diferencias menores (<0.5 Mbps).

- En las pruebas actuales, tanto encriptado como sin encriptar, se observaron picos de velocidad más altos, indicando un mejor manejo de intervalos de transmisión.
- Latencia
  - La latencia promedio de las pruebas encriptadas mejoró levemente (1 ms frente a 2 ms), mientras que los datos sin encriptar mantuvieron valores similares.

En resumen, ambas series de pruebas reflejan un comportamiento estable de la red, con diferencias mínimas entre configuraciones y entre pruebas realizadas en diferentes momentos. Esto respalda la idea de que el cifrado tiene un impacto prácticamente imperceptible en el rendimiento global.



---

## CAPÍTULO 5

# Conclusiones

---

Después de una exploración en profundidad de los túneles PPTP, llegamos a algunas conclusiones interesantes. Aunque este protocolo alguna vez fue una opción popular para crear una red segura, ahora ha sido reemplazado por alternativas más sólidas y confiables. Su principal debilidad son los problemas de seguridad que causa, lo que lo hace menos adecuado para entornos de misión crítica.

Nuestras pruebas muestran que PPTP tiene muchas ventajas, como la facilidad de uso y la velocidad de implementación. Sin embargo, carece de las capacidades de seguridad avanzadas que requieren los entornos modernos. Entonces, si la seguridad no es una prioridad, el protocolo puede ser una opción funcional. En el panorama actual, PPTP ha sido reemplazado en gran medida por protocolos más modernos y seguros, como IPsec y OpenVPN. Sin embargo, sigue siendo útil en casos concretos donde la simplicidad es fundamental y no se requiere la máxima seguridad. Sin embargo, su uso ha disminuido significativamente debido a la disponibilidad de soluciones más robustas y seguras.



# Bibliografía

---

- [1] Jorge Prieto Cancino. *Implementación de accesos remotos utilizando VPN*. PhD thesis, Universidad Católica del Maule, Facultad de Ciencias de la Ingeniería, 2005.
- [2] William Simpson. The point-to-point protocol (ppp). Technical report, 1993.
- [3] Jon C Snader. *VPNs Illustrated: Tunnels, VPNs, and IPsec: Tunnels, VPNs, and IPsec*. Addison-Wesley Professional, 2015.
- [4] Wireshark. Wireshark, 2020. Disponible en <https://wiki.wireshark.org/NTP>, consultada el 18/11/2024.
- [5] Yúbal Fernández. Tracert o traceroute: qué es, cómo funciona o cómo se utiliza, 2020. Disponible en <https://www.xataka.com/basics/tracert-traceroute-que-como-funciona-como-se-utiliza>, consultada el 18/11/2024.
- [6] J. Taarud W. Little K. Hamzeh, W. Verthein and G.Zorn. Point-to-point tunneling protocol (pptp), 1999. Disponible en <https://datatracker.ietf.org/doc/html/rfc2637>.
- [7] W. Simpson. The point-to-point protocol (ppp), 1994. Disponible en <https://datatracker.ietf.org/doc/html/rfc1661>.
- [8] Linkedin. What are the best ways to measure data center network architecture performance and efficiency?, 2024. Disponible en <https://es.linkedin.com/advice/0/what-best-ways-measure-data-center-network?lang=en>.
- [9] Redes Zone. iperf3-medir-velocidad-lan-wifi-internet, 2024. Disponible en <https://www.redeszone.net/tutoriales/redes-cable/iperf3-medir-velocidad-lan-wifi-internet/>.