Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

PPTP

Trabajo Redes Corporativas

Grado en Ingeniería Informática

Autores: Adrián Carrillo y Vicente Arnau

Router: 17

2019/2020

Resumen

PPTP es un protocolo que pese a estar obsoleto, permite implementar VPN entre equipos vía internet. En concreto vamos a apoyarnos en el firmware DD-WRT que permite la configuración de un túnel de este tipo. Además, veremos su funcionamiento, así como la forma que tienen los mensajes a la hora de ser transmitidos por la red.

Palabras clave: pptp, túnel, configuración, dd-wrt.

Abstract

PPTP is a protocol that despite being obsolete, allows VPN to be implemented between computers via the Internet. Specifically, we will rely on the DD-WRT firmware that allows the configuration of such a tunnel. We will also see how they work, as well as the way messages have when they are transmitted over the network.

Keywords : pptp, tunnel, configuration, dd-wrt.

Tabla de contenidos

1. Introducción

El uso de redes privadas virtuales (VPN[1]) es muy extendido actualmente por todo el globo debido a la facilidad de su uso y a su flexibilidad. El poder acceder a una red completamente externa a la que tenemos en casa se ha convertido en muchos lugares en algo completamente habitual y necesario si se requiere de conexión a una red ajena.

No hace mucho, este tipo de conexión se realizaba punto a punto, utilizando la línea telefónica para el intercambio de comunicaciones. Esta solución es, probablemente, una de las más seguras, ya que se dispone de una línea dedicada, pero por la misma parte, también es una solución cara, ya que es necesario que la red destino (normalmente empresarial) disponga de un RAS[2] que proporcione al usuario una IP interna. [1]

Como solución a este problema surgió PPTP[3]. [2] PPTP es un protocolo de comunicaciones ya obsoleto desarrollado por un consorcio de empresas, entre ellas Microsoft, Ascend Communications y 3Com. Provee encapsulamiento a paquetes punto a punto y usa TCP[4] como control de conexión.

Estos avances han permitido emular el funcionamiento de un RAS sobre internet, dividiéndolo en dos partes. Por un lado, el proveedor de servicios de internet del cliente provee la estructura para la creación de un túnel, llamado PAC (*PPTP Access concentrator*) y por otro, la empresa destino mantiene un servidor de red conocido como PNS (*PPTP Network server*). Debido a estos dos elementos, a este tipo de modelo se le conoce como modelo PAC-PNS. [3]

En el caso que se expone, hemos conseguido crear dos tipos de túnel PPTP que realizan en esencia la misma función pero que son muy diferentes. En el primer caso, hemos creado un túnel entre dos enrutadores que permite acceder a los miembros de una red interna a otra ajena a esta. En el segundo, el túnel creado sirve directamente a una máquina cliente y le permite conectarse a la red interna objetivo. Este método ignora todas las demás máquinas, aunque se encuentren dentro de la misma red que el cliente, ya que la conexión se realiza de manera privada.

[1] Siglas en inglés de *virtual private network*.
[2] Siglas en inglés de *remote access server*.
[3] Siglas en inglés de *point-to-point tunnel protocol*.
[4] Siglas en inglés de t*ransmission control protocol*.

La red sobre la que partimos es la que se indica en la siguiente figura:

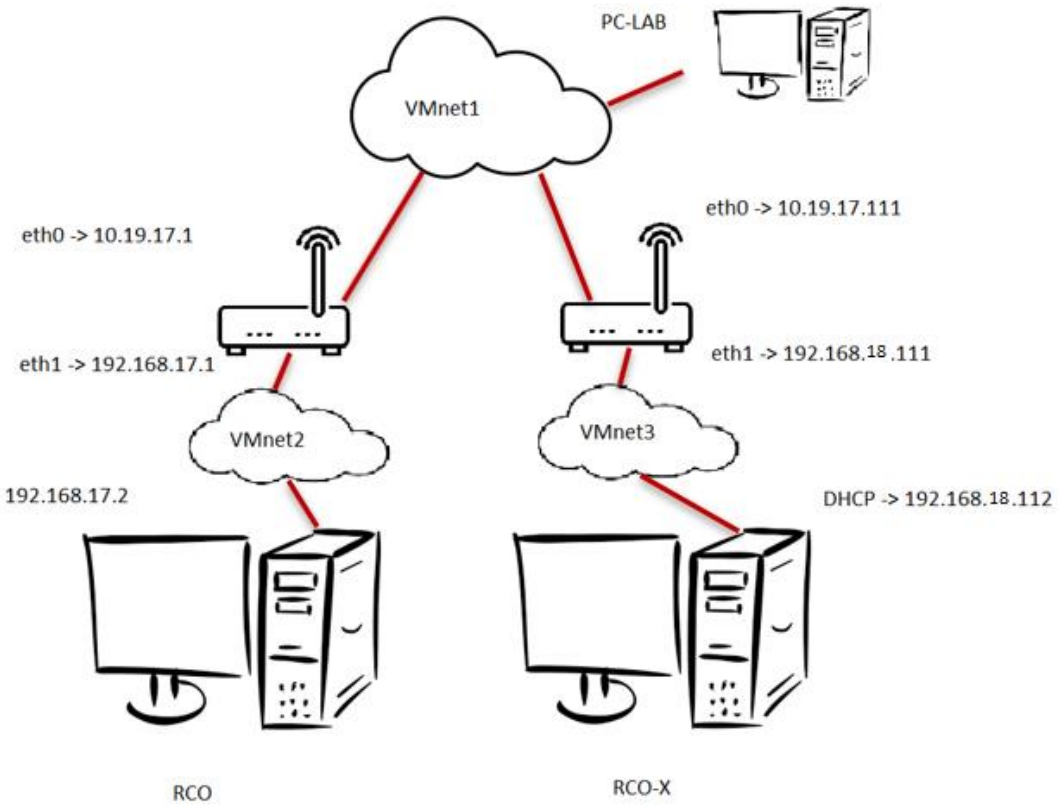
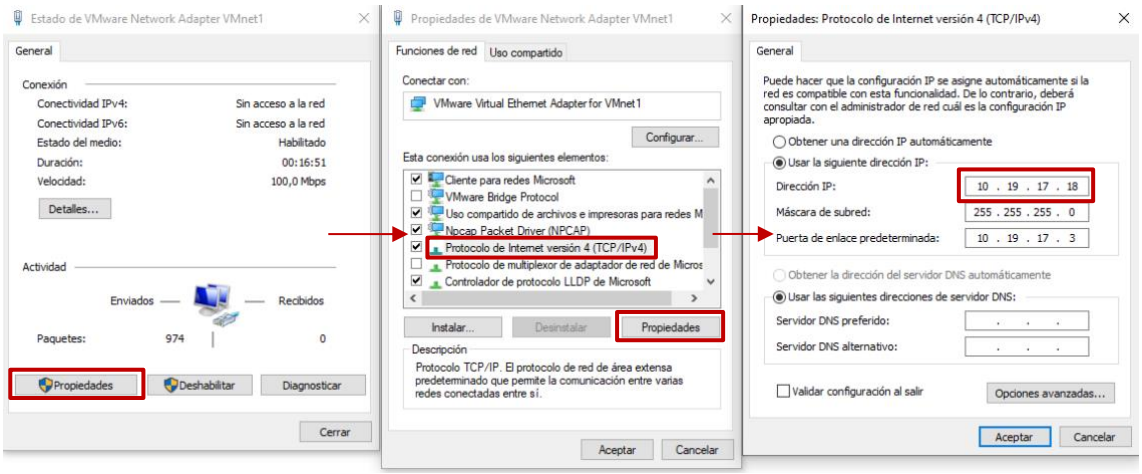


Figura 1: Red inicial

2. Configuración

Partimos con cuatro máquinas virtuales. Estas cuatro máquinas se dividen en dos enrutadores y dos clientes; los primeros los nombraremos `ddwrt-X` y `ddwrt-noX` y los segundos, `RCO-X` y `RCO`. Tal y como se muestra en el capítulo 1 (fig. 1), `ddwrt-X` sirve a la máquina `RCO-X` y `ddwrt-noX` sirve a `RCO`.

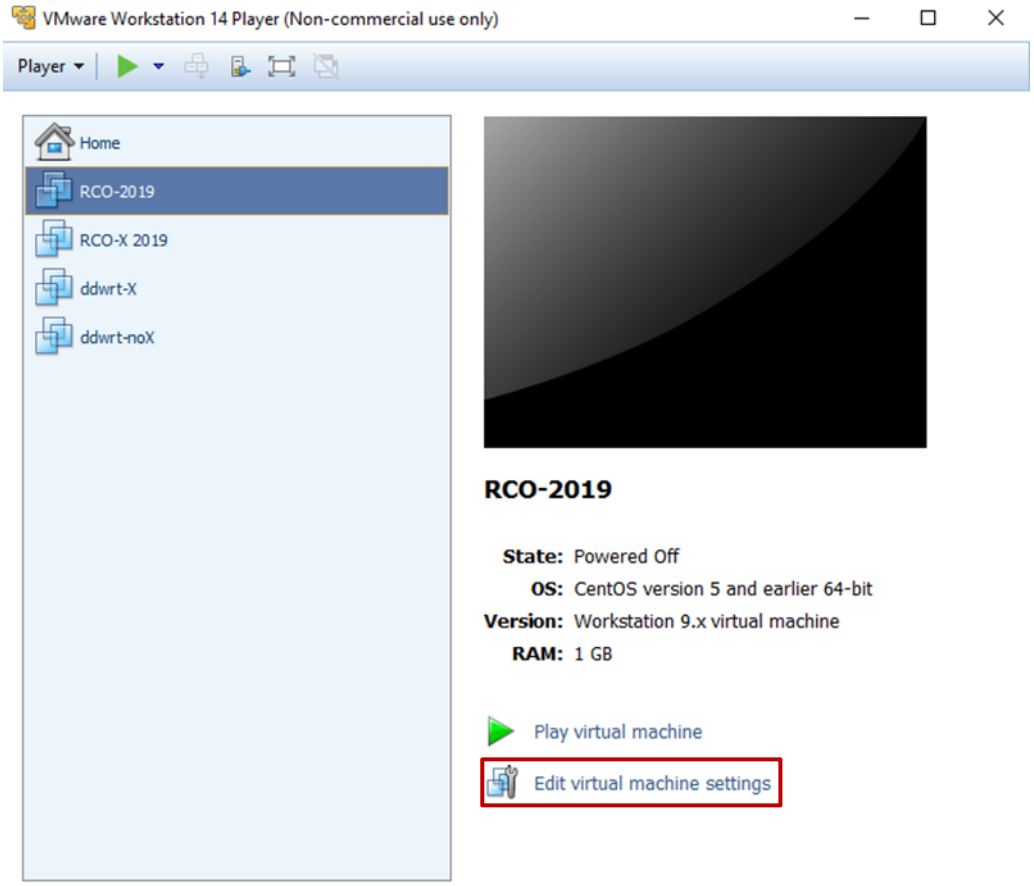
Por un lado, configuramos la máquina del laboratorio para que tuviera acceso a la red virtual (a partir de este momento, PC) que creamos. Para ello, hemos cambiado la dirección IP de nuestro de nuestro PC a una coherente para situar a esta máquina dentro de la red virtual. [4] En el caso que se expone, esta IP es `10.19.17.18`.



Comprobamos que la IP se configuró correctamente con el comando `ipconfig`:

```
Adaptador de Ethernet VMware Network Adapter VMnet1:

   Sufijo DNS específico para la conexión. . :
   Vínculo: dirección IPv6 local. . . : fe80::b8d9:5091:dd49:a813%3
   Dirección IPv4. . . . . . . . . . . . : 10.19.17.18
   Máscara de subred . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 10.19.17.3
```

En otro lugar, la configuración de las máquinas virtuales se realizó directamente desde la propia aplicación VMware Workstation 14 Player[5]. La interfaz de la aplicación ofrece las herramientas necesarias para modificar y adaptar el hardware virtualizado.

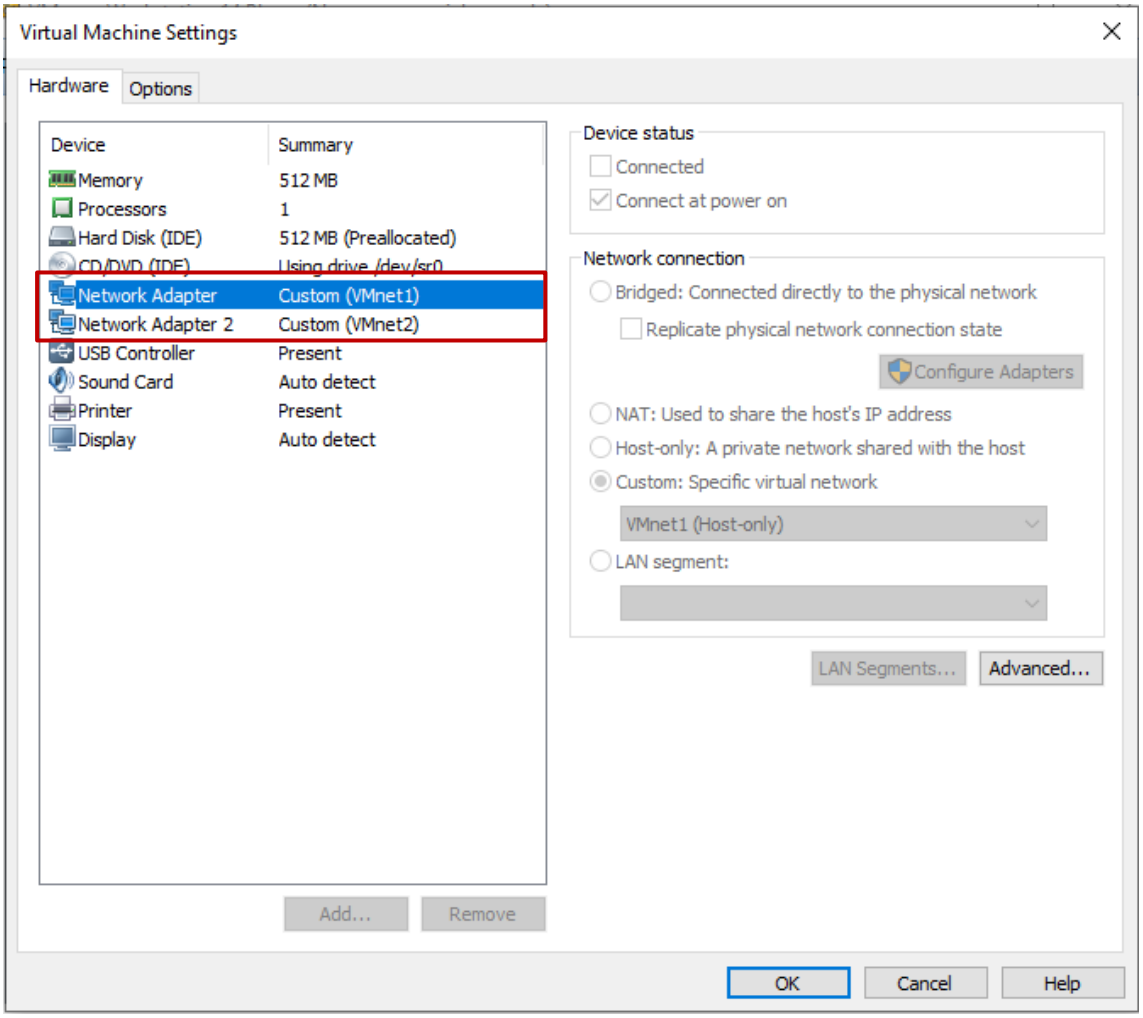


En las máquinas virtuales no era necesario modificar las direcciones MAC, pues no hubo ningún tipo de conflicto al no estar conectadas a la red de la UPV; estas se dejaron por defecto, generadas automáticamente.

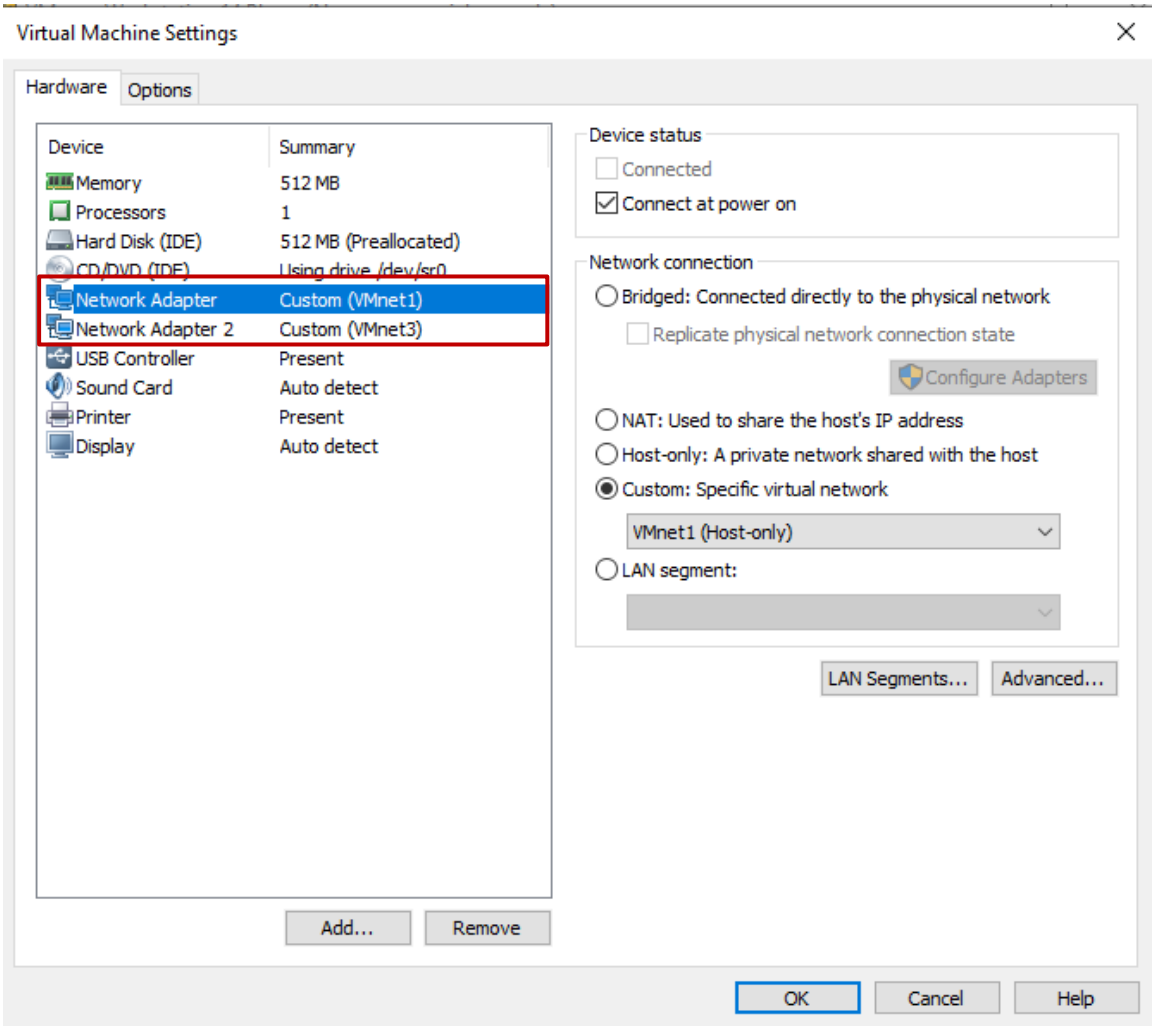
Para que ambos enrutadores tuvieran conexión directa al PC era necesario configurar el adaptador de red principal de cada uno de ellos. Por defecto esta configuración estaba marcada como *bridged* y fue necesario cambiarla a *custom* para seleccionar libremente la red a la que se quiere conectar. [5] La red que se seleccionó fue `VMnet1`, pues es la ofrecida por VMware para tener una red Ethernet privada que conecta la máquina virtual y la máquina anfitriona. [6]

[5] www.vmware.com

En `ddwrt-noX` el adaptador de red principal se configuró para usar VMnet1 y el secundario se conectó a `VMnet2`:

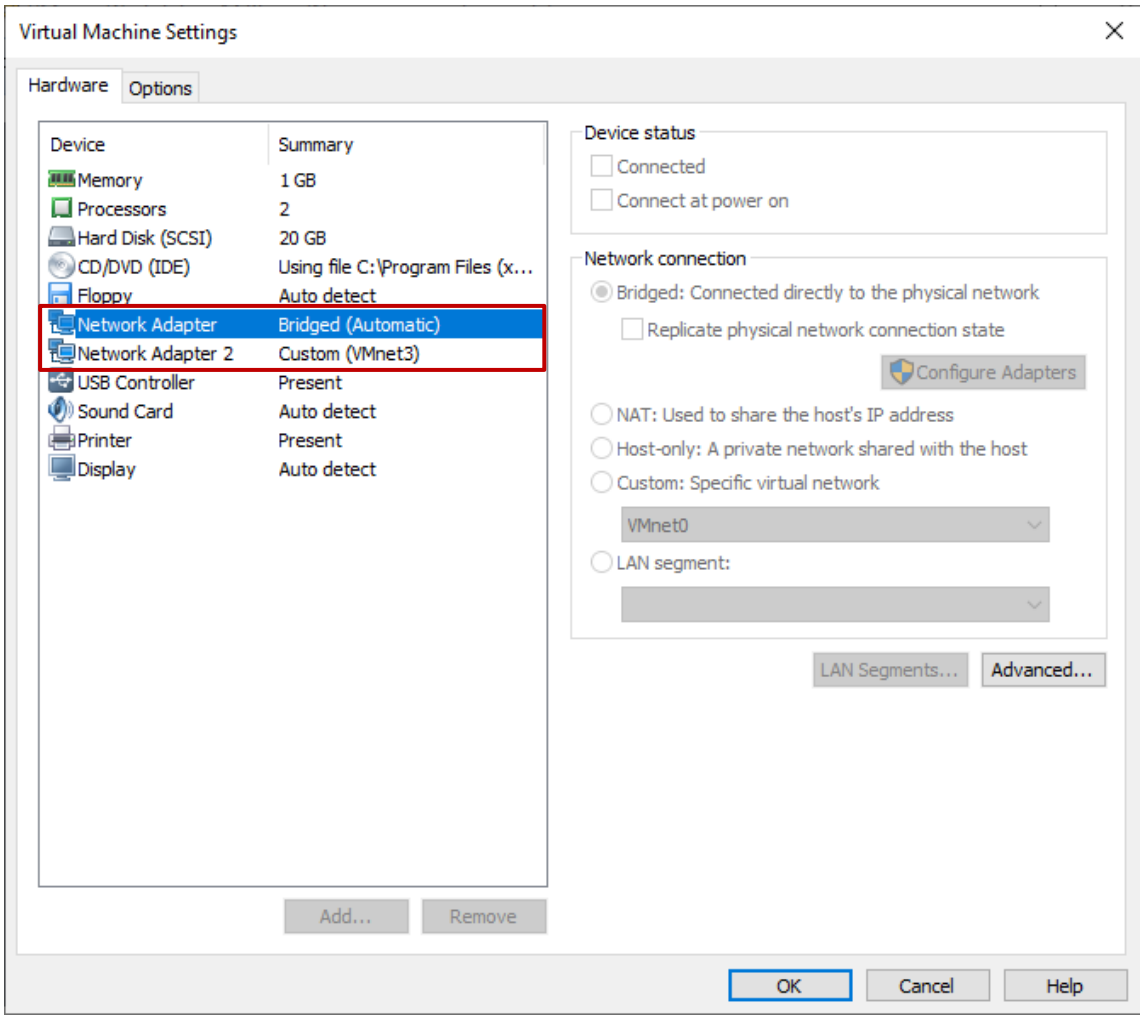


En `ddwrt-X` el adaptador de red principal se configuró de manera análoga al primer enrutador, mientras que su adaptador secundario fue conectado a `VMnet3`:

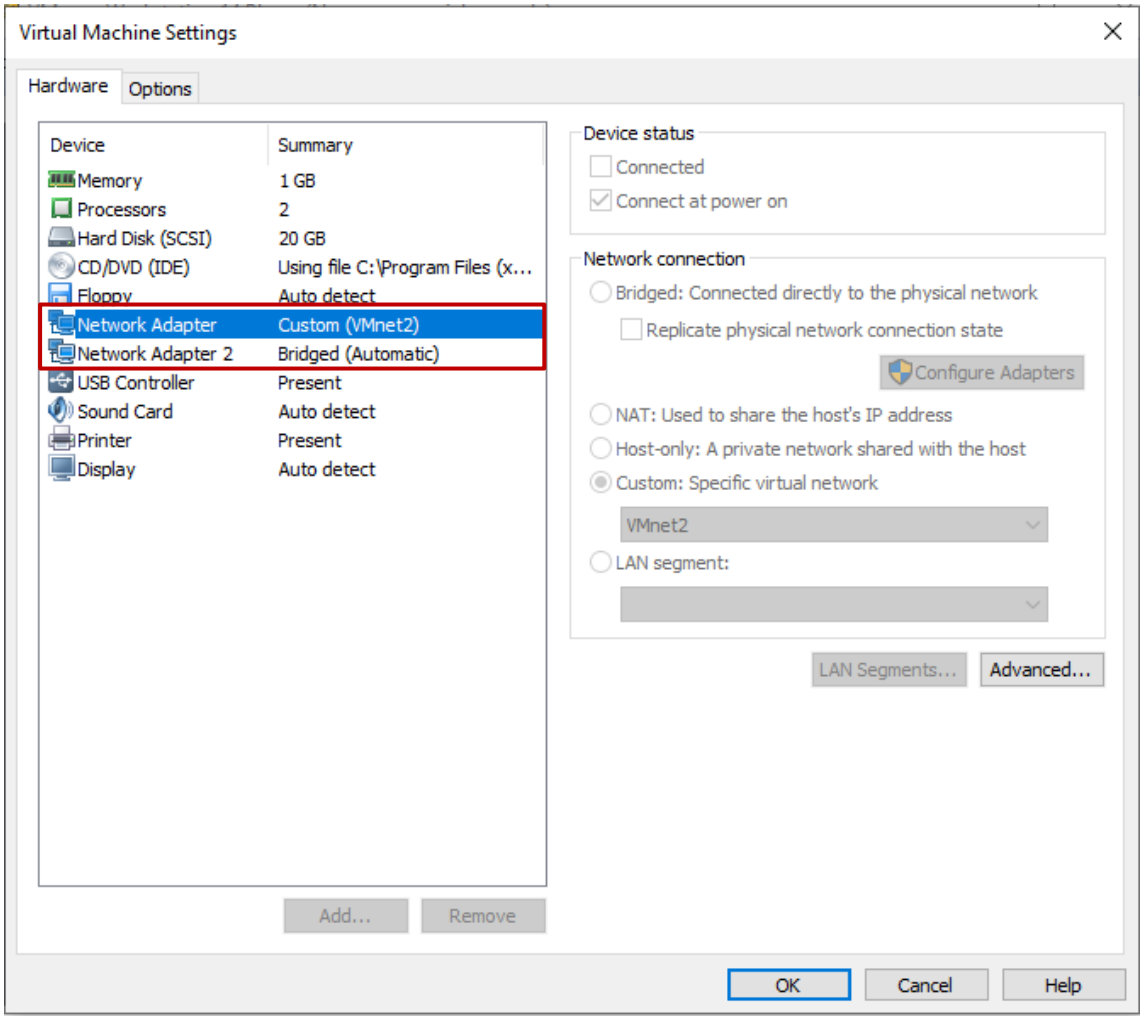


En el caso de las máquinas clientes, la configuración de los adaptadores de red fue coherente con su posición en la propia red, es decir, a cada una de las máquinas le fue asignada la subred adecuada dependiendo de qué enrutador le sirviera. En el caso de RCO-X se dejó el adaptador principal en la configuración por defecto *bridged (automatic)*. Esto consigue que la máquina virtual comparta la misma interfaz de red que la máquina anfitriona. [5] Por otro lado, el segundo adaptador se conectó a VMnet3 para que esté situado en la misma subred que ddwrt-x.

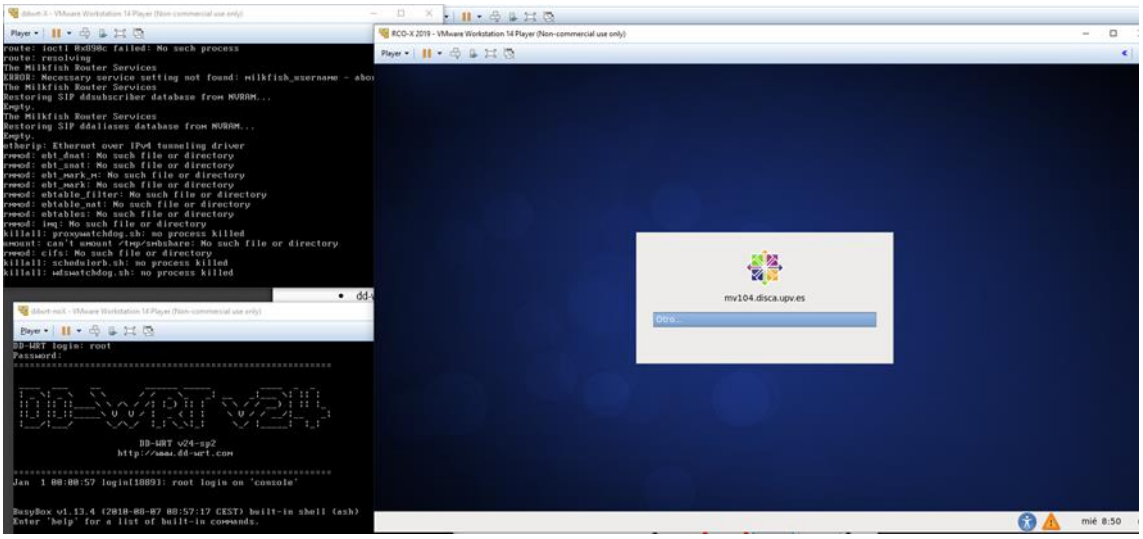
La configuración es como sigue:



Para RCO, la configuración es muy similar, simplemente se procedió a conectarlo a la subred requerida para que fuera accesible al enrutador ddwrt-noX:



Tras estas breves pero necesarias configuraciones y revisiones, se procedió al encendido de las máquinas y a la comprobación de que todas cuatro funcionaban según lo esperado.



Para conseguir una correcta configuración de las subredes y lograr conectividad entre las distintas partes de nuestra red fue necesario dotar a los distintos miembros de ésta IP congruentes con la figura 1, mostrada en el capítulo 1. En el caso de los encaminadores, esta configuración se realizó con la orden `ifconfig`. [7] Para `ddwrt-noX` se usó:

```
ifconfig eth0 10.19.17.1 netmask 255.255.255.0
```

Mientras que para `ddwrt-X` utilizamos:

```
ifconfig eth0 10.19.17.111 netmask 255.255.255.0
```

Tras estas órdenes, las interfaces de red de ambos encaminadores quedaron como sigue. Para `ddwrt-noX`:

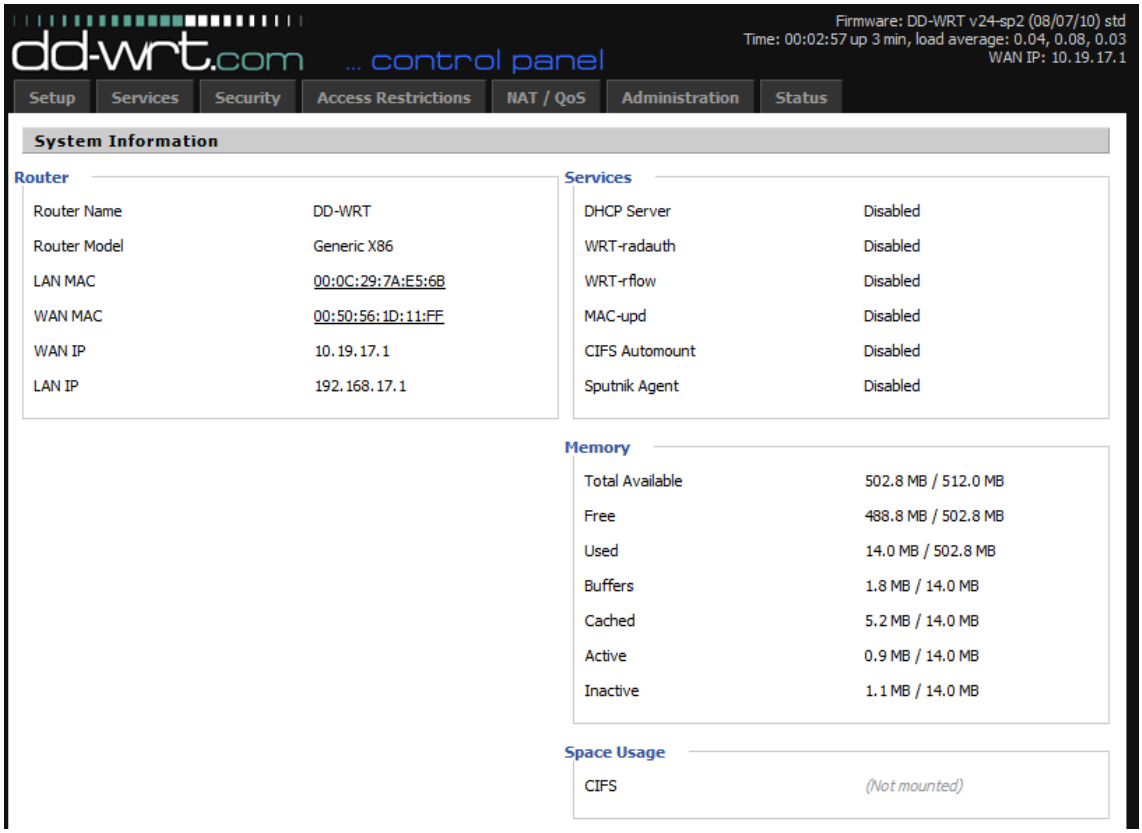
```
root@DD-WRT:~# ifconfig eth0 10.19.17.1 netmask 255.255.255.0
root@DD-WRT:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:56:1D:11:FF
          inet addr:10.19.17.1  Bcast:10.19.17.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:598 errors:0 dropped:0 overruns:0 frame:0
          TX packets:63 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:54552 (53.2 KiB)  TX bytes:20412 (19.9 KiB)
          Interrupt:5 Base address:0x2000

root@DD-WRT:~#
```

Para `ddwrt-X`:

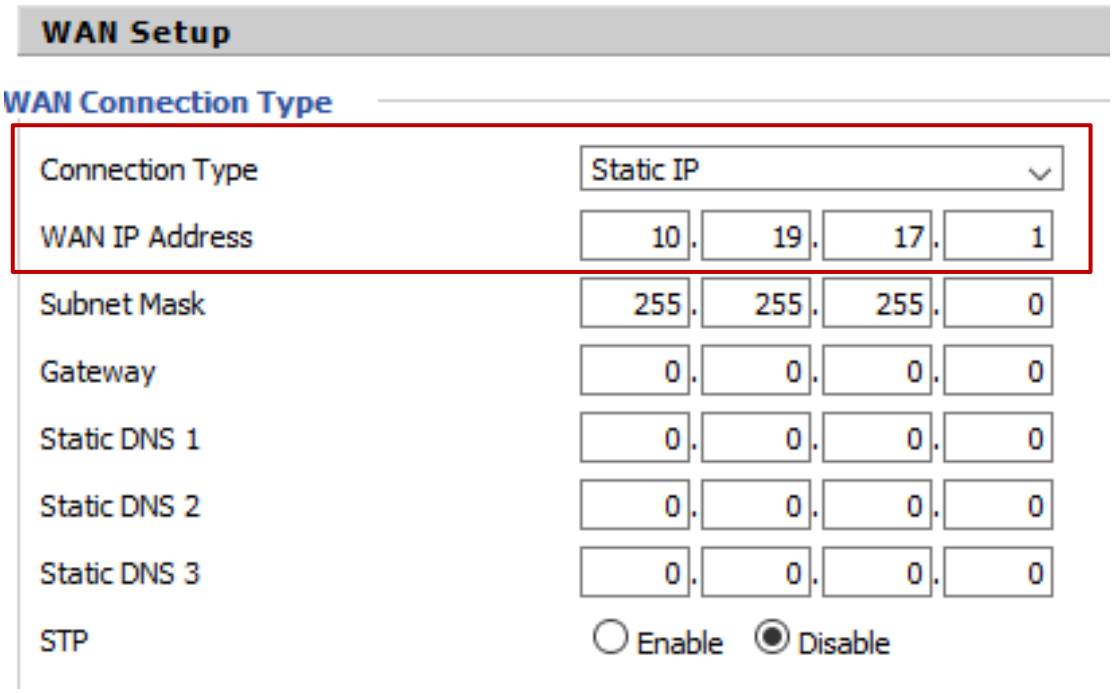
```
root@DD-WRT:~# ifconfig eth0 10.19.17.111 netmask 255.255.255.0
root@DD-WRT:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:56:1D:12:FF
          inet addr:10.19.17.111  Bcast:10.19.17.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:573 errors:0 dropped:0 overruns:0 frame:0
          TX packets:57 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:53688 (52.4 KiB)  TX bytes:18468 (18.0 KiB)
          Interrupt:5 Base address:0x2000

root@DD-WRT:~#
```

Para que los cambios fuesen permanentes en los enrutadores fue necesario guardar los cambios en la memoria no volátil. Si bien esto posible mediante comandos, la opción más factible resulta por hacerlo desde la interfaz gráfica que nos ofrece DD-WRT. Para acceder a dicha interfaz, bastó con escribir la dirección IP que acabamos de configurar en el navegador del PC. El hecho de que estas máquinas virtuales fuesen ya accesibles mediante el navegador de nuestra máquina física muestra que están en la misma red y son visibles entre ellas.

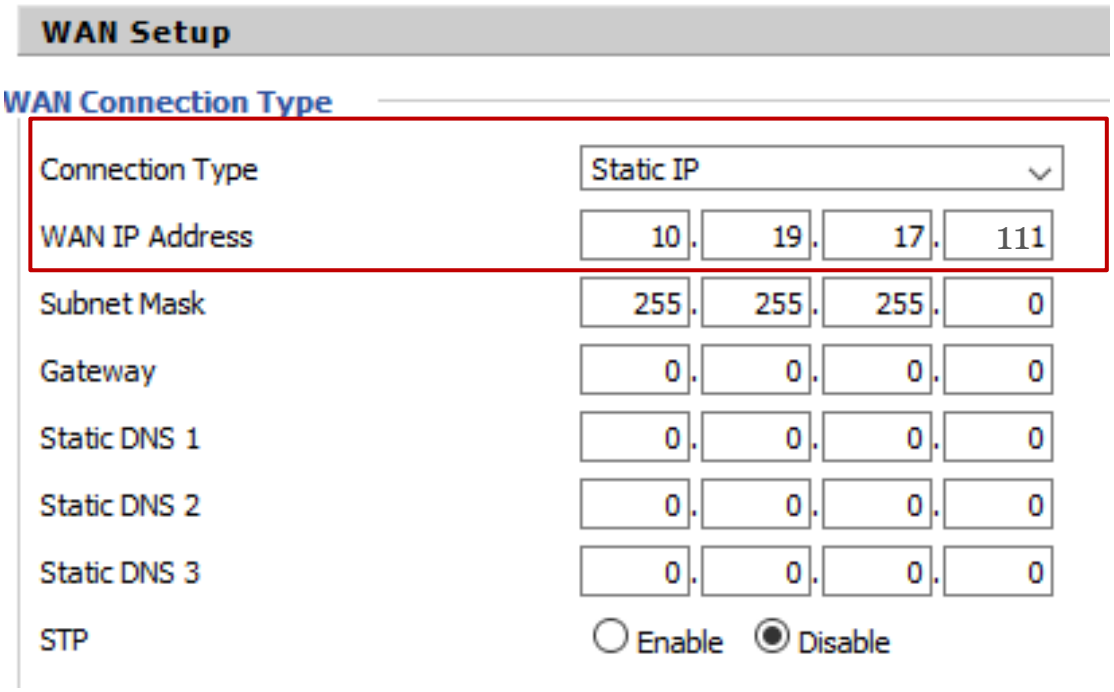
El panel de control del enrutador `ddwrt-noX` es el siguiente. Obsérvese la correcta IP externa que se muestra en la esquina superior derecha.



Dentro de esta interfaz, nos dirigimos a la pestaña “Setup”, y en el desplegable llamado “Connection Type”, en “WAN Setup”, seleccionamos “Static IP”. Esto sirve para configurar de manera correcta y permanente la IP externa, correspondiente a `VMnet1`, del enrutador.

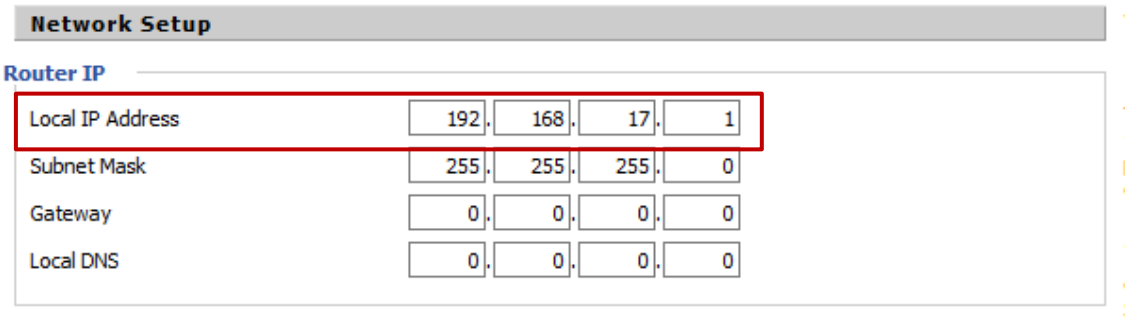


En ddwrt-X, esta configuración se realizó de la misma manera, ya que no hay ningún cambio de procedimiento en estos dos encaminadores. Hay que destacar el cambio de IP externa, que responde al diagrama presentado en la figura 1 del capítulo 1.

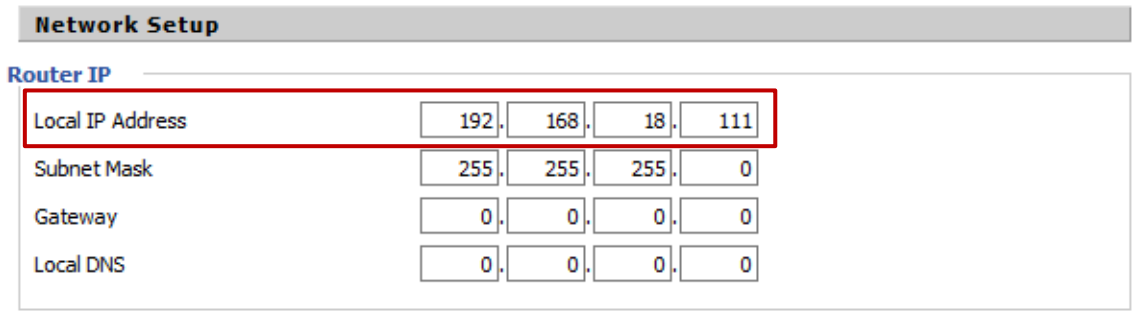


En esta pantalla también encontramos la interfaz para cambiar la IP interna del enrutador, que pertenece al segundo adaptador de red que hemos especificado con anterioridad.

Dos secciones por debajo de "*WAN Setup*" encontramos "*Network setup*". Aquí completamos "*Local IP Adress*" con la IP interna, correspondiente a `VMnet2`, requerida para `ddwrt-noX`.



Para el enrutador `ddwrt-X` hicimos el mismo procedimiento, pero con la IP correspondiente a dicho encaminador:



Para mantener ambas IP configuradas tras apagar los encaminadores es necesario pulsar "*Apply*" y "*Save*", pues son estos dos botones quienes escriben y guardan en la memoria no volátil los cambios que hemos realizado hasta que el encaminador sea reseteado.

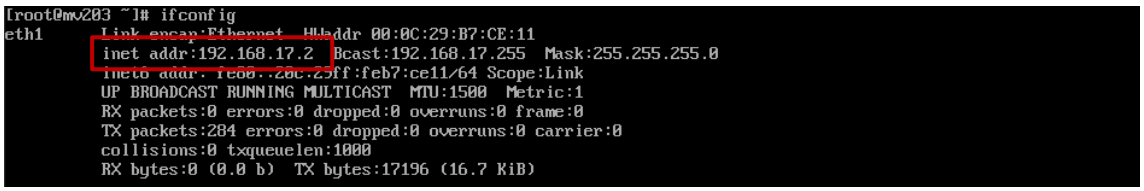
Una vez tuvimos ambos encaminadores bien configurados, pasamos a ajustar los parámetros de las máquinas virtuales. El cliente `RCO-X` toma su dirección IP por DHCP. En este caso y por este hecho, no fue necesaria la configuración de su interfaz principal, ya que fue el propio encaminador `ddwrt-X` quien le sirvió su propia dirección única.

La máquina cliente que configuramos fue RCO. Para cambiar su dirección IP accedimos al archivo `ifcfg-eth1` y lo editamos. Este archivo se encuentra en

`/etc/sysconfig/network-scripts`. Una vez abierto el archivo, cambiamos la entrada `IPADDR` por la requerida, en nuestro caso fue `192.168.17.2`:

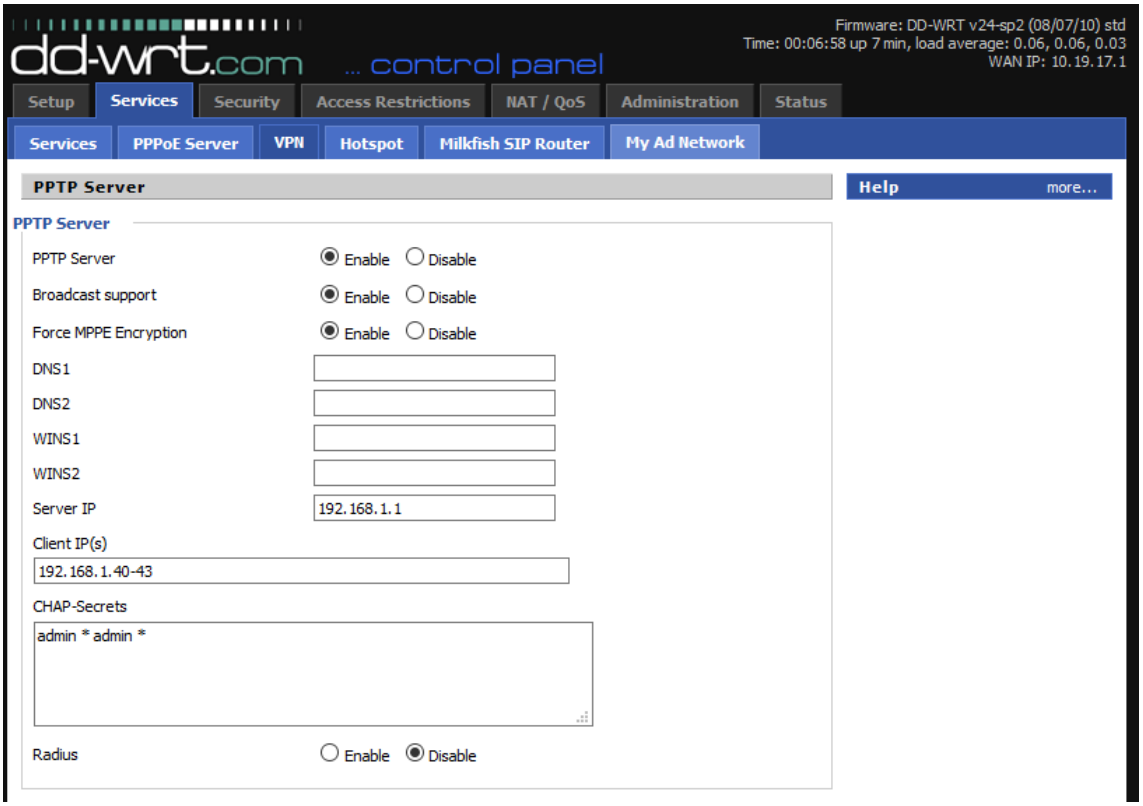
Para que el cambio de IP fuera definitivo tumbamos y volvimos a levantar la interfaz asociada con las órdenes `ifdown eth1` y `ifup eth1`.

La comprobación de que dicha IP fue configurada correctamente la obtuvimos con `ifconfig`. [7]



Con todo esto, la red que hemos mostrado en la figura 1 estuvo ya completamente funcional. El siguiente paso que dimos fue la preparación de los enrutadores para que admitiesen túneles PPTP. Para esta parte, `ddwrt-noX` fue configurado como servidor y `ddwrt-X` como cliente.

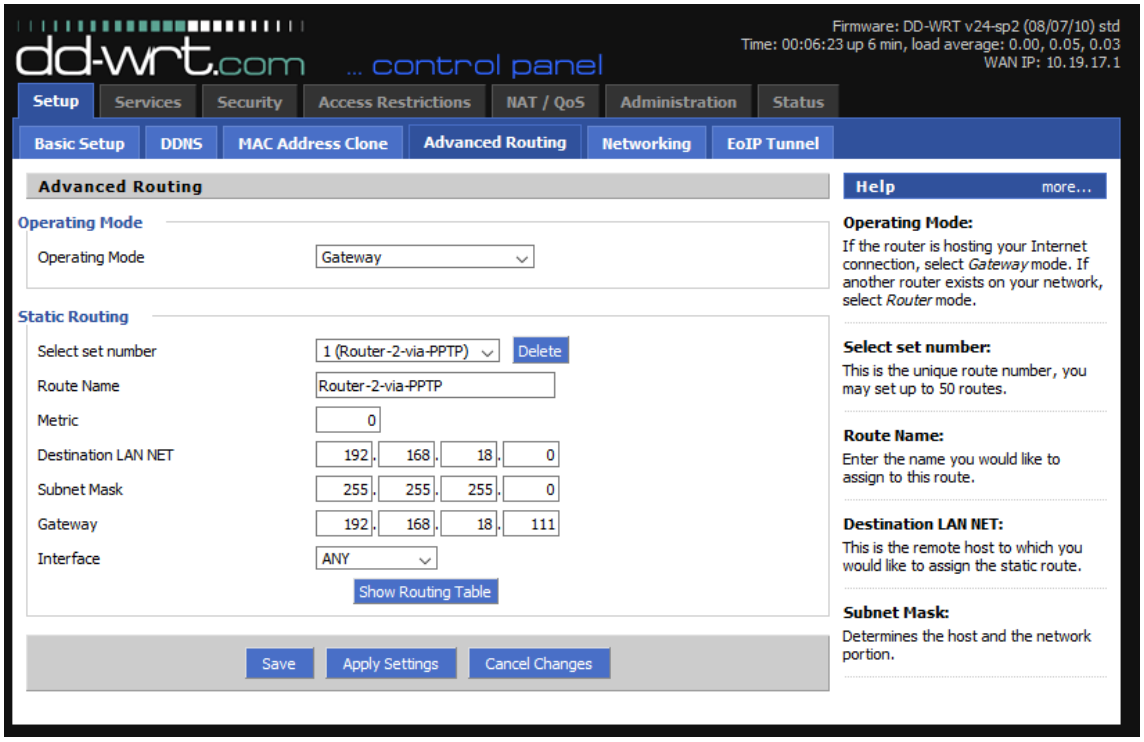
La configuración del enrutador servidor se hizo como sigue: Primeramente, accedimos a la interfaz gráfica del enrutador y nos dirigimos a la pestaña "*Services*" y luego, a la sub-pestaña "*VPN*", donde activamos la opción "*PPTP Server*".



En el apartado "*Server IP*" fue necesario poner la dirección del servidor, la local del enrutador, en nuestro caso `192.168.1.1` y en "*Client IP(s)*" las direcciones de los clientes que quisiéramos aceptar. En este caso, el rango elegido fue de `192.168.1.40` a `192.168.1.43`, como se indica.

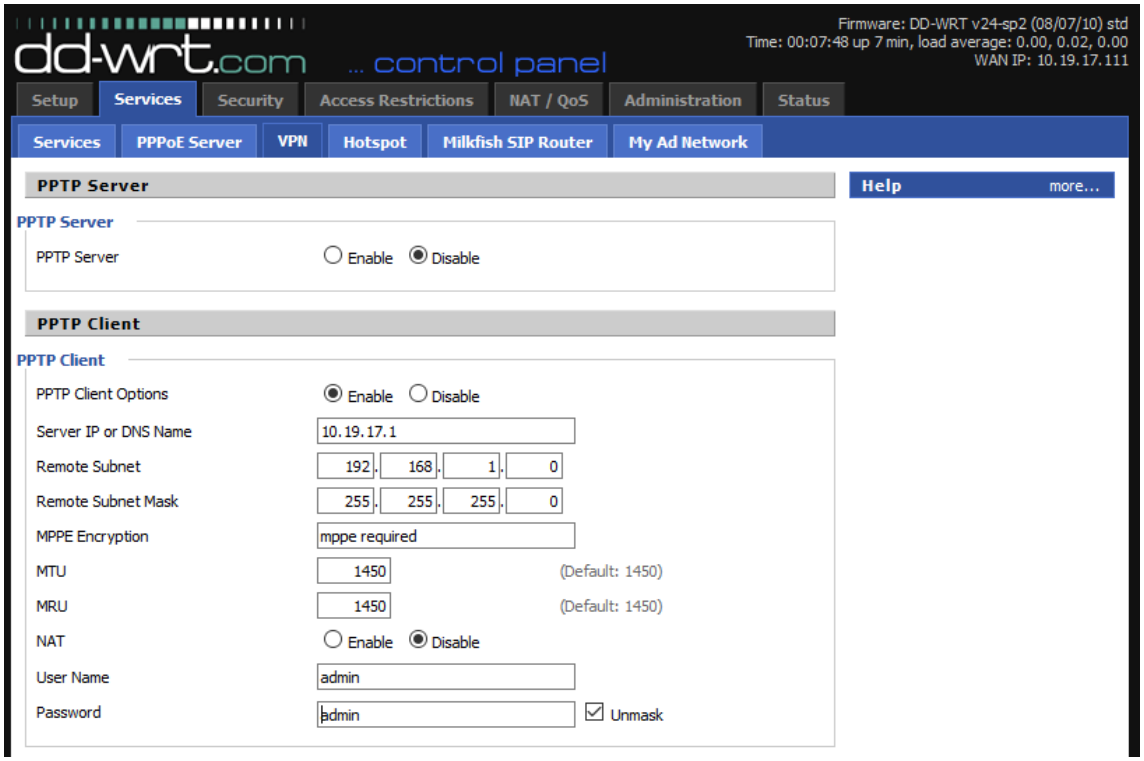
La seguridad del túnel se especificó en el campo "*CHAP-Secrets*", que almacena las credenciales de los usuarios que usen el túnel.

El siguiente paso fue indicar al enrutador qué camino deben tomar los paquetes que deban ir al otro enrutador. Esta configuración se hizo desde la pestaña "*Setup*", en la sub-pestaña "*Advanced routing*".



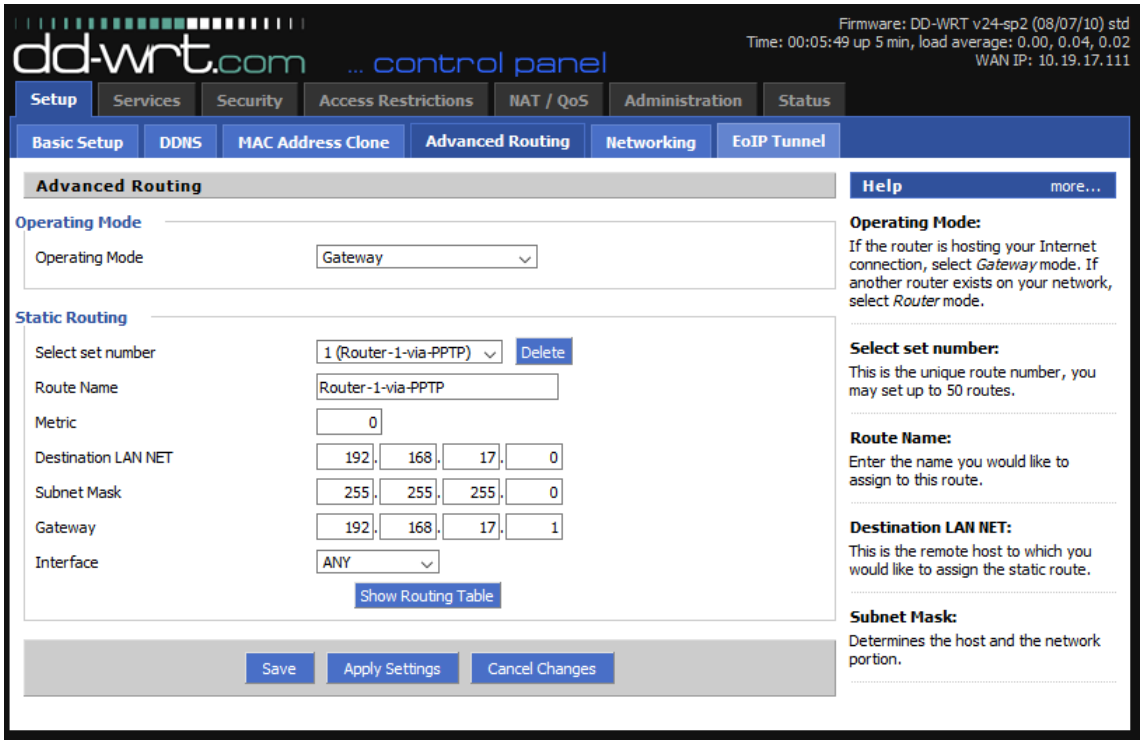
Después de haber seleccionado del desplegable una de las 20 conexiones posibles, fue necesario introducir en los campos siguientes la subred del enrutador destino, la máscara y su puerta de enlace. En nuestro caso, la subred destino es `192.168.18.0`, la máscara `255.255.255.0` y la puerta de enlace `192.168.18.111`, siendo "*ANY*" la interfaz.

La configuración del enrutador cliente, `ddwrt-X`, se hizo de manera similar. Accedimos a la pestaña "*Services*" y luego, a la sub-pestaña "*VPN*", pero esta vez no activamos la función de servidor, sino la de cliente.



En el campo "*Server IP or DNS Name*" introdujimos la dirección del servidor, en nuestro caso 10.19.17.1. Como subred se especificó la máscara `255.255.255.0`. También especificamos el tipo de encriptación y las credenciales de conexión que el enrutador servidor tiene guardadas.

La tabla de encaminamiento se ajustó de manera análoga a la de `ddwrt-noX`. Obsérvese el cambio en las IP internas y la puerta de enlace para que se ajusten a la red de `ddwrt-noX`.



Una vez salvadas y aplicadas ambas dos configuraciones, fue necesario reiniciar los enrutadores para que los cambios tomaran efecto en su totalidad. Este reinicio se realizó desde la pestaña "*Administration*" de la interfaz gráfica.

3. Funcionamiento del túnel PPTP site-to-site

Para comprobar que la configuración funcionaba correctamente, hicimos un *ping* desde la máquina `RCO` a `RCO-X` e intentamos capturar dicho tráfico mediante Wireshark[6] en el PC del laboratorio. [8]

```
ping 192.168.18.112
```

Tras leer los resultados, observamos claramente como los paquetes son ilegibles desde la máquina que captura.

81	31.182679	10.19.17.111	10.19.17.1	PPP Comp	66 Compressed data
82	31.183548	10.19.17.1	10.19.17.111	PPP Comp	76 Compressed data
83	31.425418	10.19.17.1	10.19.17.111	PPP LCP	56 Echo Request
84	31.426009	10.19.17.111	10.19.17.1	PPP LCP	60 Echo Reply
85	31.480091	10.19.17.1	10.19.17.111	GRE	46 Encapsulated PPP
86	32.022705	10.19.17.111	10.19.17.1	PPP LCP	56 Echo Request
87	32.023453	10.19.17.1	10.19.17.111	PPP LCP	60 Echo Reply
88	32.186248	10.19.17.111	10.19.17.1	PPP Comp	70 Compressed data
89	32.187821	10.19.17.1	10.19.17.111	PPP Comp	76 Compressed data
90	32.686063	10.19.17.111	10.19.17.1	GRE	46 Encapsulated PPP
91	33.189792	10.19.17.111	10.19.17.1	PPP Comp	67 Compressed data
92	33.191470	10.19.17.1	10.19.17.111	PPP Comp	77 Compressed data
93	33.690982	10.19.17.111	10.19.17.1	GRE	46 Encapsulated PPP
94	34.023187	10.19.17.111	10.19.17.1	PPP LCP	56 Echo Request
95	34.023824	10.19.17.1	10.19.17.111	PPP LCP	60 Echo Reply
96	34.193119	10.19.17.111	10.19.17.1	PPP Comp	70 Compressed data
97	34.194904	10.19.17.1	10.19.17.111	PPP Comp	76 Compressed data
98	34.695545	10.19.17.111	10.19.17.1	GRE	46 Encapsulated PPP
99	36.027142	10.19.17.111	10.19.17.1	PPP LCP	56 Echo Request
100	36.027842	10.19.17.1	10.19.17.111	PPP LCP	60 Echo Reply
101	36.431603	10.19.17.1	10.19.17.111	PPP LCP	56 Echo Request
102	36.432570	10.19.17.111	10.19.17.1	PPP LCP	60 Echo Reply

No obstante, si a su vez realizamos una captura con Wireshark en la máquina `RCO-X` comprobamos que los paquetes ya pueden ser leídos de manera normal.

1	0.000000000	192.168.18.112	192.168.17.2	ICMP	98 Echo (ping) request id=0xf40f, seq=1/256, ttl=64
2	0.001209155	192.168.17.2	192.168.18.112	ICMP	98 Echo (ping) reply id=0xf40f, seq=1/256, ttl=62
3	1.002292977	192.168.18.112	192.168.17.2	ICMP	98 Echo (ping) request id=0xf40f, seq=2/512, ttl=64
4	1.005664690	192.168.17.2	192.168.18.112	ICMP	98 Echo (ping) reply id=0xf40f, seq=2/512, ttl=62
5	2.006528635	192.168.18.112	192.168.17.2	ICMP	98 Echo (ping) request id=0xf40f, seq=3/768, ttl=64
6	2.009548594	192.168.17.2	192.168.18.112	ICMP	98 Echo (ping) reply id=0xf40f, seq=3/768, ttl=62
7	3.009748250	192.168.18.112	192.168.17.2	ICMP	98 Echo (ping) request id=0xf40f, seq=4/1024, ttl=64
8	3.012821325	192.168.17.2	192.168.18.112	ICMP	98 Echo (ping) reply id=0xf40f, seq=4/1024, ttl=62
9	4.013386000	192.168.18.112	192.168.17.2	ICMP	98 Echo (ping) request id=0xf40f, seq=5/1280, ttl=64

6 www.wireshark.org

Este funcionamiento es el esperado del protocolo conforme lo hemos configurado. PPTP está diseñado para encapsular tráfico punto a punto, aportando cierta seguridad. Este tráfico se encapsula en `ddwrt-noX` con las credenciales especificadas y se decodifica en `ddwrt-X`, donde se sirve como tráfico interno común. En este caso, el tráfico va encapsulado en paquetes del protocolo PPP COMP, así como PPP LCP y GRE. [9] [10] [11]

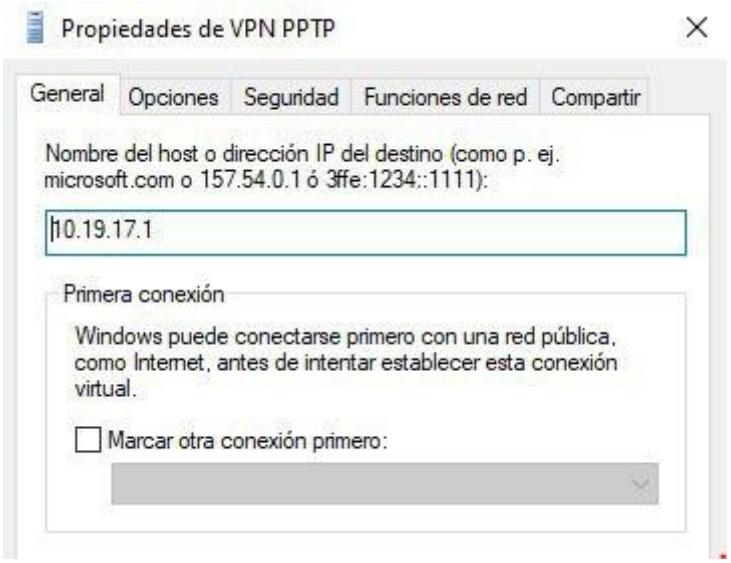
En el caso de PPP COMP, dentro tendríamos los paquetes propios del *ping*, es decir, los ICMP que comúnmente veríamos generados. En la información que arroja Wireshark, en el apartado de información, solamente vemos "*Compressed Data"* pero al ver que los paquetes tienen direcciones origen y destino alternadas, asumimos que son *ping* con sus respuestas.

La función de PPP LCP es la de establecer, configurar y comprobar la calidad y características del enlace entre ambos dispositivos. En nuestro caso lo observamos con "*Echo Request"* y "*Echo Reply"*, generados cada 2 segundos a lo largo de la captura realizada.

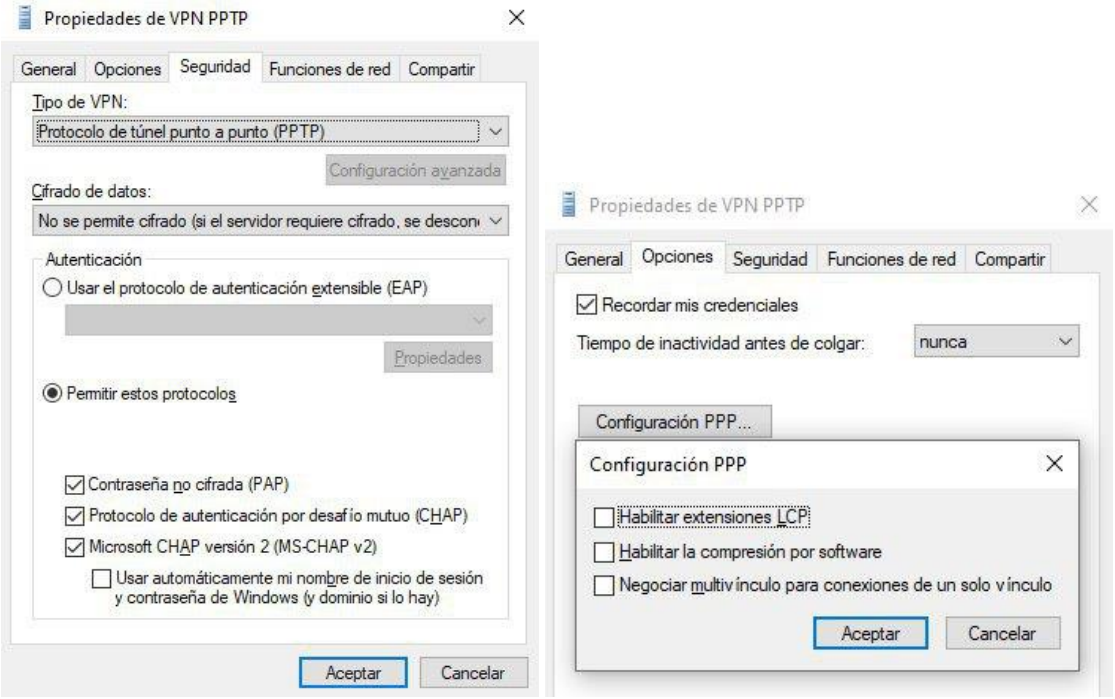
Para finalizar, también se observa tráfico bajo el protocolo GRE, un protocolo de establecimiento de túneles que permite transportar otros protocolos de red, optimizar comunicaciones entre redes y aumentar su eficiencia. Su uso se puede ver en muchos tipos de VPN y túneles PPTP como el que hemos realizado.

4. Funcionamiento del túnel PPTP remote access

El equipo de laboratorio tiene una interfaz de VMWare activa con una dirección IP compatible con la red que tenemos montada (en nuestro caso, `10.19.17.18`) para que pueda ver el tráfico que se mueve dentro de la infraestructura. Esta conexión está activa en todo momento, pero para ver con más detalle el tráfico de PPTP tenemos que conectar nuestro equipo al túnel. Esto se consigue configurando una VPN en nuestro ordenador a la dirección externa del enrutador que hace de servidor en el esquema que tenemos montado, `ddwrt-noX`.



Una vez hecho esto, tendremos que desactivar también la encriptación del túnel mediante la interfaz web del enrutador que actúa de servidor y la compresión en las opciones del túnel, cambiando la configuración de éste en Windows.



Lamentablemente, teniendo estas opciones como se requiere y así se anuncian en el boletín, el tráfico que observamos es el mismo. Además, que como Windows detecta otra nueva conexión manda tráfico que puede llegar a entorpecer aún más la interpretación de los datos.

La única manera que hemos encontrado para ver el tráfico sin el encapsulado PPTP es mandar los pings desde la propia maquina Windows, pero el caso es el mismo que en el

expuesto en el apartado anterior; al ser desde nuestra máquina el tráfico aún no ha sido encapsulado cuando Wireshark lo analiza.

En este último caso, al no pertenecer nuestro equipo a ninguna de las subredes formadas de `ddwrt-noX` y `ddwrt-X`, la propia encriptación y compresión se realizan en nuestro equipo ya que con los mensajes PPP LCP se puede configurar la conexión de manera correcta.

5. Conclusiones

PPTP es un protocolo diseñado a finales de los años 90 que, si bien ya no está recomendado por Microsoft, uno de sus principales desarrolladores, sí que sigue siendo accesible y desplegado. Se concibió como una extensión de PPP y como hemos visto, se apoya en el encapsulamiento de este protocolo para mantener un túnel.

Hemos de tener en cuenta que PPTP estándar no ofrece ningún tipo de seguridad por defecto, lo cual puede ser bastante peligroso si se trata de implementar una VPN empresarial, donde puedan pasar datos sensibles. Es por ello que se añaden otros protocolos, vistos en esta misma memoria, para asegurar confidencialidad y autenticidad. Uno de ellos es MPPE y otro MS-CHAP v2. [12] [13] [14]

MPPE es un protocolo desarrollado por Microsoft que cifra los paquetes PPP usando el algoritmo RC4, con claves de sesión tanto de 40 como de 128 bits, que pueden ser cambiadas según las opciones de negociación, llegando a poder variar en cada paquete. El encargado de esta negociación es el protocolo CCP, que establece las bases para el intercambio de la clave de sesión.

MS-CHAP v2 también es un protocolo desarrollado por Microsoft sobre CHAP. [15] Esta segunda versión de MS-CHAP incorpora tanto este primero como CHAP y cambia el método de autenticación respecto a su primera versión en tanto que incorpora *piggybacking* en el paquete de respuesta para entregar el desafío.

Ambos dos protocolos, si bien en su momento servían para mantener cierta seguridad en las comunicaciones, han quedado obsoletos hoy en día. MS-CHAP v2 presenta vulnerabilidades extremadamente severas que no solo reducen la seguridad de este protocolo a la dificultad de la contraseña que se use para encriptar la conexión, sino que dicha contraseña puede ser capturada y atravesada mediante ataques de diccionario. [16] A estos efectos, una contraseña larga y segura puede ser una buena forma de hacer casi imposible un ataque de diccionario, si no fuera porque MS-CHAP v2 puede proveer al atacante con la información necesaria para derivar la contraseña a partir de muy poca información.

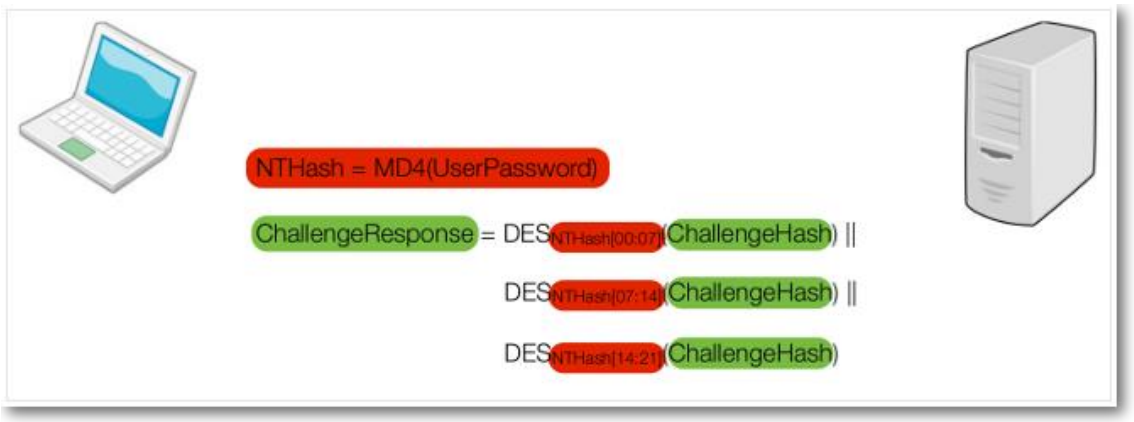


Ilustración 1: DES simple

Debido a esto, herramientas como ASLEAP, que se aprovechan de esta falla de seguridad, han sido desarrolladas, lo cual permite a un atacante hacerse con la visión total de la conexión a pesar de tener el protocolo MS-CHAP v2. [17]

MPPE no se queda atrás en cuanto a vulnerabilidades se refiere. Posee la misma brecha de seguridad que MS-CHAP v2 en tanto que es posible derivar la clave de sesión de este protocolo si se rompe la seguridad del segundo. [16] Una solución a esto reside en sustituir MS-CHAP por EAP-TLS, como recomienda Microsoft. [18] [19]. No obstante, sigue siendo posible modificar el contenido de estos paquetes mediante *bit flipping*, ya que no hay manera de asegurar que el texto cifrado mantenga su integridad si no se dispone de *checksums* u otro tipo de protocolos. [16] [20]

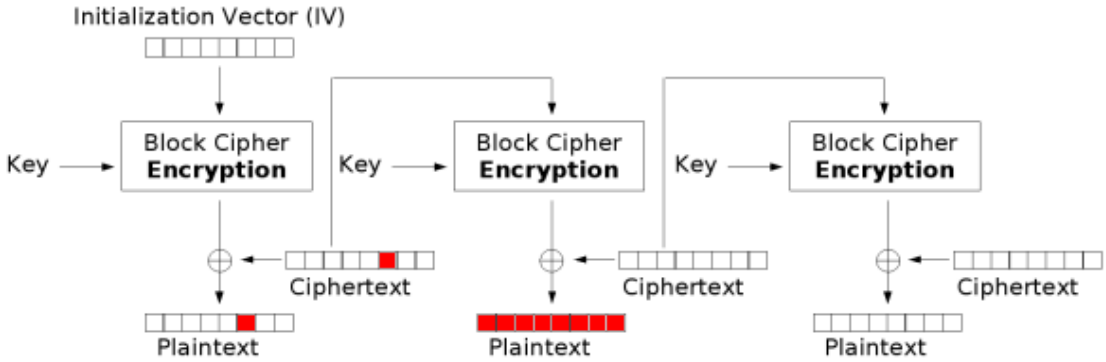


Ilustración 2: Bit-flipping attack

En conclusión, PPTP ha sido demostrado inseguro por multitud de fuentes, incluidos sus propios creadores. [21] Uno de los mayores fallos de PPTP consiste en apoyarse

completamente en MPPE, dejando al protocolo roto de nacimiento. No obstante, la existencia de otras soluciones más seguras como IPsec, L2TP o SSTP suponen un soplo de aire fresco a la creación de túneles, mejorando la seguridad existente en PPTP y manteniendo la infraestructura necesaria para la creación de VPN sin influir de manera significativa en la rapidez de las conexiones.

Referencias

[1] Wikipedia, «Remote access service,» Wikipedia, 4 Octubre 2019. [En línea]. Available: https://en.wikipedia.org/wiki/Remote_access_service.

[2] K. Hamze, G. P. W. V. J. T. W. L. y G. Z. , «RFC 2637,» The Internet Society, Julio 1999. [En línea]. Available: https://tools.ietf.org/html/rfc2637.

[3] G. Held, «Access Concentrator-Network Server Model,» de *Virtual Private Networking: A Construction, Operation and Utilization Guide*, Georgia, Wiley, 2005, p. 92.

[4] Microsoft, «Cambiar la configuración de TCP/IP,» 21 Mayo 2019. [En línea]. Available: https://support.microsoft.com/es-es/help/15089/windows-change-tcp-ip-settings.

[5] VMware, «Understanding Common Networking Configurations,» VMware, [En línea]. Available: https://pubs.vmware.com/workstation-11/topic/com.vmware.ws.using.doc/GUID-D9B0A52D-38A2-45D7-A9EB-987ACE77F93C.html.

[6] VMware, «Configuring Host-Only Networking,» VMware, [En línea]. Available: https://pubs.vmware.com/workstation-11/topic/com.vmware.ws.using.doc/GUID-93BDF7F1-D2E4-42CE-80EA-4E305337D2FC.html.

[7] Wikipedia, «Ifconfig,» Wikipedia, [En línea]. Available: https://es.wikipedia.org/wiki/Ifconfig.

[8] Wikipedia, «Ping (networking utility),» Wikipedia, 26 Septiembre 2019. [En línea]. Available: https://en.wikipedia.org/wiki/Ping_(networking_utility).

[9] D. R. Novell, «The PPP Compression Control Protocol (CCP),» Network Working Group, Junio 1996. [En línea]. Available: https://tools.ietf.org/html/rfc1962.

[10] W. Simpson, «PPP LCP Extensions,» Network Working Group, Junio 1994. [En línea]. Available: https://tools.ietf.org/html/rfc1570.

[11] T. Li, S. Hanks, D. Farinacci y P. Traina, «Generic Routing Encapsulation (GRE),» Octubre 1994. [En línea]. Available: https://tools.ietf.org/html/rfc1701.

[12] G. Pall y G. Z. , «RFC 3078,» Microsoft, Marzo 2001. [En línea]. Available: https://tools.ietf.org/html/rfc3078.

[13] Network Sorcery, «MPPE, Microsoft Point-To-Point Encryption Protocol,» [En línea]. Available: http://www.networksorcery.com/enp/protocol/mppe.htm.

[14] G. Zorn, «RFC 2759,» Microsoft, Enero 2000. [En línea]. Available:

https://tools.ietf.org/html/rfc2759.

[15] Wikipedia, «CHAP,» Wikipedia, 16 Julio 2019. [En línea]. Available: https://es.wikipedia.org/wiki/CHAP.

[16] B. Schneier, M. y D. W. , «Cryptanalysis of Microsoft's PPTP Authentication Extensions (MS-CHAPv2),» 19 Octubre 1999. [En línea]. Available: https://www.schneier.com/academic/paperfiles/paper-pptpv2.pdf.

[17] J. Wrigth, «Will hack for sushi,» 28 Mayo 2008. [En línea]. Available: https://www.willhackforsushi.com/?page_id=41.

[18] D. Simons, B. A. y R. H. , «RFC 5216,» Microsoft, Marzo 2008. [En línea]. Available: https://tools.ietf.org/html/rfc5216.

[19] Microsoft, «Choosing EAP-TLS or MS-CHAP v2 for User-Level Authentication,» Microsoft, 10 Agosto 2009. [En línea]. Available: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc739638(v=ws.10)?redirectedfrom=MSDN.

[20] Wikipedia, «Bit-flipping attack,» Wikipedia, 4 Noviembre 2013. [En línea]. Available: https://en.wikipedia.org/wiki/Bit-flipping_attack.

[21] H Security, «Microsoft says don't use PPTP and MS-CHAP,» 22 Agosto 2012. [En línea]. Available: http://www.h-online.com/security/news/item/Microsoft-says-don-t-use-PPTP-and-MS-CHAP-1672257.html.