

Práctica 1.1. Protocolo IPv4. Servicio DHCP

Objetivos

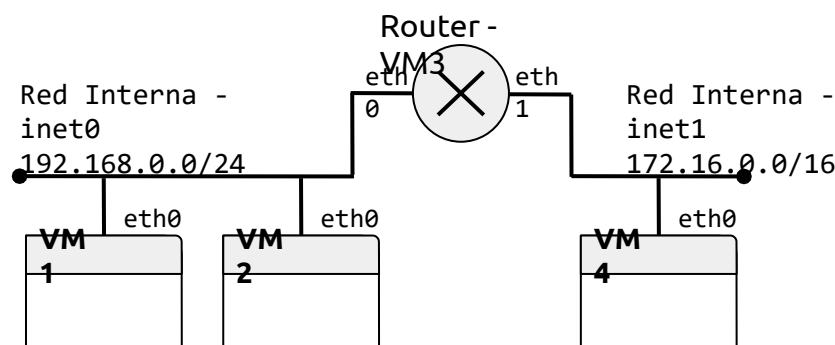
En esta práctica se presentan las herramientas que se utilizarán en la asignatura y se repasan brevemente los aspectos básicos del protocolo IPv4. Además, se analizan las características del protocolo DHCP.

Contenidos

- Preparación del entorno para la práctica
- Configuración estática
- Encaminamiento estático
- Configuración dinámica

Preparación del entorno para la práctica

Configuraremos la topología de red que se muestra en la siguiente figura:



Todos los elementos -el router y las máquinas virtuales VM- son *clones enlazados* de la máquina base ASOR-FE. La topología se creará con la utilidad `vtopo1`, que funciona en Linux y Mac (en Windows, la topología ha de crearse manualmente con VirtualBox):

1. Borrar las máquinas virtuales existentes ejecutando el siguiente comando en la consola:

```
rm -rf $HOME/VirtualBox\ VMs/
```

2. Usando el explorador de archivos, cambiar al directorio `/mnt/DiscoVMs/ASOR` y hacer doble-click sobre el fichero `ASOR-FE.ova`. Esto importará la máquina virtual base ASOR-FE en VirtualBox. Alternativamente, se puede usar la opción importar desde VirtualBox y seleccionar el OVA en el directorio anterior.

Nota: Si estás usando tu ordenador es necesario descargar el fichero [ASOR-FE.ova](#)

3. Crear un archivo `pr1.topo1` con la topología de la red, que consta de 4 máquinas y dos redes. El contenido del fichero es:

```
netprefix                                inet
machine                                1                                0
machine 2 0 0
machine                                3                                0                                0                                1                                1
machine 4 0 1
```

La sintaxis es:

```
machine <número de VM> <interfaz0> <red0> <interfaz1> <red1> ...
```

4. Crear la topología de red que arrancará las 4 máquinas virtuales (VM1, VM2, Router y VM4).

```
$ vtopol pr1.topol
```

En VirtualBox se definirán las máquinas virtuales asorfemachine_1 (VM1), asorfemachine_2 (VM2), asorfemachine_3 (Router - VM3) y asorfemachine_4 (VM4).

Nota: El comando **vtopol** está instalado en el laboratorio. En otros equipos, descargar el fichero [vtopol](#), darle permisos de ejecución (con `chmod +x vtopol`) y copiarlo, por ejemplo, en `/usr/local/bin`.



Activar el portapapeles bidireccional en las máquinas (menú Dispositivos) para copiar la salida de los comandos. Las capturas de pantalla se realizarán usando también Virtualbox (menú Ver).

Las **credenciales de la máquina virtual** son: usuario `cursoresdes`, con contraseña `cursoresdes`.

Configuración estática

En primer lugar, configuraremos cada red de forma estática asignando a cada máquina una dirección IP adecuada.

Ejercicio 1 [VM1]. Determinar los interfaces de red que tiene la máquina y las direcciones IP y MAC que tienen asignadas. Utilizar los comandos `ip address` e `ip link`.

```
[cursoresdes@localhost ~]$ sudo ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 02:00:00:00:01:00 brd ff:ff:ff:ff:ff:ff
```

```
[cursoresdes@localhost ~]$ sudo ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ether 02:00:00:00:01:00 brd ff:ff:ff:ff:ff:ff
```

Ejercicio 2 [VM1, VM2, Router]. Activar los interfaces `eth0` en VM1, VM2 y Router, y asignar una dirección IP adecuada. Utilizar los comandos `ip address` e `ip link`.

VM1

```
[cursoresdes@localhost ~]$ sudo ip address add 192.168.0.1/24 dev eth0
[cursoresdes@localhost ~]$ sudo ip link set eth0 up
```

VM2

```
[cursoredes@localhost ~]$ sudo ip address add 192.168.0.2/24 dev eth0  
[cursoredes@localhost ~]$ sudo ip link set eth0 up
```

VM3

```
[cursoredes@localhost ~]$ sudo ip address add 192.168.0.3/24 dev eth0  
[cursoredes@localhost ~]$ sudo ip link set eth0 up
```

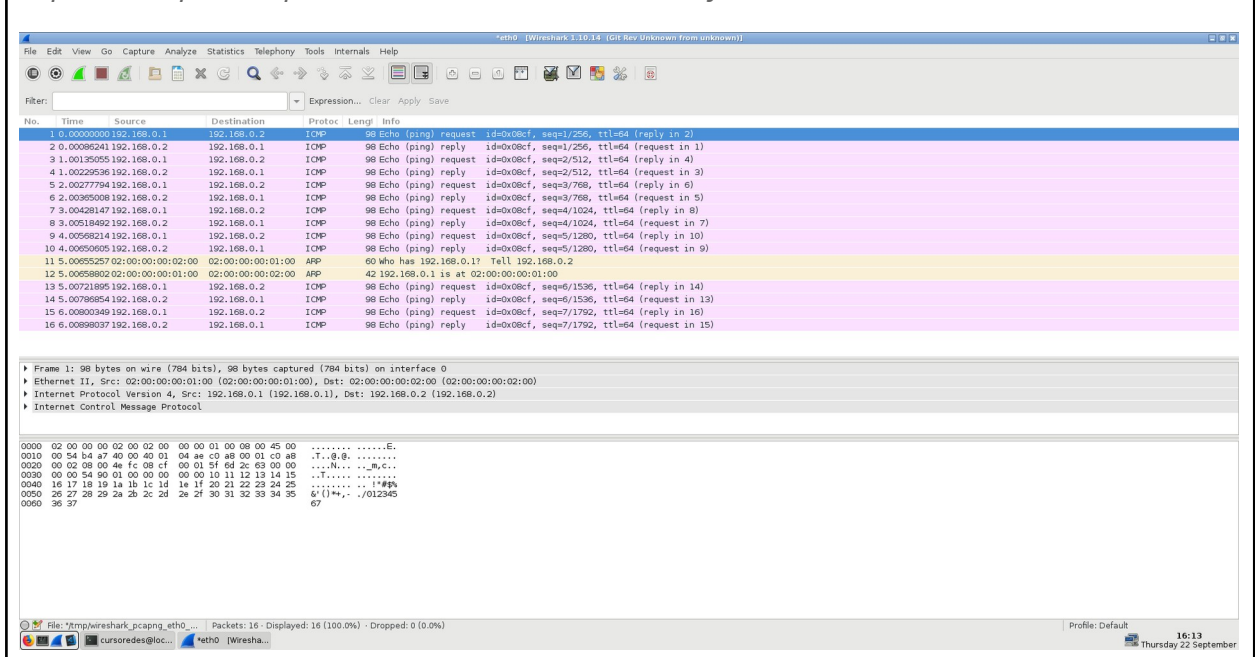
Ejercicio 3 [VM1, VM2]. Abrir la herramienta Wireshark en VM1 e iniciar una captura en el interfaz de red. Desde VM1, comprobar la conectividad con VM2 usando la orden ping. Observar el tráfico generado, especialmente los protocolos encapsulados en cada datagrama y las direcciones origen y destino. Para ver correctamente el tráfico ARP, puede ser necesario eliminar la tabla ARP en VM1 con la orden `ip neigh flush dev eth0`.

Completar la siguiente tabla para todos los mensajes intercambiados hasta la recepción del primer mensaje ICMP Echo Reply:

- Para cada protocolo, anotar las características importantes (p. ej. pregunta/respuesta ARP o tipo ICMP) en el campo "Tipo de mensaje".
- Comparar los datos observados durante la captura con el formato de los mensajes estudiados en clase.

MAC origen	MAC destino	Protocolo	IP origen	IP destino	Tipo de mensaje
02:00:00:00:01:00	ff:ff:ff:ff:ff:ff	ARP	192.168.0.1	192.168.0.2	ARP PREGUNTA
02:00:00:00:02:00	02:00:00:00:01:00	ARP	192.168.0.2	192.168.0.1	ARP RESPUESTA
02:00:00:00:1:00	02:00:00:00:02:00	ICMP	192.168.0.1	192.168.0.2	ICMP PREGUNTA
02:00:00:00:02:00	02:00:00:00:01:00	ICMP	192.168.0.2	192.168.0.1	ICMP RESPUESTA

Copiar una captura de pantalla de Wireshark con los mensajes ARP e ICMP.



Ejercicio 4 [VM1, VM2]. Ejecutar de nuevo la orden ping entre VM1 y VM2 y, a continuación, comprobar el estado de la tabla ARP en VM1 y VM2 usando el comando `ip neigh`. El significado del

estado de cada entrada de la tabla se puede consultar en la página de manual del comando.

```
[cursoredes@localhost ~]$ sudo ip neigh
192.168.0.3 dev eth0 lladdr 02:00:00:00:03:00 STALE
192.168.0.2 dev eth0 lladdr 02:00:00:00:02:00 REACHABLE
```

NOTA: Los estados pueden ser los siguientes
STATE := { permanent | noarp | stale | reachable | none |
incomplete | delay | probe | failed }

Con:

- REACHABLE: la entrada ARP es válida y hay conectividad.
- STALE: la entrada ARP es válida pero no hay conectividad. (Todavía no se ha hecho conexión)

Ejercicio 5 [Router, VM4]. Configurar Router y VM4 y comprobar su conectividad con el comando ping.

ROUTER

```
[cursoredes@localhost ~]$ sudo ip address add 172.16.0.1/16 dev eth1
[cursoredes@localhost ~]$ sudo ip link set eth1 up
```

VM4

```
cursoredes@localhost ~]$ sudo ip address add 172.16.0.2/16 dev eth0
[cursoredes@localhost ~]$ sudo ip link set eth0 up
```

Encaminamiento estático

Según la topología de esta práctica, Router puede encaminar el tráfico entre ambas redes. En esta sección, vamos a configurar el encaminamiento estático, basado en rutas que fijaremos manualmente en todas las máquinas virtuales.

Ejercicio 6 [Router]. Activar el reenvío de paquetes (*forwarding*) en Router para que efectivamente pueda funcionar como encaminador entre las redes. Ejecutar el siguiente comando:

```
$ sudo sysctl net.ipv4.ip_forward=1
```

Ejercicio 7 [VM1, VM2]. Establecer Router como encaminador por defecto para VM1 y VM2. Usar el comando `ip route`.

VM1:

```
[cursoredes@localhost ~]$ sudo ip route add default via 192.168.0.3
```

VM2:

```
[cursoredes@localhost ~]$ sudo ip route add default via 192.168.0.3
```

NOTA: En caso de poner erróneamente limpiar la red:
[cursoredes@localhost ~]\$ sudo service network restart
Restarting network (via systemctl): [OK]

Ejercicio 8 [VM4]. Aunque la configuración adecuada para la tabla de rutas en redes como las consideradas en esta práctica consiste en añadir una ruta por defecto, es posible incluir rutas para redes concretas. Añadir en VM4 una ruta a la red 192.168.0.0/24 vía Router. Usar el comando `ip route`.

VM4:

```
[cursoredes@localhost ~]$ sudo ip route add 192.168.0.0/24 via 172.16.0.1
```

Ejercicio 9 [VM1, VM4, Router]. Abrir la herramienta Wireshark en Router e iniciar dos capturas, una en cada interfaz de red. Eliminar la tabla ARP en VM1 y Router. Desde VM1, comprobar la conectividad con VM4 usando la orden ping. Completar la siguiente tabla para todos los paquetes intercambiados hasta la recepción del primer *Echo Reply*.

Red 192.168.0.0/24 - Router (eth0)

MAC origen	MAC destino	Protocolo	IP origen	IP destino	Tipo de mensaje
02:00:00:00:01:00	ff:ff:ff:ff:ff:ff	ARP	192.168.0.1	192.168.0.3	ARP PREGUNTA
02:00:00:00:03:00	02:00:00:00:01:00	ARP	192.168.0.3	192.168.0.1	ARP RESPUESTA
02:00:00:00:01:00	02:00:00:00:03:00	ICMP	192.168.0.1	172.16.0.2	ICMP PREGUNTA
02:00:00:00:03:00	02:00:00:00:01:00	ICMP	172.16.0.2	192.168.0.1	ICMP RESPUESTA

Red 172.16.0.0/16 - Router (eth1)

MAC origen	MAC destino	Protocolo	IP origen	IP destino	Tipo de mensaje
02:00:00:00:04:00	ff:ff:ff:ff:ff:ff	ARP	172.16.0.1	172.16.0.2	ARP PREGUNTA
02:00:00:00:03:01	02:00:00:00:04:00	ARP	172.16.0.2	172.16.0.1	ARP RESPUESTA
02:00:00:00:04:00	02:00:00:00:03:01	ICMP	192.168.0.1	172.16.0.2	ICMP PREGUNTA
02:00:00:00:03:01	02:00:00:00:04:00	ICMP	172.16.0.2	192.168.0.1	ICMP RESPUESTA

ETH0

*eth0 [Wireshark 3.10.14 (Git Rev Unknown from unknown)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protoc	Length	Info
1	0.00000000	02:00:00:00:01:00	Broadcast	ARP	60	who has 192.168.0.3? Tell 192.168.0.1
2	0.00000096	02:00:00:00:03:00	02:00:00:00:01:00	ARP	42	192.168.0.3 is at 02:00:00:00:03:00
3	0.00002951	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=1/256, ttl=64 (request in 4)
4	0.00009755	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=1/256, ttl=63 (reply in 3)
5	1.0029803	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=2/512, ttl=64 (request in 6)
6	1.00298253	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=2/512, ttl=63 (reply in 5)
7	2.00459648	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=3/768, ttl=64 (request in 8)
8	2.0045467	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=3/768, ttl=63 (reply in 7)
9	3.00677541	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=4/1024, ttl=64
10	3.00761923	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=4/1024, ttl=63 (reply in 9)
11	4.00841699	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=5/1280, ttl=64 (request in 12)
12	4.00874885	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=5/1280, ttl=63 (reply in 11)
13	5.00974121	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=6/1536, ttl=64 (request in 14)
14	5.01054059	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=6/1536, ttl=63 (reply in 13)
15	5.01856225	02:00:00:00:03:00	02:00:00:00:01:00	ARP	42	who has 192.168.0.1? Tell 192.168.0.3
16	5.01884153	02:00:00:00:01:00	02:00:00:00:03:00	ARP	60	192.168.0.1 is at 02:00:00:00:01:00
17	6.01170216	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=7/1792, ttl=64 (request in 18)
18	6.01255473	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=7/1792, ttl=63 (reply in 17)
19	7.01375767	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=8/2048, ttl=64
20	7.01461235	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=8/2048, ttl=63 (reply in 19)
21	8.01558779	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=9/2304, ttl=64 (request in 22)
22	8.01648628	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=9/2304, ttl=63 (reply in 21)
23	9.01769944	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=10/2560, ttl=64 (request in 24)
24	9.0185503	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=10/2560, ttl=63 (reply in 23)
25	10.0197986	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=11/2816, ttl=64 (request in 26)
26	10.0206974	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=11/2816, ttl=63 (reply in 25)

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
 Ethernet II, Src: 02:00:00:00:01:00 (02:00:00:00:01:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Address Resolution Protocol (request)

```

0000  ff ff ff ff ff 02 00 00 01 00 08 06 00 01  .....
0010  08 00 06 04 00 01 02 00 00 00 c0 a8 00 01  .....
0020  00 00 00 00 00 00 c0 a8 00 00 03 00 00 00  .....
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
  
```

File: *temp\wireshark_pcapng_eth0... | Packets: 50 - Displayed: 50 (100.0%) - Dropped: 0 (0.0%) | Profile: Default | 16:39 Thursday 22 September

ETH1

No.	Time	Source	Destination	Protoc	Length	Info
1	0.00000000	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=1/256, ttl=63 (reply in 4)
2	0.00033144	02:00:00:00:04:00	Broadcast	ARP	60	Who has 172.16.0.1? Tell 172.16.0.2
3	0.00033671	02:00:00:00:00:01	02:00:00:00:04:00	ARP	42	172.16.0.1 is at 02:00:00:00:00:01
4	0.00053530	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=1/256, ttl=64 (request in 1)
5	1.00178786	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=2/512, ttl=63
6	1.00263074	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=2/512, ttl=64 (request in 5)
7	2.00428556	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=3/768, ttl=63 (reply in 8)
8	2.00509598	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=3/768, ttl=64 (request in 7)
9	3.00646304	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=4/1024, ttl=63 (reply in 10)
10	3.00725761	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=4/1024, ttl=64 (request in 9)
11	4.00808902	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=5/1280, ttl=63 (reply in 12)
12	4.00840859	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=5/1280, ttl=64 (request in 11)
13	5.00942882	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=6/1536, ttl=63
14	5.01017975	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=6/1536, ttl=64 (request in 13)
15	6.00605998	02:00:00:00:03:01	02:00:00:00:04:00	ARP	42	Who has 172.16.0.2? Tell 172.16.0.1
16	6.00590872	02:00:00:00:04:00	02:00:00:00:03:01	ARP	60	172.16.0.2 is at 02:00:00:00:04:00
17	6.01138939	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=7/1792, ttl=63 (reply in 18)
18	6.01219559	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=7/1792, ttl=64 (request in 17)
19	7.01344446	192.168.0.1	172.16.0.2	ICMP	98	Echo (ping) request id=0x01a, seq=8/2048, ttl=63 (reply in 20)
20	7.01425024	172.16.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x01a, seq=8/2048, ttl=64 (request in 19)

▶ Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
 ▶ Ethernet II, Src: 02:00:00:00:03:01 (02:00:00:00:03:01), Dst: 02:00:00:00:04:00 (02:00:00:00:04:00)
 ▶ Internet Protocol Version 4, Src: 192.168.0.1 (192.168.0.1), Dst: 172.16.0.2 (172.16.0.2)
 ▶ Internet Control Message Protocol

```

0000  02 00 00 00 04 00 02 00 00 00 03 01 08 00 45 00  .....E:
0010  00 54 7d ed 40 00 3f 01 51 00 c0 a9 00 01 ac 10  .T).Q.....
0020  00 02 08 00 30 5a 0c 1a 00 01 6a 73 2c 63 00 00  ....02...39c...
0030  00 00 00 a1 05 00 00 00 00 00 10 11 12 13 14 15  ....
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25  ....*#%#
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35  6'()*+,-./012345
0060  36 37 67
  
```

File: "tmp/wireshark_pcapng_eth1_... | Packets: 46 - Displayed: 46 (100.0%) - Dropped: 0 (0.0%) | Profile: Default | 16:39 | Thursday 22 September

Configuración dinámica

El protocolo DHCP permite configurar dinámicamente los parámetros de red de una máquina. En esta sección configuraremos Router como servidor DHCP para las dos redes. Aunque DHCP puede incluir muchos parámetros de configuración, en esta práctica sólo fijaremos el encaminador por defecto.

Ejercicio 10 [VM1, VM2, VM4]. Eliminar las direcciones IP de los interfaces (`ip addr del`) de todas las máquinas salvo Router.

Ejercicio 11 [Router]. Configurar el servidor DHCP para las dos redes:

- Editar el fichero `/etc/dhcp/dhcpd.conf` y añadir dos secciones `subnet`, una para cada red, que definan, respectivamente, los rangos de direcciones 192.168.0.50-192.168.0.100 y

172.16.0.50-172.16.0.100. Además, incluir la opción routers con la dirección IP de Router en cada red. Ejemplo:

```

subnet      192.168.0.0          netmask      255.255.255.0      {
    range 192.168.0.11 192.168.0.50;
    option routers 192.168.0.3;
                                option      broadcast-address      192.168.0.255;
}

subnet 172.16.0.0 netmask 255.255.0.0 {
    range 172.16.0.50 172.16.0.100;
    option routers 172.16.0.3;
    option broadcast-address 172.16.255.255;
}

```

- Arrancar el servicio con el comando `sudo service dhcpd start`.

Ejercicio 12 [Router, VM1]. Iniciar una captura de paquetes en Router. Arrancar el cliente DHCP en VM1 con `dhclient -d eth0` y observar el proceso de configuración. Completar la siguiente tabla:

IP Origen	IP Destino	Mensaje DHCP	Opciones DHCP
0.0.0.0	255.255.255.255	DHCP DISCOVER	DHCP Message Type: 1 (Discover) Requested IP Address Parameter Request List End
192.168.0.3	192.168.0.11	DHCP OFFER	DHCP Message Type: 2 (Offer) DHCP Server Identifier (192.168.0.3) IP Address Lease Time (12 hours) SubnetMask (255.255.255.0) BroadcastAddress(192.168.0.255) Router (192.168.0.3) End
0.0.0.0	255.255.255.255	DHCP REQUEST	DHCP Message Type: 3 (Request) Requested IP Address (192.168.0.11) Parameter Request List End
192.168.0.3	192.168.0.11	DHCP ACK	DHCP Type: 5 (ACK) DHCP Server Identifier (192.168.0.3) IP Address Lease Time (12 hours) SubnetMask (255.255.255.0) BroadcastAddress(192.168.0.255) Router (192.168.0.3) End

CAPTURA VM1 DHCLIENT

```

[cursoredes@localhost ~]$ sudo dhclient -d eth0
Internet Systems Consortium DHCP Client 4.2.5
Copyright 2004-2013 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

```


CAPTURA WIRESHARK ROUTER

Ejercicio 13 [VM2, VM4]. Durante el arranque del sistema se pueden configurar automáticamente interfaces según la información almacenada en el disco del servidor (configuración persistente). El fichero `/etc/sysconfig/network-scripts/ifcfg-eth0` configura automáticamente eth0 usando DHCP. Consultar el fichero y comprobar la configuración en VM2 y VM4 usando las órdenes `ifup` e `ifdown`. Verificar la conectividad entre todas las máquinas de las dos redes.

Nota: Para configuración estática, se pueden usar las siguientes opciones:

Estas opciones se describen en detalle en `/usr/share/doc/initscripts-*/sysconfig.txt`.