Práctica 1.1. Protocolo IPv4. Servicio DHCP

Objetivos

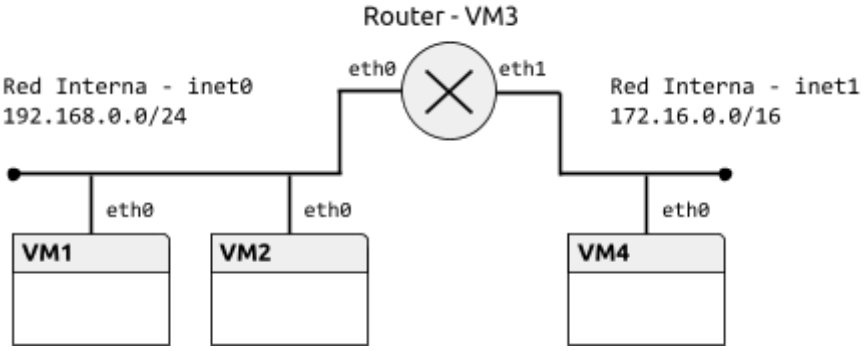
En esta práctica se presentan las herramientas que se utilizarán en la asignatura y se repasan brevemente los aspectos básicos del protocolo IPv4. Además, se analizan las características del protocolo DHCP.

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 `vtopol`, 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.topol` con la topología de la red, que consta de 4 máquinas y dos redes. El contenido del fichero es:

```
netprefix inet
machine 1 0 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 `cursoredes`, con contraseña `cursoredes`.

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`.

```
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 08:00:27:b9:6d:34 brd ff:ff:ff:ff:ff:ff
[cursoredes@localhost ~]$ 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 08:00:27:b9:6d:34 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`.

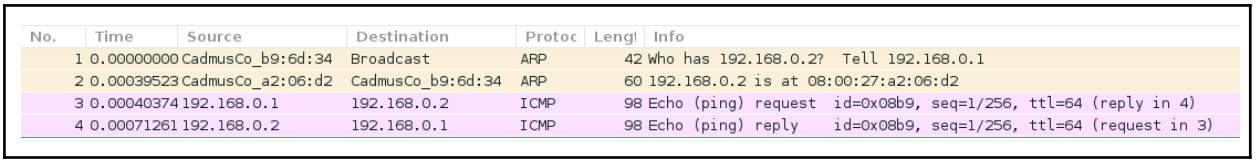
VM1:
ip link set eth0 up
ip address add 192.168.0.1/24 dev eth0
VM2:
sudo ip link set eth0 up
sudo ip address add 192.168.0.2/24 dev eth0
VM3:
sudo ip link set eth0 up
sudo ip address add 192.168.0.3/24 dev eth0

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
08:00:27:b9:6d:34 (VM1)	00:00:00:00:00:00	ARP request	192.168.0.1	Broadcast	"Who has 192.168.0.2? Tell 192.168.0.1"
08:00:27:a2:06:d2 (VM2)	08:00:27:b9:6d:34 (VM1)	ARP reply	192.168.0.2	192.168.0.1	"192.168.0.2 is at 08:00:27:a2:06:d2"
08:00:27:b9:6d:34 (VM1)	08:00:27:a2:06:d2 (VM2)	ICMP	192.168.0.1	192.168.0.2	Echo Request
08:00:27:a2:06:d2 (VM2)	08:00:27:b9:6d:34 (VM1)	ICMP	192.168.0.2	192.168.0.1	Echo Reply



No.	Time	Source	Destination	Protoc	Lengt	Info
1	0.00000000	CadmusCo_b9:6d:34	Broadcast	ARP	42	Who has 192.168.0.2? Tell 192.168.0.1
2	0.00039523	CadmusCo_a2:06:d2	CadmusCo_b9:6d:34	ARP	60	192.168.0.2 is at 08:00:27:a2:06:d2
3	0.00040374	192.168.0.1	192.168.0.2	ICMP	98	Echo (ping) request id=0x08b9, seq=1/256, ttl=64 (reply in 4)
4	0.00071261	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x08b9, seq=1/256, ttl=64 (request in 3)

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.

VM1:
sudo ip neigh
192.168.0.2 dev eth0 lladdr 08:00:27:a2:06:d2 REACHABLE
192.168.0.3 dev eth0 lladdr 08:00:27:48:35:c1 STALE

VM2:
sudo ip neigh

192.168.0.1 dev eth0 lladdr 08:00:27:b9:6d:34 STALE

Reachable*: the neighbour entry is valid until the reachability timeout expires.*

Stale*: the neighbour entry is valid but suspicious. This option to ip neigh does not change the neighbour state if it was valid and the address is not changed by this command.*

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

VM3:
sudo ip link set eth1 up
sudo ip address add 172.16.0.3/16 dev eth1

VM4:
sudo ip link set eth0 up
sudo ip address add 172.16.0.4/16 dev eth0

VM3:
ping 172.16.0.4 -c 1
PING 172.16.0.4 (172.16.0.4) 56(84) bytes of data.
64 bytes from 172.16.0.4: icmp_seq=1 ttl=64 time=0.771 ms

--- 172.16.0.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.771/0.771/0.771/0.000 ms

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 y VM2:
sudo ip route add default via 192.168.0.3

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`.

sudo ip route add 192.168.0.0/24 via 172.16.0.3

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

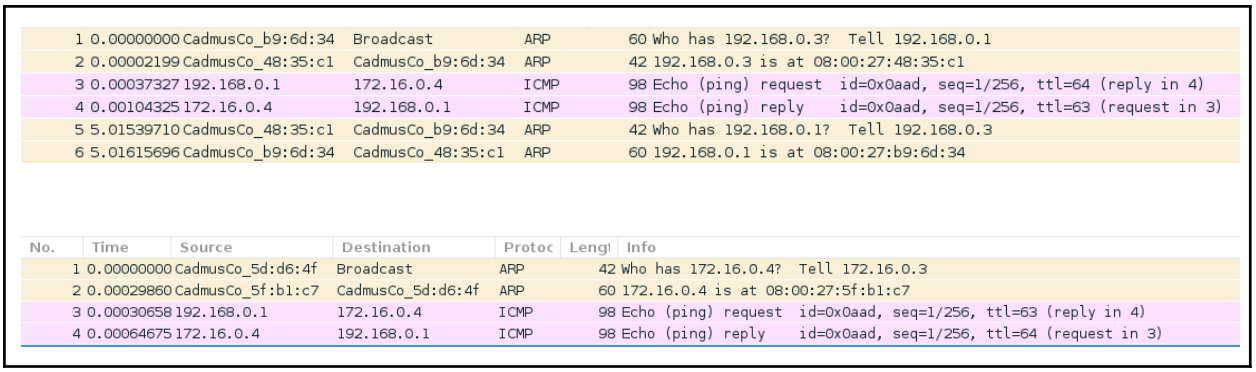
VM4 usando la orden `ping`. Completar la siguiente tabla para todos los paquetes intercambiados hasta la recepción del primer *Echo Reply*.

Red 192.168.0.0/24 - Router (eth0)

MAC origen	MAC destino	Protocolo	IP origen	IP destino	Tipo de mensaje
08:00:27:b9:6d:34 (VM1)	00:00:00:00:00:00	ARP request	192.168.0.1	Broadcast	"Who has 192.168.0.3? Tell 192.168.0.1"
08:00:27:48:35:c1 (VM3)	08:00:27:b9:6d:34 (VM1)	ARP reply	192.168.0.3	192.168.0.1	"192.168.0.3 is at 08:00:27:48:35:c1"
VM1	VM3	ICMP request	192.168.0.1	172.16.0.4	Echo request
VM3	VM1	ICMP reply	172.16.0.4	182.168.0.1	Echo reply

Red 172.16.0.0/16 - Router (eth1)

MAC origen	MAC destino	Protocolo	IP origen	IP destino	Tipo de mensaje
08:00:27:5d:d6:4f (VM3)	00:00:00:00:00:00	ARP request	172.16.0.3	Broadcast	"Who has 172.16.0.4? Tell 172.16.0.3"
08:00:27:5f:b1:c7 (VM4)	08:00:27:5d:d6:4f (VM3)	ARP reply	172.16.0.4	172.16.0.3	"172.16.0.4 is at 08:00:27:5f:b1:c7"
VM3	VM4	ICMP request	192.168.0.1	172.16.0.4	Echo request
VM4	VM3	ICMPreply	172.16.0.4	192.168.0.1	Echo reply



No.	Time	Source	Destination	Protoc	Lengt	Info
1	0.00000000	CadmusCo_b9:6d:34	Broadcast	ARP	60	Who has 192.168.0.3? Tell 192.168.0.1
2	0.00002199	CadmusCo_48:35:c1	CadmusCo_b9:6d:34	ARP	42	192.168.0.3 is at 08:00:27:48:35:c1
3	0.00037327	192.168.0.1	172.16.0.4	ICMP	98	Echo (ping) request id=0x0aad, seq=1/256, ttl=64 (reply in 4)
4	0.00104325	172.16.0.4	192.168.0.1	ICMP	98	Echo (ping) reply id=0x0aad, seq=1/256, ttl=63 (request in 3)
5	5.01539710	CadmusCo_48:35:c1	CadmusCo_b9:6d:34	ARP	42	Who has 192.168.0.1? Tell 192.168.0.3
6	5.01615696	CadmusCo_b9:6d:34	CadmusCo_48:35:c1	ARP	60	192.168.0.1 is at 08:00:27:b9:6d:34

No.	Time	Source	Destination	Protoc	Lengt	Info
1	0.00000000	CadmusCo_5d:d6:4f	Broadcast	ARP	42	Who has 172.16.0.4? Tell 172.16.0.3
2	0.00029860	CadmusCo_5f:b1:c7	CadmusCo_5d:d6:4f	ARP	60	172.16.0.4 is at 08:00:27:5f:b1:c7
3	0.00030658	192.168.0.1	172.16.0.4	ICMP	98	Echo (ping) request id=0x0aad, seq=1/256, ttl=63 (reply in 4)
4	0.00064675	172.16.0.4	192.168.0.1	ICMP	98	Echo (ping) reply id=0x0aad, seq=1/256, ttl=64 (request in 3)

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;
  }
```

```
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.example
#   see dhcpd.conf(5) man page
#

subnet 192.168.0.0 netmask 255.255.255.0 {
   range 192.168.0.50 192.168.0.100;
   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;
}
```

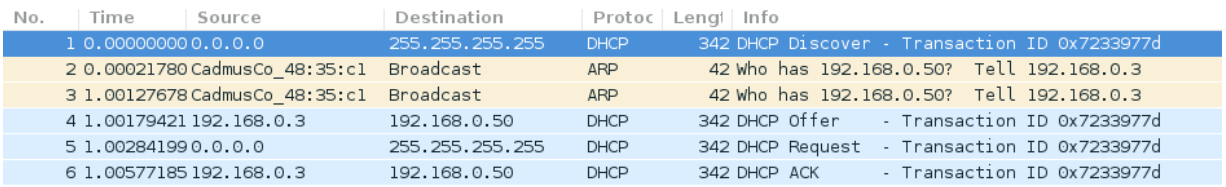
- Arrancar el servicio con el comando `sudo service dhcpd start`.

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*	*Discover*	*Requested IP Address: 10.0.2.15 (10.0.2.15)*
192.168.0.3	*192.168.0.50*	*Offer*	
0.0.0.0	*255.255.255.255*	*Request*	*Requested IP Address: 192.168.0.50 (192.168.0.50)*
192.168.0.3	*192.168.0.50*	*Ack*	

```
Listening on LPF/eth0/08:00:27:b9:6d:34
Sending on   LPF/eth0/08:00:27:b9:6d:34
Sending on   Socket/fallback
```

DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 7 (xid=0x7d973372)
DHCPREQUEST on eth0 to 255.255.255.255 port 67 (xid=0x7d973372)
DHCPOFFER from 192.168.0.3
DHCPACK from 192.168.0.3 (xid=0x7d973372)
bound to 192.168.0.50 -- renewal in 21593 seconds.



No.	Time	Source	Destination	Protoc	Lengt	Info
1	0.00000000	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x7233977d
2	0.00021780	CadmusCo_48:35:c1	Broadcast	ARP	42	Who has 192.168.0.50? Tell 192.168.0.3
3	1.00127678	CadmusCo_48:35:c1	Broadcast	ARP	42	Who has 192.168.0.50? Tell 192.168.0.3
4	1.00179421	192.168.0.3	192.168.0.50	DHCP	342	DHCP Offer - Transaction ID 0x7233977d
5	1.00284199	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x7233977d
6	1.00577185	192.168.0.3	192.168.0.50	DHCP	342	DHCP ACK - Transaction ID 0x7233977d

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:

```
TYPE=Ethernet
BOOTPROTO=none
IPADDR=<dirección IP estática>
PREFIX=<tamaño del prefijo de red>
GATEWAY=<dirección IP estática del encaminador por defecto (si existe)>
DEVICE=eth0
```

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

VM2:

```
TYPE=Ethernet
BOOTPROTO=none
IPADDR=192.168.0.2
PREFIX=24
GATEWAY=192.168.0.3
DEVICE=eth0
```

VM4:

```
TYPE=Ethernet
BOOTPROTO=none
IPADDR=172.16.0.4
PREFIX=16
GATEWAY=172.16.0.3
DEVICE=eth0
```

Ifup eth0 para cargar config de fichero, ifdown para desconfigurar.
Todas las máquinas conectan.