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](#_14a3ftqman5y)

[Configuración estática](#_7799bdv34ent)

[Encaminamiento estático](#_w2ynjijgd1kw)

[Configuración dinámica](#_nj2ry1q77ael)

# 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/ |
| --- |

1. 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](https://drive.google.com/file/d/18AvKwF3SuwXUpQvmzYMDp3Cagcrnbec7/view?usp=sharing)

1. 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> ... |
| --- |

1. 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](https://drive.google.com/file/d/0B3JrkzcA788mZTAtX0VQWVlrN2c/view?usp=sharing&resourcekey=0-kx5XvWNjkxOFuJ4RxFTd7A), 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.

| *[cursoredes@localhost ~]$ 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,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000*  *link/ether 08:00:27:b0:12:f8 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,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP mode DEFAULT group default qlen 1000*  *link/ether 08:00:27:b0:12:f8 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***  *[root@localhost ~]# ip address add 192.168.0.1/24 dev eth0*  *[root@localhost ~]# ip link set dev eth0 up*  ***VM2***  *[root@localhost ~]# ip address add 192.168.0.2/24 dev eth0*  *[root@localhost ~]# ip link set dev eth0 up*  ***Router***  *[root@localhost ~]# ip address add 192.168.0.3/24 dev eth0*  *[root@localhost ~]# ip address add 172.16.0.1/16 dev eth1*  *[root@localhost ~]# ip link set dev eth0 up*  *[root@localhost ~]# ip link set dev eth1 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** |
| --- | --- | --- | --- | --- | --- |
| 08:00:27:24:16:3c | ff:ff:ff:ff:ff:ff (Broadcast) | ARP | 192.168.0.2 | 192.168.0.1 | Who has 192.168.0.1? Tell 192.168.0.2 |
| 08:00:27:b0:12:f8 | 08:00:27:24:16:3c | ARP | 192.168.0.1 | 192.168.0.2 | MAC de 192.168.0.1 |
| 08:00:27:24:16:3c | 08:00:27:b0:12:f8 | ICMP | 192.168.0.2 | 192.168.0.1 | Echo ping request |
| 08:00:27:b0:12:f8 | 08:00:27:24:16:3c | ICMP | 192.168.0.1 | 192.168.0.2 | Echo ping reply |

|  |
| --- |

***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***  *192.168.0.2 dev eth0 lladdr 08:00:27:24:16:3c REACHABLE*  *192.168.0.2 -> La dirección IP del vecino*  *dev etho -> La interfaz a la que esta conectado el vecino*  *lladdr 08:00:27:24:16:3c -> La dirección de la capa de enlace del vecino*  *REACHABLE -> La entrada del vecino es valida hasta que expire el timeout*  ***VM2***  *192.168.0.1 dev eth0 lladdr 08:00:27:b0:12:f8 DELAY*  *192.168.0.1 ->La dirección IP del vecino*  *dev etho -> La interfaz a la que esta conectado el vecino*  *lladdr 08:00:27:b0:12:f8 -> La dirección de la capa de enlace del vecino*  *DELAY -> La validación de la entrada del vecino esta retrasada* |
| --- |

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

| ***VM4***  *[root@localhost ~]# ip address add 172.16.0.2/16 dev eth0*  *[root@localhost ~]# ip link set dev eth0 up*  *[root@localhost ~]# ping 172.16.0.1*  *PING 172.16.0.1 (172.16.0.1) 56(84) bytes of data.*  *64 bytes from 172.16.0.1: icmp\_seq=1 ttl=64 time=1.03 ms*  *64 bytes from 172.16.0.1: icmp\_seq=2 ttl=64 time=0.626 ms*  *64 bytes from 172.16.0.1: icmp\_seq=3 ttl=64 time=0.560 ms*  *^C*  *--- 172.16.0.1 ping statistics ---*  *3 packets transmitted, 3 received, 0% packet loss, time 2003ms*  *rtt min/avg/max/mdev = 0.560/0.741/1.037/0.211 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.

| *[root@localhost ~]# 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.

| *[root@localhost ~]# 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** |
| --- | --- | --- | --- | --- | --- |
| 08:00:27:b0:12:f8 | Broadcast | ARP | 192.168.0.1 | 192.168.0.3 | Who has 192.168.0.3?  Tell 192.168.0.1 |
| 08:00:27:07:61:7f | 08:00:27:b0:12:f8 | ARP | 192.168.0.3 | 192.168.0.1 | 192.168.0.3 is at  08:00:27:07:61:7f |
| 08:00:27:b0:12:f8 | 08:00:27:07:61:7f | ICMP | 192.168.0.1 | 172.16.0.2 | Echo (ping) request |
| 08:00:27:07:61:7f | 08:00:27:b0:12:f8 | ICMP | 172.16.0.2 | 192.168.0.1 | Echo (ping) reply |

**Red 172.16.0.0/16 - Router (eth1)**

| **MAC origen** | **MAC destino** | **Protocolo** | **IP origen** | **IP destino** | **Tipo de mensaje** |
| --- | --- | --- | --- | --- | --- |
| 08:00:27:7d:ad:7e | Broadcast | ARP | 172.16.0.1 | 172.16.0.2 | Who has 172.16.0.2?  Tell 172.16.0.1 |
| 08:00:27:22:79:cd | 08:00:27:7d:ad:7e | ARP | 172.16.0.2 | 172.16.0.1 | 172.16.0.2 is at  08:00:27:22:79:cd |
| 08:00:27:7d:ad:7e | 08:00:27:22:79:cd | ICMP | 192.168.0.1 | 172.16.0.2 | Echo (ping) request |
| 08:00:27:22:79:cd | 08:00:27:7d:ad:7e | ICMP | 172.16.0.2 | 192.168.0.1 | Echo (ping) reply |

| *Red 192.168.0.0/24 - Router (eth0)*    *Red 172.16.0.0/16 - Router (eth1)* |
| --- |

# 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 direcciones192.168.0.50-192.168.0.100 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; } |
| --- |

* 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* | *Option : (53) DHCP Message Type*  *Option: (50) Requested IP*  *Option : (55) Parameter Request List*  *Option: (255) End* |
| *192.168.0.3* | *192.168.0.50* | *DHCP Offer* | *Option: (1) Subnet Mask*  *Option: (28) Broadcast Address*  *Option: (3) Router*  *Option: (255) End* |
| *0.0.0.0* | *255.255.255.255* | *DHCP Request* | *Option: (53) DHCP Message Type*  *Option : (54) DHCP Server Identifier*  *Option: (50) Requested IP Address*  *Option: (55) Parameter Request List*  *Option: (255) End* |
| *192.168.0.3* | *192.168.0.50* | *DHCP ACK* | *Option: (1) Subnet Mask*  *Option: (28) Broadcast Address*  *Option: (3) Router*  *Option: (255) End* |

| *[root@localhost ~]# 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/*  *Listening on LPF/eth0/08:00:27:b0:12:f8*  *Sending on LPF/eth0/08:00:27:b0:12:f8*  *Sending on Socket/fallback*  *DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 5 (xid=0x6f6f5278)*  *DHCPREQUEST on eth0 to 255.255.255.255 port 67 (xid=0x6f6f5278)*  *DHCPOFFER from 192.168.0.3*  *DHCPACK from 192.168.0.3 (xid=0x6f6f5278)*  *bound to 192.168.0.50 -- renewal in 16836 seconds.* |
| --- |

***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.