

# Práctica 1.4. Protocolo IPv6

## Objetivos

En esta práctica se estudian los aspectos básicos del protocolo IPv6, el manejo de los diferentes tipos de direcciones y mecanismos de configuración. Además se analizarán las características más importantes del protocolo ICMP versión 6.



Activar el **portapapeles bidireccional** (menú Dispositivos) en las máquinas virtuales.

Usar la opción de Virtualbox (menú Ver) para realizar **capturas de pantalla**.

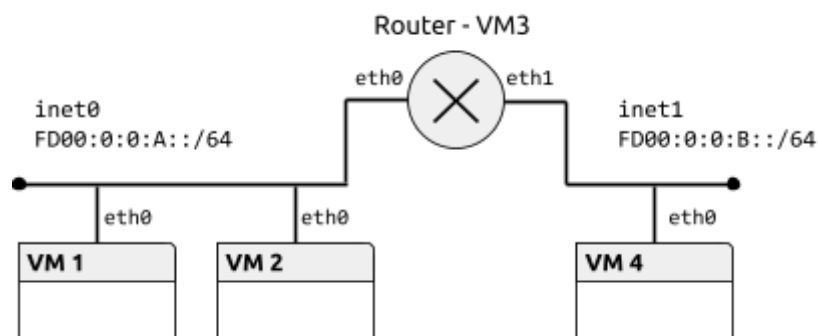
La **contraseña** del usuario cursoredes es cursoredes.

## Contenidos

- Preparación del entorno para la práctica
- Direcciones de enlace local
- Direcciones ULA
- Encaminamiento estático
- Configuración persistente
- Autoconfiguración. Anuncio de prefijos
- ICMPv6

## Preparación del entorno para la práctica

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



El fichero de configuración de la topología tendría el siguiente contenido:

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

## Direcciones de enlace local

Una dirección de enlace local es únicamente válida en la subred que está definida. Ningún encaminador dará salida a un datagrama con una dirección de enlace local como destino. El prefijo de formato para estas direcciones es fe80::/10.

**Ejercicio 1 [VM1, VM2].** Activar el interfaz eth0 en VM1 y VM2. Comprobar las direcciones de enlace local que tienen asignadas con el comando ip.

**Ejercicio 2 [VM1, VM2].** Comprobar la conectividad entre VM1 y VM2 con la orden ping6 (o ping -6). Cuando se usan direcciones de enlace local, y **sólo en ese caso**, es necesario especificar el interfaz origen, añadiendo %<nombre\_interfaz> a la dirección. Consultar las opciones del comando ping6 en la página de manual. Observar el tráfico generado con Wireshark, especialmente los protocolos encapsulados en cada datagrama y los parámetros del protocolo IPv6.

*Copiar el comando utilizados y su salida. Copiar una captura de pantalla de Wireshark donde se vean los campos de la cabecera IPv6.*

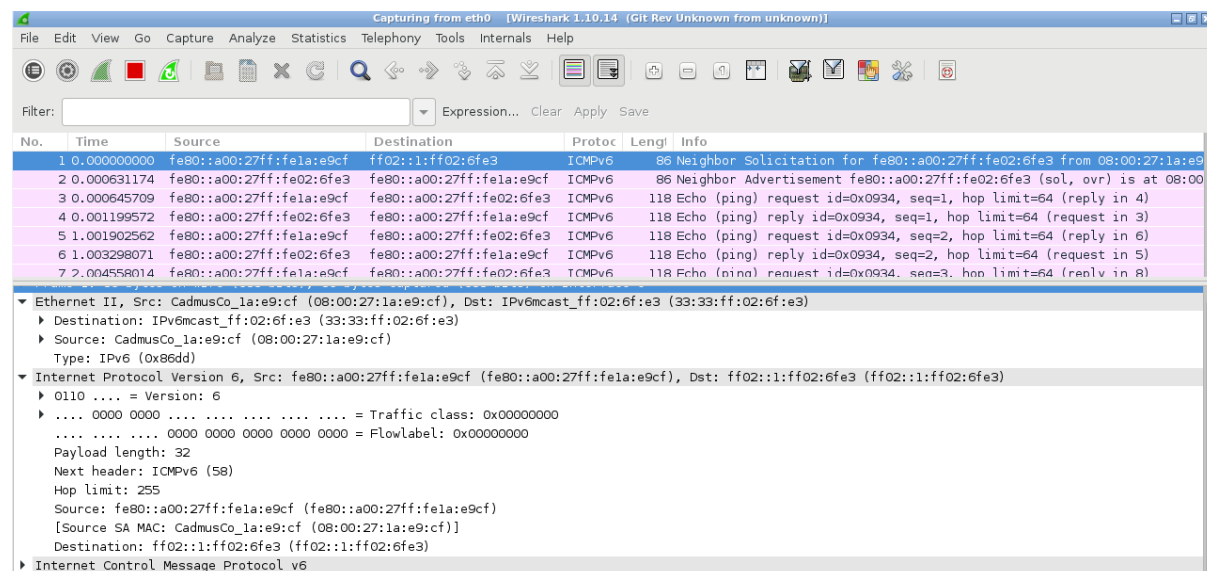
#### Direcciones IP obtenidas con el comando ip addr

VM1 (eth0): fe80::a00:27ff:fe1a:e9cf/64

VM2 (eth0): fe80::a00:27ff:fe02:6fe3/64

#### Ping de VM1 a VM2:

```
[cursoredes@localhost ~]$ ping6 fe80::a00:27ff:fe02:6fe3%eth0
PING fe80::a00:27ff:fe02:6fe3%eth0(fe80::a00:27ff:fe02:6fe3%eth0) 56 data bytes
64 bytes from fe80::a00:27ff:fe02:6fe3%eth0: icmp_seq=1 ttl=64 time=0.652 ms
64 bytes from fe80::a00:27ff:fe02:6fe3%eth0: icmp_seq=2 ttl=64 time=1.15 ms
64 bytes from fe80::a00:27ff:fe02:6fe3%eth0: icmp_seq=3 ttl=64 time=1.10 ms
64 bytes from fe80::a00:27ff:fe02:6fe3%eth0: icmp_seq=4 ttl=64 time=1.33 ms
64 bytes from fe80::a00:27ff:fe02:6fe3%eth0: icmp_seq=5 ttl=64 time=1.23 ms
^C
--- fe80::a00:27ff:fe02:6fe3%eth0 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 0.652/1.098/1.338/0.238 ms
```



**Ejercicio 3 [Router, VM4].** Activar el interfaz de VM4 y los dos interfaces de Router. Comprobar la conectividad entre Router y VM1, y entre Router y VM4 usando la dirección de enlace local.

*Copiar los comandos utilizados y su salida.*

#### Direcciones IP obtenidas con el comando ip addr

VM3 (eth0): fe80::a00:27ff:fe6b:48b7/64

VM3 (eth1): fe80::a00:27ff:fee3:29bb/64

VM4 (eth0): fe80::a00:27ff:fe51:e06c/64

#### **Conectividad Router - VM1:**

```
[cursoredes@localhost ~]$ ping6 fe80::a00:27ff:fe1a:e9cf%eth0
PING fe80::a00:27ff:fe1a:e9cf%eth0(fe80::a00:27ff:fe1a:e9cf%eth0) 56 data bytes
64 bytes from fe80::a00:27ff:fe1a:e9cf%eth0: icmp_seq=1 ttl=64 time=1.26 ms
64 bytes from fe80::a00:27ff:fe1a:e9cf%eth0: icmp_seq=2 ttl=64 time=1.14 ms
64 bytes from fe80::a00:27ff:fe1a:e9cf%eth0: icmp_seq=3 ttl=64 time=1.29 ms
64 bytes from fe80::a00:27ff:fe1a:e9cf%eth0: icmp_seq=4 ttl=64 time=1.39 ms
64 bytes from fe80::a00:27ff:fe1a:e9cf%eth0: icmp_seq=5 ttl=64 time=1.49 ms
^C
--- fe80::a00:27ff:fe1a:e9cf%eth0 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 1.143/1.319/1.496/0.128 ms
```

#### **Conectividad Router - VM2:**

```
[cursoredes@localhost ~]$ ping6 fe80::a00:27ff:fe51:e06c%eth1
PING fe80::a00:27ff:fe51:e06c%eth1(fe80::a00:27ff:fe51:e06c%eth1) 56 data bytes
64 bytes from fe80::a00:27ff:fe51:e06c%eth1: icmp_seq=1 ttl=64 time=1.32 ms
64 bytes from fe80::a00:27ff:fe51:e06c%eth1: icmp_seq=2 ttl=64 time=1.29 ms
64 bytes from fe80::a00:27ff:fe51:e06c%eth1: icmp_seq=3 ttl=64 time=1.31 ms
64 bytes from fe80::a00:27ff:fe51:e06c%eth1: icmp_seq=4 ttl=64 time=1.21 ms
64 bytes from fe80::a00:27ff:fe51:e06c%eth1: icmp_seq=5 ttl=64 time=1.18 ms
^C
--- fe80::a00:27ff:fe51:e06c%eth1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 1.188/1.267/1.325/0.058 ms
```

**Para saber más...** En el protocolo IPv4 también se reserva el bloque 169.254.0.0/16 para direcciones de enlace local, cuando no es posible la configuración de los interfaces por otras vías. Los detalles se describen en el RFC 3927.

## Direcciones ULA

Una dirección ULA (*Unique Local Address*) puede usarse dentro de una organización, de forma que los encaminadores internos del sitio deben encaminar los datagramas con una dirección ULA como destino. El prefijo de formato para estas direcciones es fc00::/7.

**Ejercicio 4 [VM1, VM2].** Configurar VM1 y VM2 para que tengan una dirección ULA en la red fd00:0:0:a::/64 con el comando ip. La parte de identificador de interfaz puede elegirse libremente, siempre que no coincida para ambas máquinas. Incluir la longitud del prefijo al fijar las direcciones.

*Copiar los comandos utilizados.*

#### **VM1**

```
sudo ip address add fd00:0:0:a::1/64 dev eth0
```

#### **VM2**

```
sudo ip address add fd00:0:0:a::2/64 dev eth0
```

**Ejercicio 5 [VM1, VM2].** Comprobar la conectividad entre VM1 y VM2 con la orden ping6 usando la nueva dirección. Observar los mensajes intercambiados con Wireshark.

**Ejercicio 6 [Router, VM4].** Configurar direcciones ULA en los dos interfaces de Router (redes

fd00:0:0:a::/64 y fd00:0:0:b::/64) y en el de VM4 (red fd00:0:0:b::/64). Elegir el identificador de interfaz de forma que no coincida dentro de la misma red.

*Copiar los comandos utilizados.*

#### **VM3**

```
sudo ip address add fd00:0:0:a::3/64 dev eth0
sudo ip address add fd00:0:0:b::3/64 dev eth1
```

#### **VM4**

```
sudo ip address add fd00:0:0:b::4/64 dev eth0
```

**Ejercicio 7 [Router].** Comprobar la conectividad entre Router y VM1, y entre Router y VM4 usando direcciones ULA. Comprobar además que VM1 no puede alcanzar a VM4.

*Copiar los comandos utilizados.*

#### **Conectividad Router - VM1:**

```
[cursoredes@localhost ~]$ ping6 fd00:0:0:a::1
PING fd00:0:0:a::1(fd00:0:0:a::1) 56 data bytes
64 bytes from fd00:0:0:a::1: icmp_seq=1 ttl=64 time=1.11 ms
64 bytes from fd00:0:0:a::1: icmp_seq=2 ttl=64 time=1.38 ms
64 bytes from fd00:0:0:a::1: icmp_seq=3 ttl=64 time=1.37 ms
64 bytes from fd00:0:0:a::1: icmp_seq=4 ttl=64 time=1.48 ms
64 bytes from fd00:0:0:a::1: icmp_seq=5 ttl=64 time=1.37 ms
^C
--- fd00:0:0:a::1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 1.118/1.348/1.483/0.121 ms
```

#### **Conectividad Router - VM4:**

```
[cursoredes@localhost ~]$ ping6 fd00:0:0:b::4
PING fd00:0:0:b::4(fd00:0:0:b::4) 56 data bytes
64 bytes from fd00:0:0:b::4: icmp_seq=1 ttl=64 time=1.23 ms
64 bytes from fd00:0:0:b::4: icmp_seq=2 ttl=64 time=1.44 ms
64 bytes from fd00:0:0:b::4: icmp_seq=3 ttl=64 time=1.19 ms
64 bytes from fd00:0:0:b::4: icmp_seq=4 ttl=64 time=1.52 ms
64 bytes from fd00:0:0:b::4: icmp_seq=5 ttl=64 time=1.23 ms
^C
--- fd00:0:0:b::4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 1.199/1.326/1.528/0.140 ms
```

#### **Conectividad VM1 - VM4:**

```
[cursoredes@localhost ~]$ ping6 fd00:0:0:b::4
connect: Network is unreachable
```

## **Encaminamiento estático**

Según la topología que hemos configurado en esta práctica, Router debe encaminar el tráfico entre las redes fd00:0:0:a::/64 y fd00:0:0:b::/64. En esta sección vamos a configurar un encaminamiento estático basado en las rutas que fijaremos manualmente en todas las máquinas.

**Ejercicio 8 [VM1, Router].** Consultar las tablas de rutas en VM1 y Router con el comando `ip route`. Consultar la página de manual del comando para seleccionar las rutas IPv6.

**Ejercicio 9 [Router].** Para que Router actúe efectivamente como encaminador, hay que activar el reenvío de paquetes (*packet forwarding*). De forma temporal, se puede activar con el comando `sysctl -w net.ipv6.conf.all.forwarding=1`.

**Ejercicio 10 [VM1, VM2, VM4].** Finalmente, hay que configurar la tabla de rutas en las máquinas virtuales. Añadir la dirección correspondiente de Router como ruta por defecto con el comando `ip route`. Comprobar la conectividad entre VM1 y VM4 usando el comando `ping6`.

*Copiar los comandos utilizados y su salida.*

#### VM1 y VM2

```
sudo ip route add default via fd00:0:0:a::3
```

#### VM4

```
sudo ip route add default via fd00:0:0:b::3
```

#### Conectividad VM1 - VM4

```
[cursoredes@localhost ~]$ ping6 fd00:0:0:b::4
PING fd00:0:0:b::4(fd00:0:0:b::4) 56 data bytes
64 bytes from fd00:0:0:b::4: icmp_seq=1 ttl=63 time=1.20 ms
64 bytes from fd00:0:0:b::4: icmp_seq=2 ttl=63 time=2.02 ms
64 bytes from fd00:0:0:b::4: icmp_seq=3 ttl=63 time=2.14 ms
64 bytes from fd00:0:0:b::4: icmp_seq=4 ttl=63 time=2.24 ms
64 bytes from fd00:0:0:b::4: icmp_seq=5 ttl=63 time=2.26 ms
^C
--- fd00:0:0:b::4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 1.202/1.977/2.262/0.396 ms
```

**Ejercicio 11 [VM1, Router, VM4].** Abrir Wireshark en Router e iniciar dos capturas, una en cada interfaz de red. Borrar la tabla de vecinos en VM1 y Router (con `ip neigh flush dev <interfaz>`). Usar la orden `ping6` entre VM1 y VM4. Completar la siguiente tabla con todos los mensajes hasta el primer ICMP Echo Reply:

#### Red fd00:0:0:a::/64 - Router (eth0)

MAC Origen	MAC Destino	IPv6 Origen	IPv6 Destino	ICMPv6 Tipo
08:00:27:1a:e9:cf	33:33:ff:00:00:03 (Broadcast)	fd00:0:0:a::1	ff02::1:ff00:3 (Multicast)	Neighbor Solicitation
08:00:27:6b:48:b7	08:00:27:1a:e9:cf	fd00:0:0:a::3	fd00:0:0:a::1	Neighbor Advertisement
08:00:27:1a:e9:cf	08:00:27:6b:48:b7	fd00:0:0:a::1	fd00:0:0:b::4	Echo Request
08:00:27:6b:48:b7	08:00:27:1a:e9:cf	fd00:0:0:b::4	fd00:0:0:a::1	Echo Reply

#### Red fd00:0:0:b::/64 - Router (eth1)

MAC Origen	MAC Destino	IPv6 Origen	IPv6 Destino	ICMPv6 Tipo
08:00:27:e3:29:bb	33:33:ff:00:00:04 (Broadcast)	fe80::a00:27ff: :fee3:29bb	ff02::1:ff00:4 (Multicast)	Neighbor Solicitation
08:00:27:51:e0:6c	08:00:27:e3:29:bb	fd00:0:0:b::4	fe80::a00:27ff:f ee3:29bb	Neighbor Advertisement

08:00:27:e3:29:bb	08:00:27:51:e0:6c	fd00:0:0:a::1	fd00:0:0:b::4	Echo Request
08:00:27:51:e0:6c	08:00:27:e3:29:bb	fd00:0:0:b::4	fd00:0:0:a::1	Echo Reply

*Copiar dos capturas de pantalla de Wireshark.*

## Eth0

VM3 [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Capturing from eth0 [Wireshark 1.10.14 (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	Protocol	Length	Info
1	0.000000000	fd00:0:0:a::1	ff02::1:ff00:3	ICMPv6	86	Neighbor Solicitation for fd00:0:0:a::3 from 08:00:27:1a:e9:cf
2	0.000043996	fd00:0:0:a::3	fd00:0:0:a::1	ICMPv6	86	Neighbor Advertisement fd00:0:0:a::3 (rtr, sol, ovr) is at 08:00:27:6b:48:b7
3	0.000065405	fe80::a00:27ff:fe1a:e9cf	ff02::1:ff00:3	ICMPv6	86	Neighbor Solicitation for fd00:0:0:a::3 from 08:00:27:1a:e9:cf
4	0.000074613	fd00:0:0:a::3	fe80::a00:27ff:fe1a:e9cf	ICMPv6	86	Neighbor Advertisement fd00:0:0:a::3 (rtr, sol, ovr) is at 08:00:27:6b:48:b7
5	0.000084388	fe80::a00:27ff:fe1a:e9cf	ff02::1:ff00:3	ICMPv6	86	Neighbor Solicitation for fd00:0:0:a::3 from 08:00:27:1a:e9:cf
6	0.000091554	fd00:0:0:a::3	fe80::a00:27ff:fe1a:e9cf	ICMPv6	86	Neighbor Advertisement fd00:0:0:a::3 (rtr, sol, ovr) is at 08:00:27:6b:48:b7
7	0.000485019	fd00:0:0:a::1	fd00:0:0:b::4	ICMPv6	118	Echo (ping) request id=0xb6d, seq=1, hop limit=64 (reply in 8)
8	0.001700569	fd00:0:0:b::4	fd00:0:0:a::1	ICMPv6	118	Echo (ping) reply id=0xb6d, seq=1, hop limit=63 (request in 7)
9	1.002940085	fd00:0:0:a::1	fd00:0:0:b::4	ICMPv6	118	Echo (ping) request id=0xb6d, seq=2, hop limit=64 (reply in 10)
10	1.004375967	fd00:0:0:b::4	fd00:0:0:a::1	ICMPv6	118	Echo (ping) reply id=0xb6d, seq=2, hop limit=63 (request in 9)
11	2.005090234	fd00:0:0:a::1	fd00:0:0:b::4	ICMPv6	118	Echo (ping) request id=0xb6d, seq=3, hop limit=64 (reply in 12)
12	2.006432871	fd00:0:0:b::4	fd00:0:0:a::1	ICMPv6	118	Echo (ping) reply id=0xb6d, seq=3, hop limit=63 (request in 11)
13	5.013746473	fe80::a00:27ff:fe6b:48b7	fd00:0:0:a::1	ICMPv6	86	Neighbor Solicitation for fd00:0:0:a::1 from 08:00:27:6b:48:b7
14	5.013865815	fe80::a00:27ff:fe6b:48b7	fe80::a00:27ff:fe1a:e9cf	ICMPv6	86	Neighbor Solicitation for fe80::a00:27ff:fe1a:e9cf from 08:00:27:6b:48:b7
15	5.015015563	fd00:0:0:a::1	fe80::a00:27ff:fe6b:48b7	ICMPv6	78	Neighbor Advertisement fd00:0:0:a::1 (sol)

▶ Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0

▼ Ethernet II, Src: CadmusCo\_la:e9:cf (08:00:27:1a:e9:cf), Dst: IPv6mcast\_ff:00:00:03 (33:33:ff:00:00:03)

▶ Destination: IPv6mcast\_ff:00:00:03 (33:33:ff:00:00:03)

▶ Source: CadmusCo\_la:e9:cf (08:00:27:1a:e9:cf)

Type: IPv6 (0x86dd)

▼ Internet Protocol Version 6, Src: fd00:0:0:a::1 (fd00:0:0:a::1), Dst: ff02::1:ff00:3 (ff02::1:ff00:3)

▶ 0110 .... = Version: 6

▶ .... 0000 0000 .... = Traffic class: 0x00000000

.... 0000 0000 0000 0000 0000 = FlowLabel: 0x00000000

Payload length: 32

Next header: ICMPv6 (58)

Hop limit: 255

Source: fd00:0:0:a::1 (fd00:0:0:a::1)

Destination: ff02::1:ff00:3 (ff02::1:ff00:3)

▶ Internet Control Message Protocol v6

## Eth1

VM3 [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Capturing from eth1 [Wireshark 1.10.14 (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	Lengt	Info
1	0.000000000	fe80::a00:27ff:fee3:29bb	ff02::1:ff00:4	ICMPv6	86	Neighbor Solicitation for fd00:0:0:b::4 from 08:00:27:e3:29:bb
2	0.000362865	fd00:0:0:b::4	fe80::a00:27ff:fee3:29bb	ICMPv6	86	Neighbor Advertisement fd00:0:0:b::4 (sol, ovr) is at 08:00:27:51:e0:6c
3	0.000371228	fd00:0:0:a::1	fd00:0:0:b::4	ICMPv6	118	Echo (ping) request id=0xd01, seq=1, hop limit=63 (reply in 4)
4	0.000725365	fd00:0:0:b::4	fd00:0:0:a::1	ICMPv6	118	Echo (ping) reply id=0xd01, seq=1, hop limit=64 (request in 3)
5	1.001514236	fd00:0:0:a::1	fd00:0:0:b::4	ICMPv6	118	Echo (ping) request id=0xd01, seq=2, hop limit=63 (reply in 6)
6	1.001918779	fd00:0:0:b::4	fd00:0:0:a::1	ICMPv6	118	Echo (ping) reply id=0xd01, seq=2, hop limit=64 (request in 5)
7	2.001958681	fd00:0:0:a::1	fd00:0:0:b::4	ICMPv6	118	Echo (ping) request id=0xd01, seq=3, hop limit=63 (reply in 8)
8	2.002400446	fd00:0:0:b::4	fd00:0:0:a::1	ICMPv6	118	Echo (ping) reply id=0xd01, seq=3, hop limit=64 (request in 7)
9	5.011162180	fe80::a00:27ff:fe51:e06c	fe80::a00:27ff:fee3:29bb	ICMPv6	86	Neighbor Solicitation for fe80::a00:27ff:fee3:29bb from 08:00:27:51:e0:6c
10	5.011190919	fe80::a00:27ff:fee3:29bb	fe80::a00:27ff:fe51:e06c	ICMPv6	78	Neighbor Advertisement fe80::a00:27ff:fee3:29bb (rtr, sol)
11	5.011203822	fe80::a00:27ff:fe51:e06c	fd00:0:0:b::3	ICMPv6	86	Neighbor Solicitation for fd00:0:0:b::3 from 08:00:27:51:e0:6c
12	5.011207746	fd00:0:0:b::3	fe80::a00:27ff:fe51:e06c	ICMPv6	78	Neighbor Advertisement fd00:0:0:b::3 (rtr, sol)
13	10.016954114	fe80::a00:27ff:fee3:29bb	fe80::a00:27ff:fe51:e06c	ICMPv6	86	Neighbor Solicitation for fe80::a00:27ff:fe51:e06c from 08:00:27:e3:29:bb
14	10.017404966	fe80::a00:27ff:fe51:e06c	fe80::a00:27ff:fee3:29bb	ICMPv6	78	Neighbor Advertisement fe80::a00:27ff:fe51:e06c (sol)

▶ Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0

▼ Ethernet II, Src: CadmusCo\_e3:29:bb (08:00:27:e3:29:bb), Dst: IPv6mcast\_ff:00:00:04 (33:33:ff:00:00:04)

▶ Destination: IPv6mcast\_ff:00:00:04 (33:33:ff:00:00:04)

▶ Source: CadmusCo\_e3:29:bb (08:00:27:e3:29:bb)

Type: IPv6 (0x86dd)

▼ Internet Protocol Version 6, Src: fe80::a00:27ff:fee3:29bb (fe80::a00:27ff:fee3:29bb), Dst: ff02::1:ff00:4 (ff02::1:ff00:4)

▶ 0110 .... = Version: 6

▶ .... 0000 0000 .... = Traffic class: 0x00000000

.... 0000 0000 0000 0000 0000 = FlowLabel: 0x00000000

Payload length: 32

Next header: ICMPv6 (58)

Hop limit: 255

Source: fe80::a00:27ff:fee3:29bb (fe80::a00:27ff:fee3:29bb)

[Source SA MAC: CadmusCo\_e3:29:bb (08:00:27:e3:29:bb)]

Destination: ff02::1:ff00:4 (ff02::1:ff00:4)

▶ Internet Control Message Protocol v6

## Configuración persistente

Las configuraciones realizadas en los apartados anteriores son volátiles y desaparecen cuando se reinician las máquinas. Durante el arranque del sistema se pueden configurar automáticamente los interfaces según la información almacenada en el disco.

**Ejercicio 12 [Router].** Crear los ficheros ifcfg-eth0 e ifcfg-eth1 en el directorio /etc/sysconfig/network-scripts/ con la configuración de cada interfaz. Usar las siguientes opciones (descritas en /usr/share/doc/initscripts-\*/sysconfig.txt):

```
TYPE=Ethernet
BOOTPROTO=none
IPV6ADDR=<dirección IP en formato CIDR>
IPV6_DEFAULTGW=<dirección IP del encaminador por defecto (en este caso, no tiene)>
DEVICE=<nombre del interfaz>
```

*Copiar el contenido de los ficheros.*

### **Fichero ifcfg-eth0:**

```
TYPE=Ethernet
BOOTPROTO=none
IPV6ADDR=fd00:0:0:a::3/64
DEVICE=eth0
```

### **Fichero ifcfg-eth1:**

```
TYPE=Ethernet
BOOTPROTO=none
IPV6ADDR=fd00:0:0:b::3/64
DEVICE=eth1
```

**Ejercicio 13 [Router].** Comprobar la configuración persistente con las órdenes ifup e ifdown.

*Copiar los comandos utilizados y su salida.*

### **Comandos ifdown e ifup:**

```
[cursoredes@localhost ~]$ sudo ifdown eth0
[cursoredes@localhost ~]$ sudo ifdown eth1
[cursoredes@localhost ~]$ sudo ifup eth0
INFO : [ipv6_wait_tentative] Waiting for interface eth0 IPv6 address(es) to leave the 'tentative' state
INFO : [ipv6_wait_tentative] Waiting for interface eth0 IPv6 address(es) to leave the 'tentative' state
[cursoredes@localhost ~]$ sudo ifup eth1
INFO : [ipv6_wait_tentative] Waiting for interface eth1 IPv6 address(es) to leave the 'tentative' state
INFO : [ipv6_wait_tentative] Waiting for interface eth1 IPv6 address(es) to leave the 'tentative' state
```

### **Salida comando ip addr:**

```
[cursoredes@localhost ~]$ ip addr
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 08:00:27:6b:48:b7 brd ff:ff:ff:ff:ff:ff
    inet6 fd00:0:0:a::3/64 scope global
    valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe6b:48b7/64 scope link
    valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default
qlen 1000
    link/ether 08:00:27:e3:29:bb brd ff:ff:ff:ff:ff:ff
    inet6 fd00:0:0:b::3/64 scope global
    valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fee3:29bb/64 scope link
    valid_lft forever preferred_lft forever

```

## Autoconfiguración. Anuncio de prefijos

El protocolo de descubrimiento de vecinos se usa también para la autoconfiguración de los interfaces de red. Cuando se activa un interfaz, se envía un mensaje de descubrimiento de encaminadores. Los encaminadores presentes responden con un anuncio que contiene, entre otros, el prefijo de la red.

**Ejercicio 14 [VM1, VM2, VM4].** Eliminar las direcciones ULA de los interfaces desactivándolos con `ip link`.

**Ejercicio 15 [Router].** Configurar el servicio zebra para que el encaminador anuncie prefijos. Para ello, crear el archivo `/etc/quagga/zebra.conf` e incluir la información de los prefijos para las dos redes. Cada entrada será de la forma:

```

interface eth0
no ipv6 nd suppress-ra
ipv6 nd prefix fd00:0:0:a::/64

```

Finalmente, arrancar el servicio con el comando `service zebra start`.

**Ejercicio 16 [VM4].** Comprobar la autoconfiguración del interfaz de red en VM4, volviendo a activar el interfaz y consultando la dirección asignada.

*Copiar la dirección asignada.*

**Dirección VM4:** `fd00::b:a00:27ff:fe51:e06c/64`

**Ejercicio 17 [VM1, VM2].** Estudiar los mensajes del protocolo de descubrimiento de vecinos:

- Activar el interfaz en VM2, comprobar que está configurado correctamente e iniciar una captura de paquetes con Wireshark.
- Activar el interfaz en VM1 y estudiar los mensajes ICMP de tipo Router Solicitation y Router Advertisement.
- Comprobar las direcciones destino y origen de los datagramas, así como las direcciones destino y origen de la trama Ethernet. Especialmente la relación entre las direcciones IP y MAC. Estudiar la salida del comando `ip maddr`.

*Copiar una captura de pantalla de Wireshark.*



**Dirección VM2:** *fd00::a:a00:27ff:fe02:6fe3/64*

**Dirección VM1:** *fd00::a:a00:27ff:fe1a:e9cf/64*

VM2 [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Capturing from eth0 [Wireshark 1.10.14 (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	Protocol	Length	Info
1	0.000000000	::	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
2	0.537126701	::	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
3	0.906141315	::	ff02::1:ff1a:e9cf	ICMPv6	78	Neighbor Solicitation for fe80::a00:27ff:fe1a:e9cf
4	1.907817791	fe80::a00:27ff:fe1a:e9cf	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
5	1.907829625	fe80::a00:27ff:fe1a:e9cf	ff02::2	ICMPv6	70	Router Solicitation from 08:00:27:1a:e9:cf
6	1.908096752	fe80::a00:27ff:fe6b:48b7	ff02::1	ICMPv6	110	Router Advertisement from 08:00:27:6b:48:b7
7	2.340930063	fe80::a00:27ff:fe1a:e9cf	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
8	2.761340261	::	ff02::1:ff1a:e9cf	ICMPv6	78	Neighbor Solicitation for fd00::a:a00:27ff:fe1a:e9cf
9	365.706712318	fe80::a00:27ff:fe6b:48b7	ff02::1	ICMPv6	110	Router Advertisement from 08:00:27:6b:48:b7

Frame 4: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface 0

Ethernet II, Src: CadmusCo\_1a:e9:cf (08:00:27:1a:e9:cf), Dst: IPv6mcast\_00:00:00:16 (33:33:00:00:00:16)

Destination: IPv6mcast\_00:00:00:16 (33:33:00:00:00:16)

Source: CadmusCo\_1a:e9:cf (08:00:27:1a:e9:cf)

Type: IPv6 (0x86dd)

Internet Protocol Version 6, Src: fe80::a00:27ff:fe1a:e9cf (fe80::a00:27ff:fe1a:e9cf), Dst: ff02::16 (ff02::16)

0110 .... = Version: 6

.... 0000 0000 .... = Traffic class: 0x00000000

.... 0000 0000 0000 0000 = Flowlabel: 0x00000000

Payload length: 36

Next header: IPv6 hop-by-hop option (0)

Hop limit: 1

Source: fe80::a00:27ff:fe1a:e9cf (fe80::a00:27ff:fe1a:e9cf)

[Source SA MAC: CadmusCo\_1a:e9:cf (08:00:27:1a:e9:cf)]

Destination: ff02::16 (ff02::16)

Hop-by-Hop Option

Next header: ICMPv6 (58)

Length: 0 (8 bytes)

IPv6 Option (Router Alert)

IPv6 Option (PadN)

Internet Control Message Protocol v6

### **Salida comando ip maddr:**

1: lo

inet 224.0.0.1

inet6 ff02::1

inet6 ff01::1

2: eth0

link 01:00:5e:00:00:01

link 33:33:00:00:00:01

link 33:33:ff:1a:e9:cf

inet 224.0.0.1

inet6 ff02::1:ff1a:e9cf users 2

inet6 ff02::1

inet6 ff01::1

**Para saber más...** En el proceso de autoconfiguración se genera también el identificador de interfaz según el *Extended Unique Identifier* (EUI-64) modificado. La configuración del protocolo de anuncio de encaminadores tiene múltiples opciones que se pueden consultar en la documentación de zebra (ej. intervalo entre anuncios no solicitados). Cuando sólo se necesita un servicio que implemente el anuncio de prefijos, y no algoritmos de enrutamiento para el router, se puede usar el proyecto de código libre *Router Advertisement Daemon*, radvd.

**Ejercicio 18 [VM1].** La generación del identificador de interfaz mediante EUI-64 supone un problema de privacidad para las máquinas clientes, que pueden ser rastreadas por su dirección MAC. En estos casos, es conveniente activar las extensiones de privacidad para generar un identificador de interfaz pseudoaleatorio temporal para las direcciones globales. Activar las extensiones de privacidad en VM1

con `sysctl -w net.ipv6.conf.eth0.use_tempaddr=2`.

*Copiar la dirección asignada.*

### **Dirección VM1**

`fd00::a:48b8:2f1a:ff91:7958/64`

## ICMPv6

El protocolo ICMPv6 permite el intercambio de mensajes para el control de la red, tanto para la detección de errores como para la consulta de la configuración de ésta. Durante el desarrollo de la práctica hemos visto los más importantes.

**Ejercicio 19.** Generar mensajes de los siguientes tipos en la red y estudiarlos con ayuda de Wireshark:

- Solicitud y respuesta de eco.
- Solicitud y anuncio de encaminador.
- Solicitud y anuncio de vecino.
- Destino inalcanzable - Sin ruta al destino (Code: 0).
- Destino inalcanzable - Dirección inalcanzable (Code: 3)
- Destino inalcanzable - Puerto inalcanzable (Code: 4)

*Copiar capturas de pantalla de Wireshark con los tres últimos mensajes.*

### **Destino inalcanzable - Sin ruta al destino (Code: 0)**

VM1 [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Capturing from eth0 [Wireshark 1.10.14 (Git Rev Unknown from u...]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protoc	Lengtl	Info
1	0.000000000	fd00::a:48b8:2f1a:ff91:7958	fd00:0:0:c::4	ICMPv6	118	Echo (ping) request id=0x2d94, seq=1, hop limit=64
2	0.000475919	fd00:0:0:a::3	fd00::a:48b8:2f1a:ff91:7958	ICMPv6	166	Destination Unreachable (no route to destination)
3	1.000752512	fd00::a:48b8:2f1a:ff91:7958	fd00:0:0:c::4	ICMPv6	118	Echo (ping) request id=0x2d94, seq=2, hop limit=64
4	1.001202456	fd00:0:0:a::3	fd00::a:48b8:2f1a:ff91:7958	ICMPv6	166	Destination Unreachable (no route to destination)
5	2.002123208	fd00::a:48b8:2f1a:ff91:7958	fd00:0:0:c::4	ICMPv6	118	Echo (ping) request id=0x2d94, seq=3, hop limit=64
6	2.002612015	fd00:0:0:a::3	fd00::a:48b8:2f1a:ff91:7958	ICMPv6	166	Destination Unreachable (no route to destination)
7	3.002063304	fd00::a:48b8:2f1a:ff91:7958	fd00:0:0:c::4	ICMPv6	118	Echo (ping) request id=0x2d94, seq=4, hop limit=64
8	3.002492527	fd00:0:0:a::3	fd00::a:48b8:2f1a:ff91:7958	ICMPv6	166	Destination Unreachable (no route to destination)

▶ Frame 2: 166 bytes on wire (1328 bits), 166 bytes captured (1328 bits) on interface 0

▶ Ethernet II, Src: CadmusCo\_6b:48:b7 (08:00:27:6b:48:b7), Dst: CadmusCo\_1a:e9:cf (08:00:27:1a:e9:cf)

▶ Internet Protocol Version 6, Src: fd00:0:0:a::3 (fd00:0:0:a::3), Dst: fd00::a:48b8:2f1a:ff91:7958 (fd00::a:48b8:2f1a:ff91:7958)

▼ Internet Control Message Protocol v6

Type: Destination Unreachable (1)

Code: 0 (no route to destination)

Checksum: 0x7979 [correct]

Reserved: 00000000

▶ Internet Protocol Version 6, Src: fd00::a:48b8:2f1a:ff91:7958 (fd00::a:48b8:2f1a:ff91:7958), Dst: fd00:0:0:c::4 (fd00:0:0:c::4)

▼ Internet Control Message Protocol v6

Type: Echo (ping) request (128)

Code: 0

Checksum: 0x71f6 [correct]

Identifier: 0x2d94

Sequence: 1

▶ Data (56 bytes)