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**.

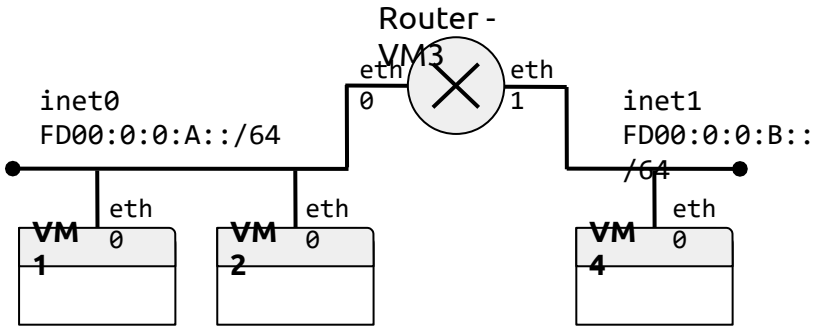
La **contraseña** del usuario `cursoredes` es `cursoredes`.

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`.

Las IPv6 son las predefinidas por cada máquina en modo local.
[cursoredes@localhost ~]$ sudo ip link set dev eth0 up

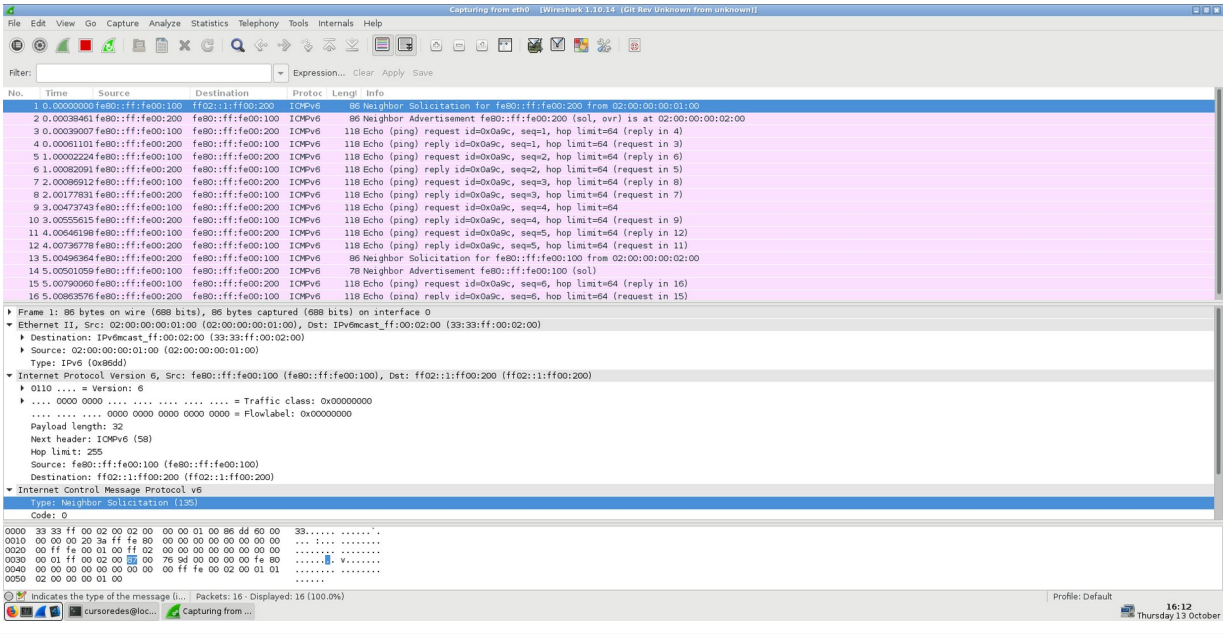
VM1
fe80::ff:fe00:100/64

VM2:
fe80::ff:fe00:200/64

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 utilizado y su salida. Copiar una captura de pantalla de Wireshark donde se vean los campos de la cabecera IPv6.

COMANDO desde VM1 a VM2:
ping -6 fe80::ff:fe00:200 %eth0



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.
VM4:
fe80::ff:fe00:400/64

VM3(Router):
eth0: inet6 fe80::ff:fe00:300/64 scope link
eth1: inet6 fe80::ff:fe00:301/64 scope link

Comando ping Router VM1
ping -6 fe80::ff:fe00:100%eth0

Comando ping Router VM4

ping -6 fe80::ff:fe00:400%eth1

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`.

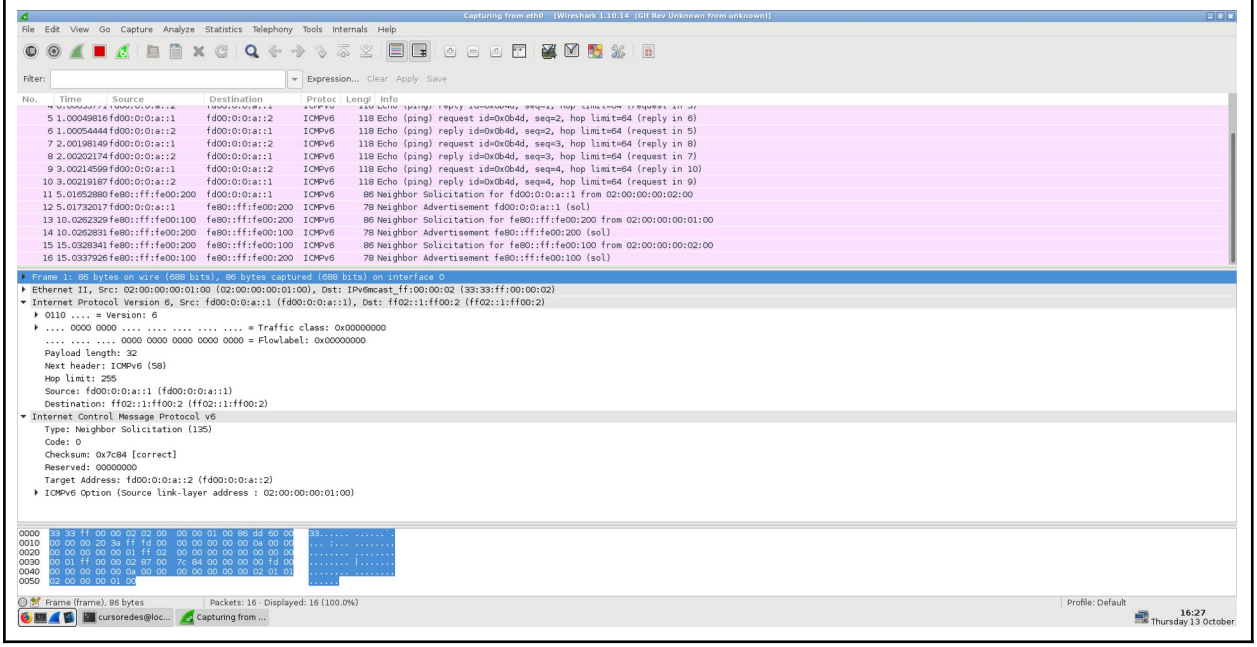
Ejercicio 4 [VM1, VM2]. Configurar VM1 y VM2 para que tengan una dirección ULA en la red `fd00:0:0:a::/64` con el comando `ip`. La parte de identificador de interfaz puede elegirse libremente, siempre que no coincida para ambas máquinas. Incluir la longitud del prefijo al fijar las direcciones.

VM2:
sudo ip address add fd00:0:0:a::2/64 dev eth0

VM1:
sudo ip address add fd00:0:0:a::1/64 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.

Copiar los comandos utilizados.
Ping de VM1 a VM2:
sudo ping6 fd00:0:0:a::2



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.

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.
VM3 a VM1:
ping6 fd00:0:0:a::1

VM3 a VM4:
ping6 fd00:0:0:b::4

VM4 a VM1:
[cursoredes@localhost ~]$ ping6 fd00:0:0:a::1
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.

Copiar los comandos utilizados y su salida.

[cursoredes@localhost ~]$ sudo ip -6 route
unreachable ::/96 dev lo metric 1024 error -113 pref medium
unreachable ::ffff:0.0.0.0/96 dev lo metric 1024 error -113 pref medium
unreachable 2002:a00::/24 dev lo metric 1024 error -113 pref medium
unreachable 2002:7f00::/24 dev lo metric 1024 error -113 pref medium
unreachable 2002:a9fe::/32 dev lo metric 1024 error -113 pref medium
unreachable 2002:ac10::/28 dev lo metric 1024 error -113 pref medium
unreachable 2002:c0a8::/32 dev lo metric 1024 error -113 pref medium
unreachable 2002:e000::/19 dev lo metric 1024 error -113 pref medium
unreachable 3ffe:ffff::/32 dev lo metric 1024 error -113 pref medium
fd00:0:0:a::/64 dev eth0 proto kernel metric 256 pref medium
fd00:0:0:b::/64 dev eth1 proto kernel metric 256 pref medium
fe80::/64 dev eth0 proto kernel metric 256 pref medium
fe80::/64 dev eth1 proto kernel metric 256 pref medium

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 net.ipv6.conf.all.forwarding=1`.

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

VM1: *sudo ip route add default via fd00:0:0:a::3*
VM2: *sudo ip route add default via fd00:0:0:a::3*
VM3: *sudo ip route add default via fd00:0:0:b::3*
Salida: *ping -6 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=63 time=0.663 ms* *64 bytes from fd00:0:0:a::1: icmp_seq=2 ttl=63 time=1.72 ms* *64 bytes from fd00:0:0:a::1: icmp_seq=3 ttl=63 time=1.85 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)**

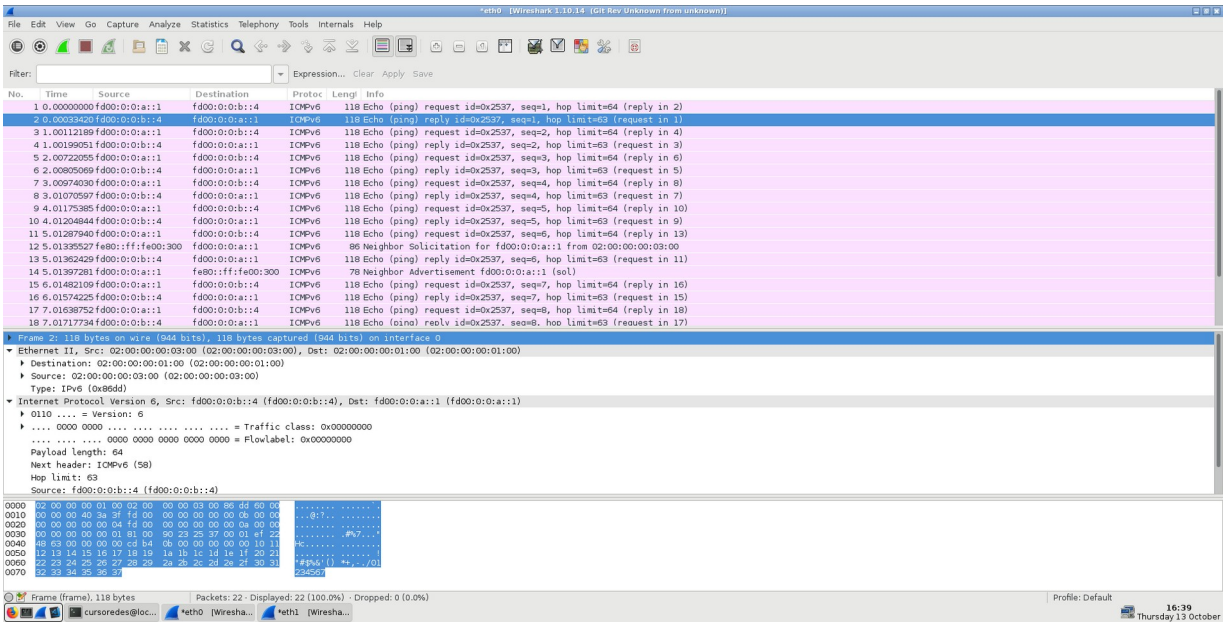
MAC Origen	MAC Destino	IPv6 Origen	IPv6 Destino	ICMPv6 Tipo
02:00:00:00:01:00	02:00:00:00:03:00	Fe80::ff:fe00:300 VM1 Link Local	Fd00:0:0:a::1 ULA VM1	Solicitud
02:00:00:00:03:00	02:00:00:00:01:00	Fd00:0:0:a::1 ULA VM1	fe80::ff:fe00:300	Anuncio
02:00:00:00:01:00	02:00:00:00:03:00	Fd00:0:0:a::1 ULA VM1	Fd00:0:0:b::4 ULA VM4	Request
02:00:00:00:03:00	02:00:00:00:01:00	Fd00:0:0:b::4 ULA VM4	Fd00:0:0:a::1 ULA VM1	Reply

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

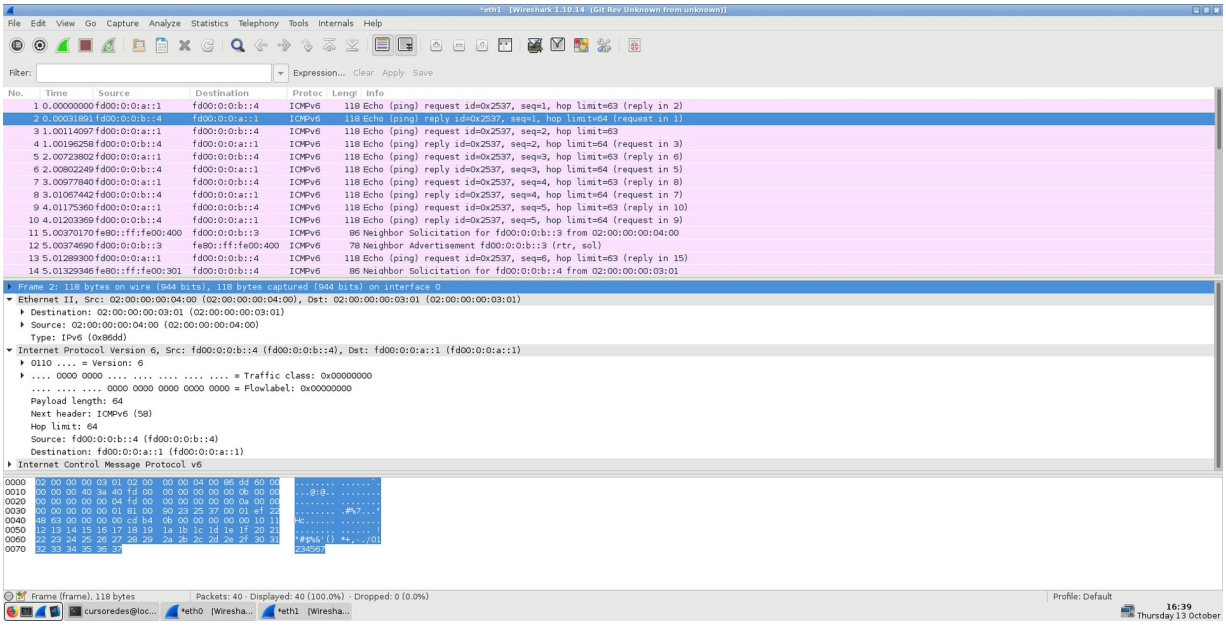
MAC Origen	MAC Destino	IPv6 Origen	IPv6 Destino	ICMPv6 Tipo
02:00:00:00:04:00	02:00:00:00:03:01	Fe80::ff:fe00:301 VM3-Link local	Fd00:0:0:b::4 ULA VM4	Solicitud
02:00:00:00:03:01	02:00:00:00:04:00	Fd00:0:0:b::4 ULA VM4	fe80::ff:fe00:301	Anuncio
02:00:00:00:03:01	02:00:00:00:04:00	Fd00:0:0:a::1 ULA VM1	Fd00:0:0:b::4 ULA VM4	Request

02:00:00:00:04:00	02:00:00:00:03:01	Fd00:0:0:b::4 ULA VM4	Fd00:0:0:a::1 ULA VM1	Reply

ETH0:



ETH1:



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 (si tiene)>
DEVICE=<nombre del interfaz>
```

Fichero eth0:

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

Fichero eth1:

```
TYPE=Ethernet
BOOTPROTO=none
IPV6ADDR=fd00:0:0:b::1/64
DEVICE=eth1
```

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

Comando → sudo ifup ifcfg-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

Comando → sudo ifup ifcfg-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

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`.

Copiar la dirección asignada.
COMANDO USADO EN TODAS LAS MÁQUINAS:
sudo ip link set eth0 down

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.
fe80::ff:fe00:400/64

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`.

*Salida del comando sudo 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:00:02:00
inet 224.0.0.1
inet6 ff02::1:ff00:200 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 encaminamiento 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 net.ipv6.conf.eth0.use_tempaddr=2` y repetir el proceso de autoconfiguración.

Realizamos el comando → sudo sysctl -w net.ipv6.conf.eth0.use_tempaddr=2

Y realizamos un down y un up.

inet6 fd00::a:a0d5:9590:a281:ef64/64 scope global temporary dynamic

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)