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**.

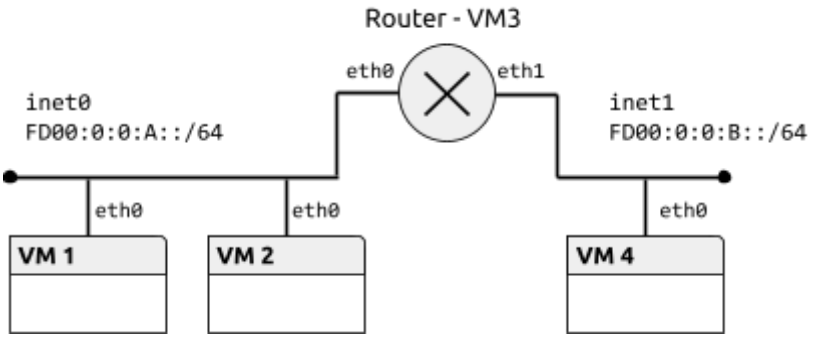
La **contraseña** del usuario `cursoredes` es `cursoredes`.

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`.

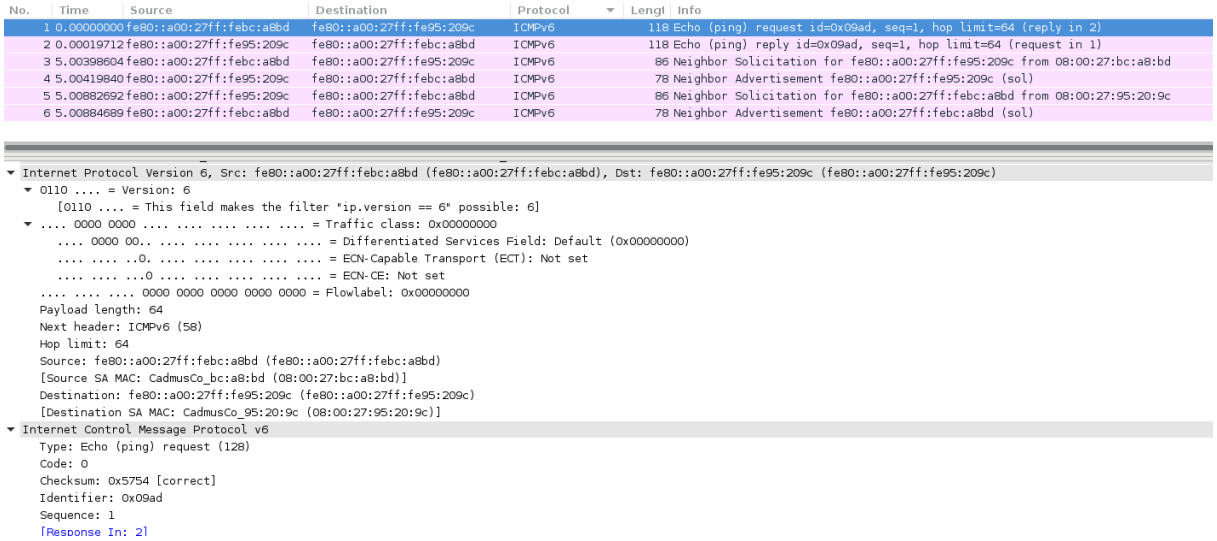
ip link set dev eth0 up

VM1: fe80::a00:27ff:febc:a8bd/64
VM2: fe80::a00:27ff:fe95:209c/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 utilizados y su salida. Copiar una captura de pantalla de Wireshark donde se vean los campos de la cabecera IPv6.

ping6 fe80::a00:27ff:fe95:209c -I eth0 -c 1



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.

ip link set dev eth0 up
ip link set dev eth1 up

ip address

Router:
eth0: fe80::a00:27ff:fe63:dc54/64 **eth1:** fe80::a00:27ff:fe2a:bcbb/64
VM3: fe80::a00:27ff:fed2:ea4d/64

ping6 fe80::a00:27ff:febc:a8bd -I eth0 -c 1
ping6 fe80::a00:27ff:fed2:ea4d -I eth1 -c 1

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.

VM1> ip address add fd00:0:0:a::1/64 dev eth0

VM2> 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.

Router> ip address add fd00:0:0:a::3/64 dev eth0
Router> ip address add fd00:0:0:b::3/64 dev eth1

VM4> 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.

Router > VM1
ping6 fd00:0:0:a::1 -I eth0 -c 1

Router > VM4
ping6 fd00:0:0:b::4 -I eth1 -c 1

VM1 > VM4
ping6 fd00:0:0:b::4 -I eth0 -c 1

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`.

VM1> ip -6 route add fd00:0:0:b::/64 via fd00:0:0:a::3
VM2> ip -6 route add fd00:0:0:b::/64 via fd00:0:0:a::3

VM4>ip -6 route add fd00:0:0:a::/64 via fd00:0:0:b::3

VM1 > VM4
ping6 fd00:0:0:b::4 -I eth0 -c 1
1 packets transmitted, 1 received, 0% packet loss, time 0ms

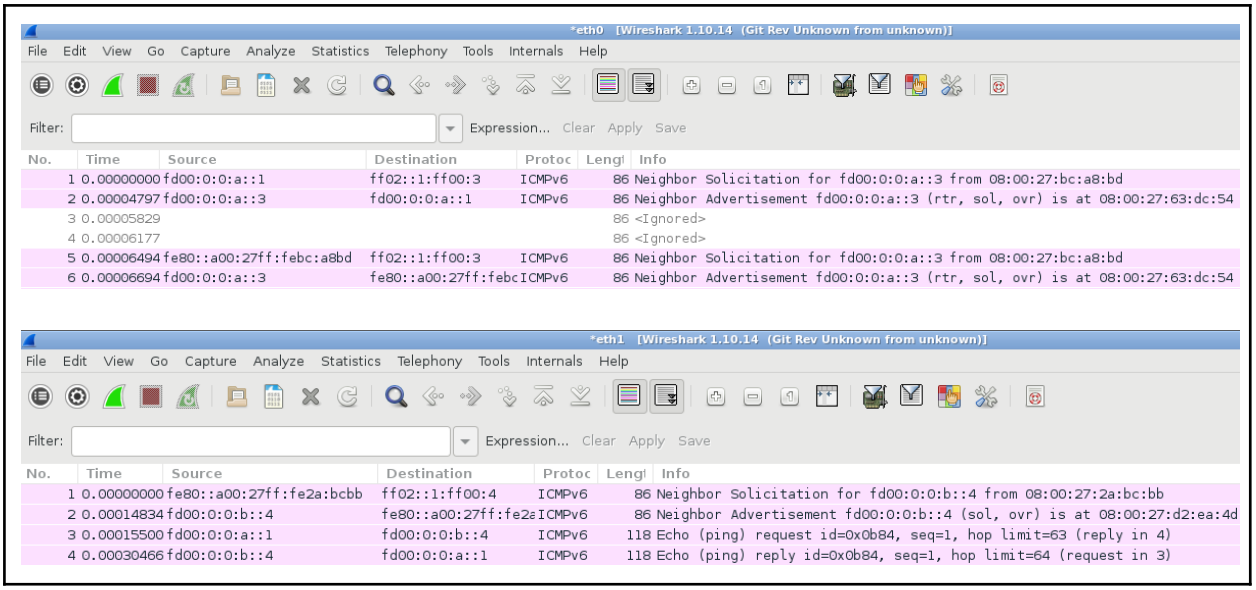
Ejercicio 11 [VM1, Router, VM4]. Abrir Wireshark en Router e iniciar dos capturas, una en cada interfaz de red. Borrar la tabla de vecinos en VM1 y Router (con `ip neigh flush dev <interfaz>`). Usar la orden `ping6` entre VM1 y VM4. Completar la siguiente tabla con todos los mensajes hasta el primer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

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

MAC Origen	MAC Destino	IPv6 Origen	IPv6 Destino	ICMPv6 Tipo
08:00:27:bc:a8:bd	Multicast	fd00:0:0:a::1	ff02::1:ff00:3 (Multicast de nodo)	Neighbor solicitation
08:00:27:63:dc:54	08:00:27:bc:a8:bd	fd00:0:0:a::3	fd00:0:0:a::1	Neighbor advertisement
08:00:27:bc:a8:bd	08:00:27:63:dc:54	fd00:0:0:a::1	fd00:0:0:b::4	Echo request
08:00:27:63:dc:54	08:00:27:bc:a8:bd	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
08:00:27:2a:bc:bb	Multicast	fe80::a00:27ff:fe2a :bcbb (VM3 Scope Link)	ff02::1:ff00:4 (Multicast de nodo)	Neighbor solicitation
08:00:27:d2:ea:4d	08:00:27:2a:bc:bb	fd00:0:0:b::4	fe80::a00:27ff:fe2a :bcbb	Neighbor advertisement
08:00:27:2a:bc:bb	08:00:27:d2:ea:4d	fd00:0:0:a::1	fd00:0:0:b::4	Echo request
08:00:27:d2:ea:4d	08:00:27:2a:bc:bb	fd00:0:0:b::4	fd00:0:0:a::1	Echo reply



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>
```

```
TYPE=Ethernet
BOOTPROTO=none
IPV6ADDR=fd00:0:0:a::/64
IPV6FORWARDING=yes
DEVICE=eth0

TYPE=Ethernet
BOOTPROTO=none
IPV6ADDR=fd00:0:0:b::/64
IPV6FORWARDING=yes
DEVICE=eth1
```

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

```
ip link set dev eth0 down
ip link set dev eth1 down
ifup eth0
ifup eth1
```

```
ifdown eth0
ifdown eth1
ip link set dev eth0 up
ip link set dev eth1 up
ifup eth0
ifup  eth1
```

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

interface eth1
  no ipv6 nd suppress-ra
  ipv6 nd prefix fd00:0:0:b::/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.

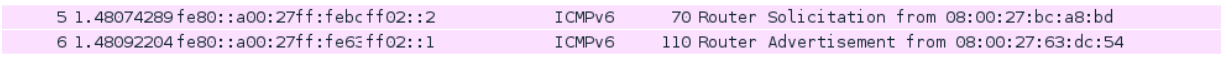
```
Copiar la dirección asignada.

ip link set dev eth0 up
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:d2:ea:4d brd ff:ff:ff:ff:ff:ff
   inet6 fd00::b:a00:27ff:fed2:ea4d/64 scope global mngtmpaddr dynamic
      valid_lft 2591999sec preferred_lft 604799sec
   inet6 fe80::a00:27ff:fed2:ea4d/64 scope link
      valid_lft forever preferred_lft forever
```

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.



EN VM1:

ip maddr
1: lo
inet 224.0.0.1
inet6 ff02::1
inet6 ff01::1
2: eth0
link 33:33:00:00:00:01
link 01:00:5e:00:00:01
link 33:33:ff:bc:a8:bd
inet 224.0.0.1
inet6 ff02::1:ffbc:a8bd 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 -w net.ipv6.conf.eth0.use_tempaddr=2`.

Copiar la dirección asignada.
ip link set dev eth0 down
sysctl -w net.ipv6.conf.eth0.use_tempaddr=2
ip link set dev eth0 up
ip address

inet6 **fd00::a:4d63:f59d:8dc8:6beb/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)

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