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.

Contenidos

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

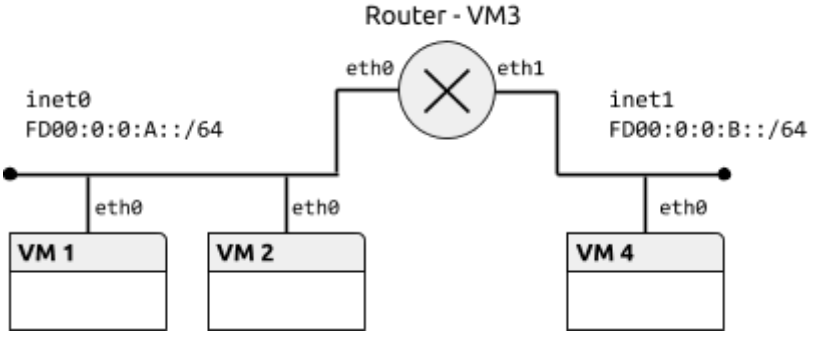
Activar el **portapapeles bidireccional** (menú Dispositivos) en las máquinas virtuales.

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

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

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`.

[VM1]
ip link set dev eth0 up
ip addr
Vemos que se ha puesto dirección de enlace local: fe80::a00:27ff:febd:4e3e/64

[VM2]
ip link set dev eth0 up
ip addr
fe80::a00:27ff:fe13:477c/64

(Vemos que se ha usado EUI-64 para obtener esta dirección si nos fijamos en las MAC de las interfaces)

Ejercicio 2 [VM1, VM2]. Comprobar la conectividad entre VM1 y VM2 con la orden `ping6`. 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.

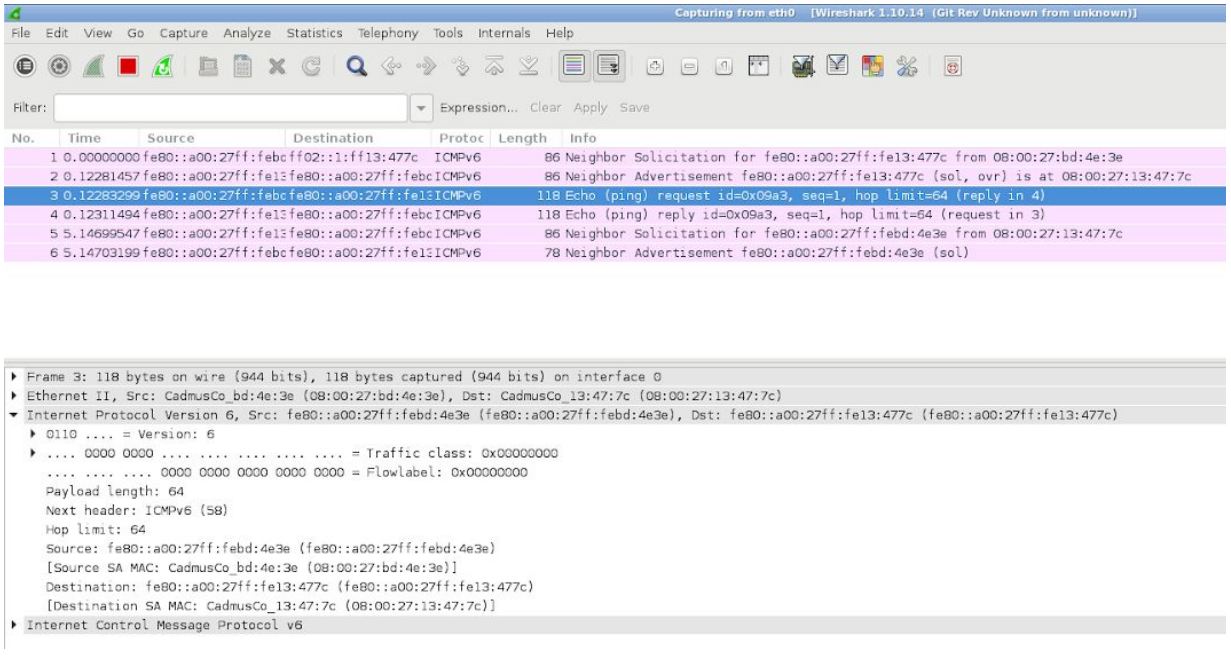
Copiar el comando utilizados y su salida. Copiar una captura de pantalla de Wireshark donde se vean los campos de la cabecera IPv6.

Comando: ping6 -c 1 fe80::a00:27ff:fe13:477c%eth0

Salida:

PING fe80::a00:27ff:fe13:477c%eth0(fe80::a00:27ff:fe13:477c%eth0) 56 data bytes
64 bytes from fe80::a00:27ff:fe13:477c%eth0: icmp_seq=1 ttl=64 time=123 ms

--- fe80::a00:27ff:fe13:477c%eth0 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 123.131/123.131/123.131/0.000 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.

(Hacemos ping hacia la MV1)
[Router]
***Comando**: ping6 -c 1 fe80::a00:27ff:febd:4e3e%eth0*

***Salida**:*
PING fe80::a00:27ff:febd:4e3e%eth0(fe80::a00:27ff:febd:4e3e%eth0) 56 data bytes
64 bytes from fe80::a00:27ff:febd:4e3e%eth0: icmp_seq=1 ttl=64 time=0.819 ms

--- fe80::a00:27ff:febd:4e3e%eth0 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.819/0.819/0.819/0.000 ms

(Hacemos ping hacia la MV4)

[VM4]
***Comando**: ip addr*

***Salida**:*
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:c6:26 brd ff:ff:ff:ff:ff:ff
inet6 fe80::a00:27ff:fe6b:c626/64 scope link
valid_lft forever preferred_lft forever

[Router]
***Comando**: ping6 -c 1 fe80::a00:27ff:fe6b:c626%eth1*

***Salida**:*
PING fe80::a00:27ff:fe6b:c626%eth1(fe80::a00:27ff:fe6b:c626%eth1) 56 data bytes
64 bytes from fe80::a00:27ff:fe6b:c626%eth1: icmp_seq=1 ttl=64 time=0.770 ms

--- fe80::a00:27ff:fe6b:c626%eth1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.770/0.770/0.770/0.000 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]
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.

Copiar los comandos utilizados.

[Router]
ip address add fd00:0:0:a::3/64 dev eth0
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.

Copiar los comandos utilizados.

[Router]
(Conectividad con MV1)
ping6 -c 1 fd00:0:0:a::1
(La salida muestra que todo ok)
(Conectividad con MV4)
ping6 -c 1 fd00:0:0:b::4
(La salida muestra que todo ok)

[VM1]
ping6 -c 1 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.

[VM1]
Comando:
ip -6 route
Salida:
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
fe80::/64 dev eth0 proto kernel metric 256 pref medium

[Router]
Comando:
ip -6 route
Salida:
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 -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]
ip route add default via fd00:0:0:a::3
[VM2]
ip route add default via fd00:0:0:a::3
[VM4]
ip route add default via fd00:0:0:b::3

[VM1]
Comando:
ping6 -c 1 fd00:0:0:b::4

Salida:
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=0.936 ms

--- fd00:0:0:b::4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.936/0.936/0.936/0.000 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:

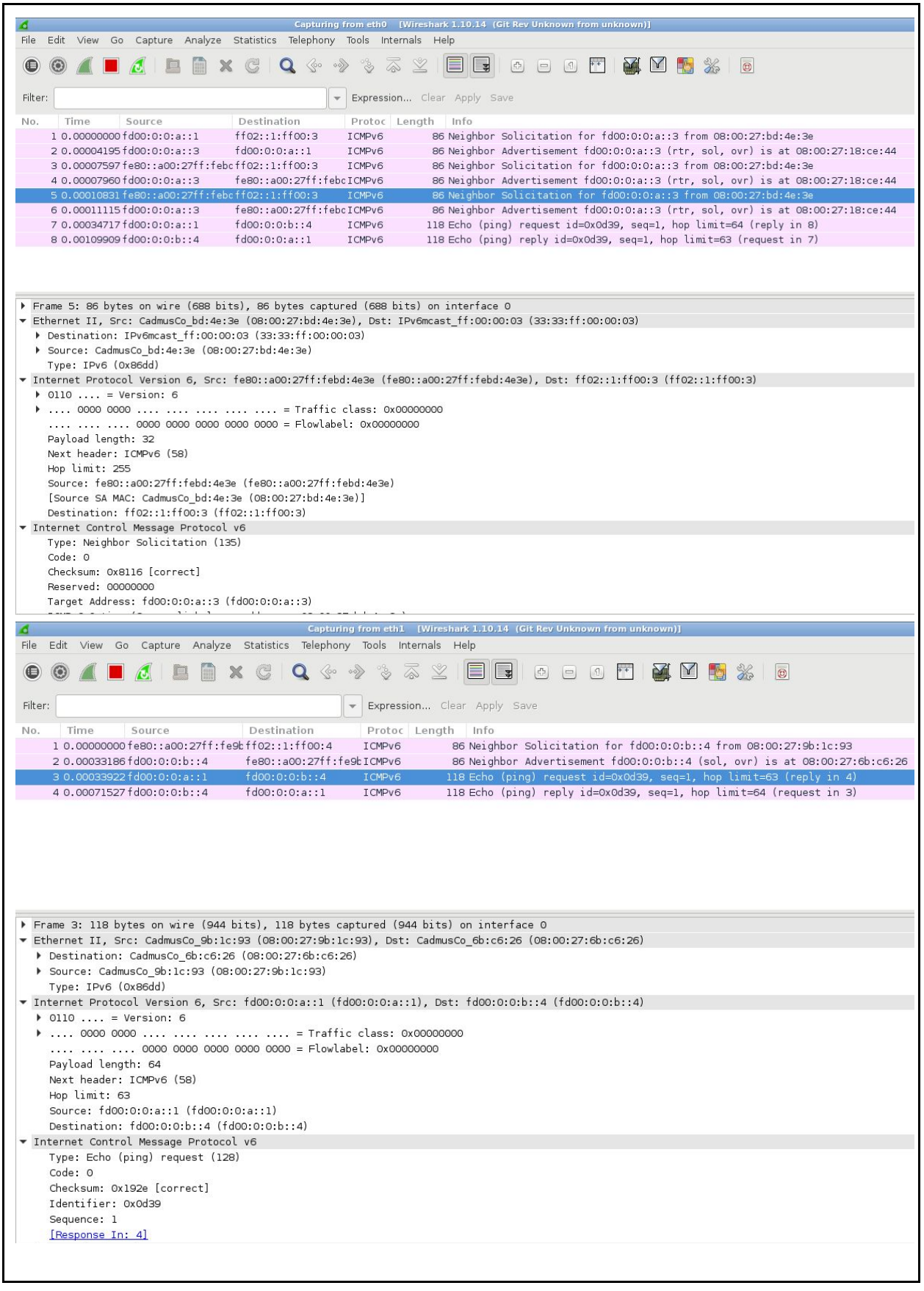
Red `fd00:0:0:a::/64` - Router (eth0)

MAC Origen	MAC Destino	IPv6 Origen	IPv6 Destino	ICMPv6 Tipo
08:00:27:bd:4e:3e	33:33:ff:00:00:04 (Broadcast)	fe80::a00:27ff:febd :4e3e (VM1 Link- Local)	ff02::1:ff00:3 (Dir de nodo solicitado Router)	Solicitud de Vecino
08:00:27:18:ce:44	08:00:27:bd:4e:3e	fd00:0:0:a::3 (Router ULA)	fe80::a00:27ff:febd :4e3e (VM1 Link- Local)	Anuncio de vecino
08:00:27:bd:4e:3e	08:00:27:18:ce:44	fd00:0:0:a::1 (VM1 ULA)	fd00:0:0:b::4 (VM4 ULA)	Echo request
08:00:27:18:ce:44	08:00:27:bd:4e:3e	fd00:0:0:b::4 (VM4 ULA)	fd00:0:0:a::1 (VM1 ULA)	Echo Reply

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

MAC Origen	MAC Destino	IPv6 Origen	IPv6 Destino	ICMPv6 Tipo
08:00:27:9b:1c:93	33:33:ff:00:00:04 (Broadcast)	Fe80::a00:27ff:fe9b:1 c93 (VM3 eth1 Link-Local)	ff02::1:ff00:4 (Dir de nodo solicitado VM4)	Solicitud de Vecino
08:00:27:6b:c6:26	08:00:27:9b:1c:93	fd00:0:0:b::4 (VM4 ULA)	Fe80::a00:27ff:fe9b:1c 93 (VM3 eth1 Link-Local)	Anuncio de vecino
08:00:27:9b:1c:93	08:00:27:6b:c6:26	fd00:0:0:a::1 (VM1 ULA)	fd00:0:0:b::4 (VM4 ULA)	Echo request
08:00:27:6b:c6:26	08:00:27:9b:1c:93	fd00:0:0:b::4 (VM4 ULA)	fd00:0:0:a::1 (VM1 ULA)	Echo Reply

Copiar dos capturas de pantalla de Wireshark.


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 Protoc Length Info
1 0.00000000 fd00:0:0:a::1 ff02::1:ff00:3 ICMPv6 86 Neighbor Solicitation for fd00:0:0:a::3 from 08:00:27:bd:4e:3e
2 0.00004195 fd00:0:0:a::3 fd00:0:0:a::1 ICMPv6 86 Neighbor Advertisement fd00:0:0:a::3 (rtr, sol, ovr) is at 08:00:27:18:ce:44
3 0.00007597 fe80::a00:27ff:febc ff02::1:ff00:3 ICMPv6 86 Neighbor Solicitation for fd00:0:0:a::3 from 08:00:27:bd:4e:3e
4 0.00007960 fd00:0:0:a::3 fe80::a00:27ff:febc ICMPv6 86 Neighbor Advertisement fd00:0:0:a::3 (rtr, sol, ovr) is at 08:00:27:18:ce:44
5 0.00010831 fe80::a00:27ff:febc ff02::1:ff00:3 ICMPv6 86 Neighbor Solicitation for fd00:0:0:a::3 from 08:00:27:bd:4e:3e
6 0.00011115 fd00:0:0:a::3 fe80::a00:27ff:febc ICMPv6 86 Neighbor Advertisement fd00:0:0:a::3 (rtr, sol, ovr) is at 08:00:27:18:ce:44
7 0.00034717 fd00:0:0:a::1 fd00:0:0:b::4 ICMPv6 118 Echo (ping) request id=0x0d39, seq=1, hop limit=64 (reply in 8)
8 0.00109909 fd00:0:0:b::4 fd00:0:0:a::1 ICMPv6 118 Echo (ping) reply id=0x0d39, seq=1, hop limit=63 (request in 7)
Frame 5: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
Ethernet II, Src: CadmusCo_bd:4e:3e (08:00:27:bd:4e:3e), 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_bd:4e:3e (08:00:27:bd:4e:3e)
Type: IPv6 (0x86dd)
Internet Protocol Version 6, Src: fe80::a00:27ff:febd:4e3e (fe80::a00:27ff:febd:4e3e), 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: fe80::a00:27ff:febd:4e3e (fe80::a00:27ff:febd:4e3e)
[Source SA MAC: CadmusCo_bd:4e:3e (08:00:27:bd:4e:3e)]
Destination: ff02::1:ff00:3 (ff02::1:ff00:3)
Internet Control Message Protocol v6
Type: Neighbor Solicitation (135)
Code: 0
Checksum: 0x8116 [correct]
Reserved: 00000000
Target Address: fd00:0:0:a::3 (fd00:0:0:a::3)
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 Length Info
1 0.00000000 fe80::a00:27ff:fe9b ff02::1:ff00:4 ICMPv6 86 Neighbor Solicitation for fd00:0:0:b::4 from 08:00:27:9b:1c:93
2 0.00033186 fd00:0:0:b::4 fe80::a00:27ff:fe9b ICMPv6 86 Neighbor Advertisement fd00:0:0:b::4 (sol, ovr) is at 08:00:27:6b:c6:26
3 0.00033922 fd00:0:0:a::1 fd00:0:0:b::4 ICMPv6 118 Echo (ping) request id=0x0d39, seq=1, hop limit=63 (reply in 4)
4 0.00071527 fd00:0:0:b::4 fd00:0:0:a::1 ICMPv6 118 Echo (ping) reply id=0x0d39, seq=1, hop limit=64 (request in 3)
Frame 3: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
Ethernet II, Src: CadmusCo_9b:1c:93 (08:00:27:9b:1c:93), Dst: CadmusCo_6b:c6:26 (08:00:27:6b:c6:26)
Destination: CadmusCo_6b:c6:26 (08:00:27:6b:c6:26)
Source: CadmusCo_9b:1c:93 (08:00:27:9b:1c:93)
Type: IPv6 (0x86dd)
Internet Protocol Version 6, Src: fd00:0:0:a::1 (fd00:0:0:a::1), Dst: fd00:0:0:b::4 (fd00:0:0:b::4)
0110 .... = Version: 6
.... 0000 0000 .... = Traffic class: 0x00000000
.... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
Payload length: 64
Next header: ICMPv6 (58)
Hop limit: 63
Source: fd00:0:0:a::1 (fd00:0:0:a::1)
Destination: fd00:0:0:b::4 (fd00:0:0:b::4)
Internet Control Message Protocol v6
Type: Echo (ping) request (128)
Code: 0
Checksum: 0x192e [correct]
Identifier: 0x0d39
Sequence: 1
[Response In: 4]

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 estática en formato CIDR>
IPV6_DEFAULTGW=<dirección IP estática del encaminador por defecto (si existe)>
DEVICE=<nombre del interfaz>
```

Copiar el contenido de los ficheros.

Fichero /etc/sysconfig/network-scripts/ifcfg-eth0
TYPE=Ethernet
BOOTPROTO=none
IPV6ADDR=fd00:0:0:a::3/64
DEVICE=eth0

Fichero /etc/sysconfig/network-scripts/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 eth0
Ifdown eth1
ifup eth0
ifup eth1
ip addr

Salida:
eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
link/ether 08:00:27:18:ce:44 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:fe18:ce44/64 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 08:00:27:9b:1c:93 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:fe9b:1c93/64 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.

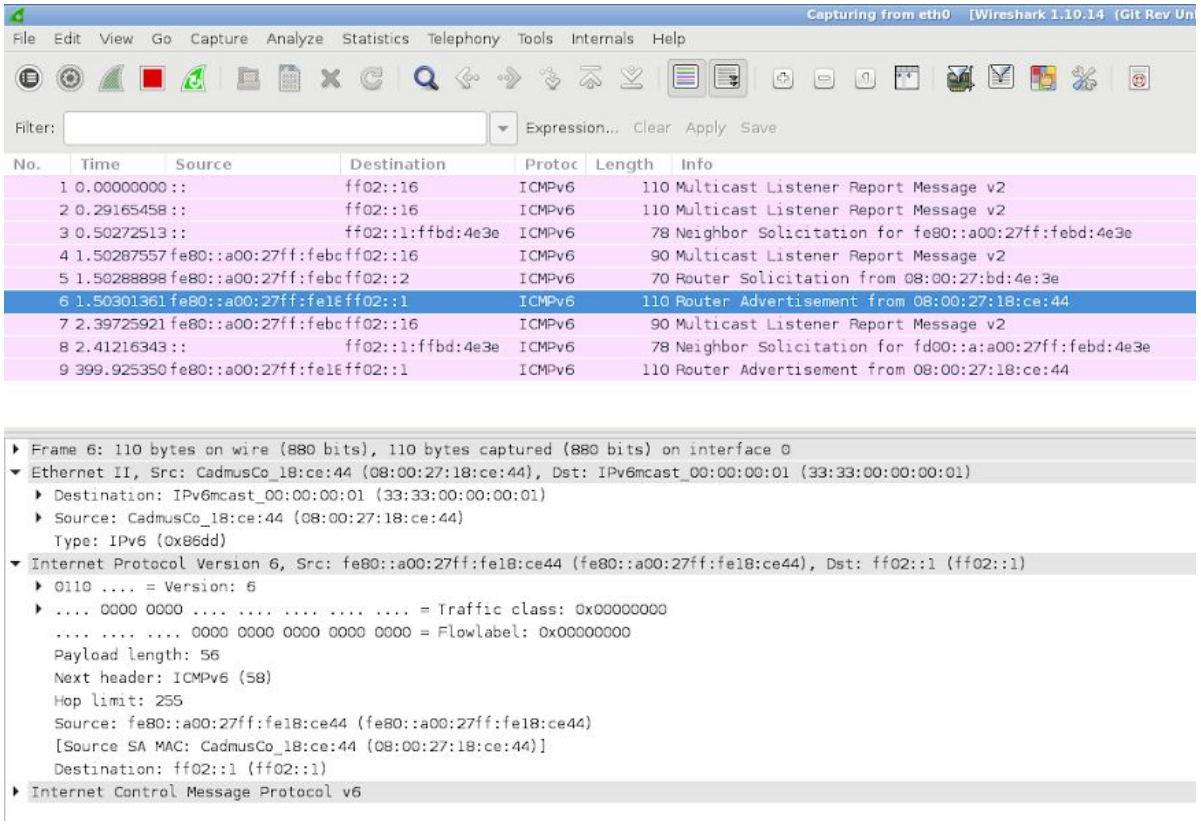
Copiar la dirección asignada.

fd00::b:a00:27ff:fe6b:c626/64

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.



En el Router Solicitation la MAC destino es la de difusión.
La IP origen es la local de VM1 generada con EUI-64
La IP destino es ff02::2 (la del grupo multicast de todos los routers del enlace local)

En el Router Advertisement la MAC destino también es la de difusión.
La IP origen es la local del router generada con EUI-64
La IP destino es ff02::1 (la del grupo multicast de todos los nodos del enlace local)

EL comando ipmaddr nos permite comprobar los grupos multicast en los que esá cada interfaz.
Haciendolo en VM1:

Comando: *ip maddr*
Salida:
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:bd:4e:3e
inet 224.0.0.1
inet6 ff02::1:ffbd:4e3e 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) que se describe en el RFC 4193. 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.

fd00::a:d466:38b5:b3ab:cd57/64

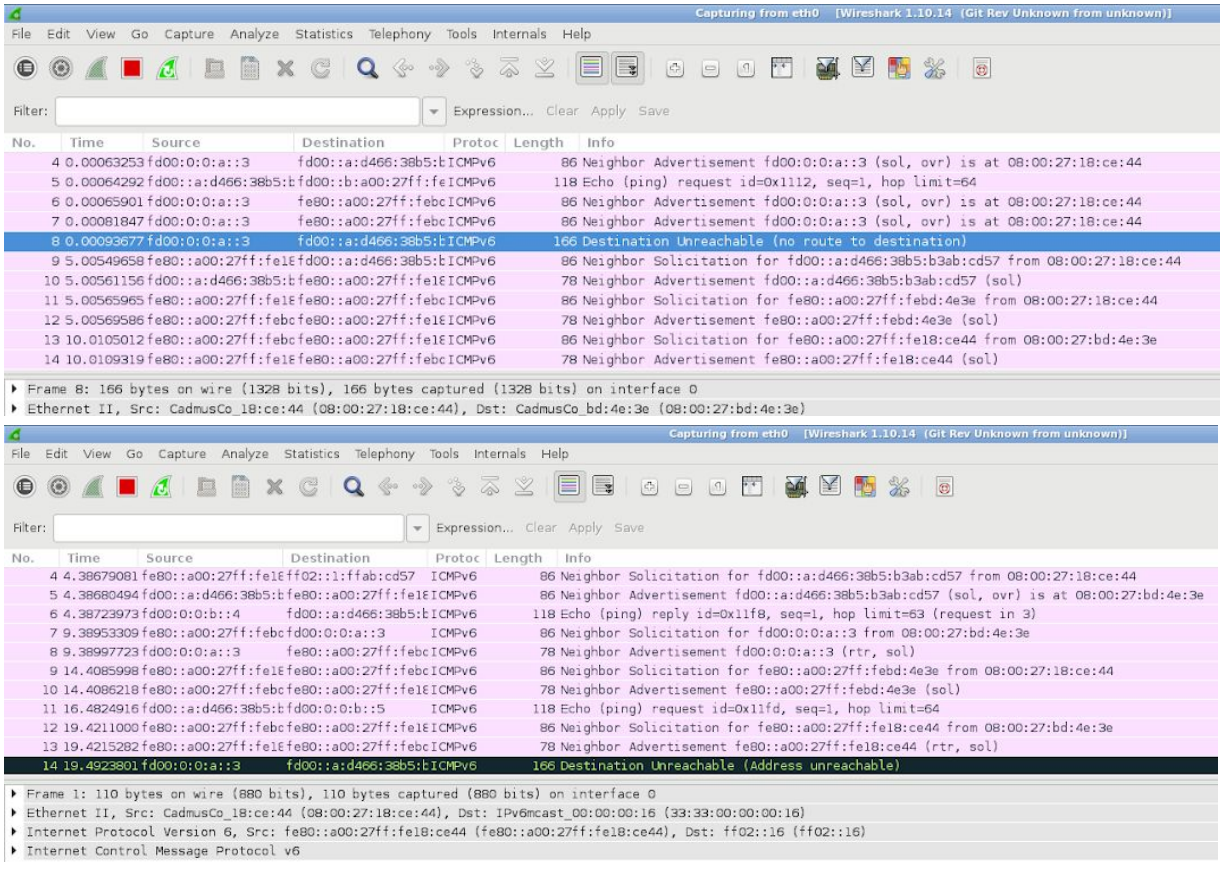
ICMPv6

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 destino inalcanzable (Code: 3)

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



Para que no haya ruta mandamos un mensaje a una ULA de una red para la que el router no tenga ninguna interfaz.

Para destino inalcanzable lo mandamos a la red fd00::b, pero a un interfaz que no exista.