

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**.

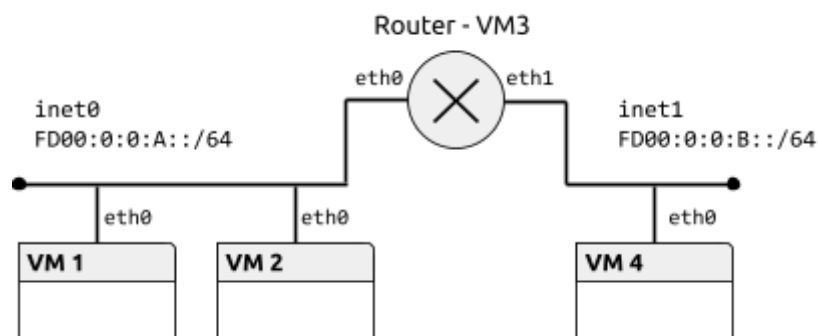
La **contraseña** del usuario cursoredes es cursoredes.

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.

```
[VM1]
sudo ip link set dev eth0 up
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 02:00:00:00:01:00 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::ff:fe00:100/64 scope link
        valid_lft forever preferred_lft forever

[VM2]
sudo ip link set dev eth0 up
[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 02:00:00:00:02:00 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::ff:fe00:200/64 scope link
        valid_lft forever preferred_lft forever
```

Vemos que ambas tienen el prefijo fe80::, que las identifica como direcciones de enlace local.

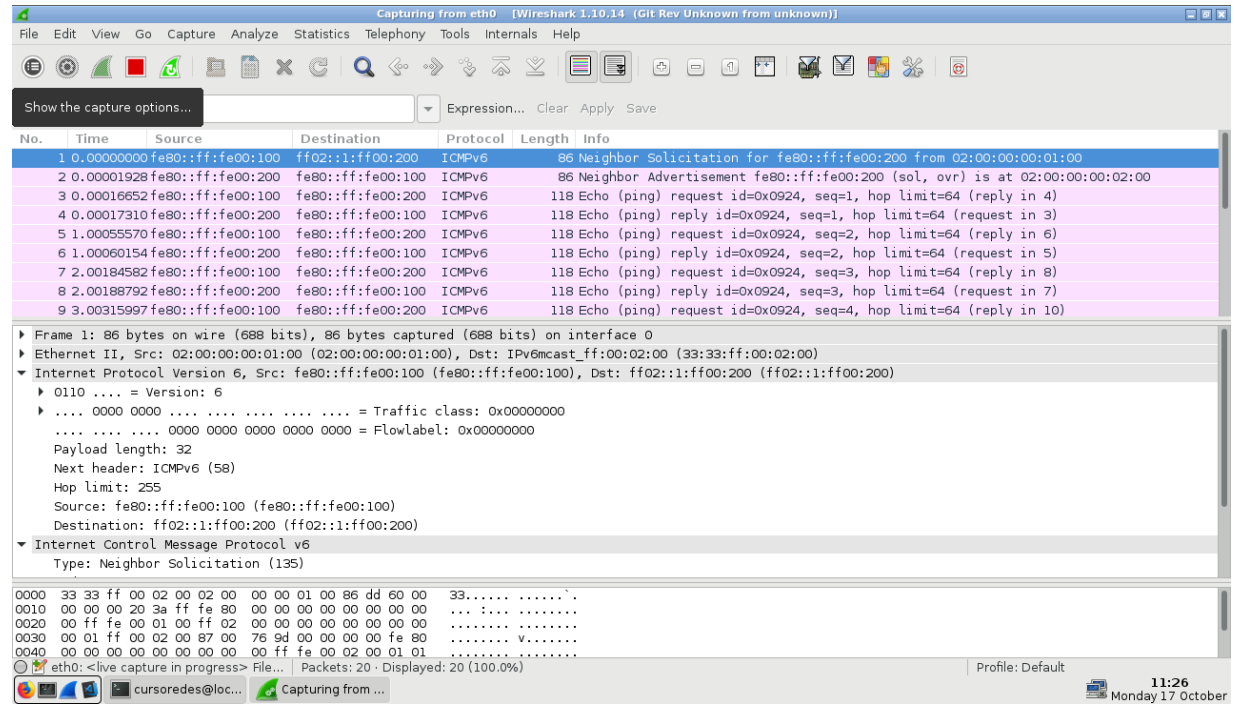
Ejercicio 2 [VM1, VM2]. Comprobar la conectividad entre VM1 y VM2 con la orden ping6 (o ping -6). Cuando se usan direcciones de enlace local, y **sólo en ese caso**, es necesario especificar el interfaz origen, añadiendo %<nombre_interfaz> a la dirección. Consultar las opciones del comando ping6 en la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

```
[VM1]
ping6 fe80::ff:fe00:200%eth0
PING fe80::ff:fe00:200%eth0(fe80::ff:fe00:200%eth0) 56 data bytes
64 bytes from fe80::ff:fe00:200%eth0: icmp_seq=1 ttl=64 time=0.315 ms
64 bytes from fe80::ff:fe00:200%eth0: icmp_seq=2 ttl=64 time=0.936 ms
64 bytes from fe80::ff:fe00:200%eth0: icmp_seq=3 ttl=64 time=0.983 ms
64 bytes from fe80::ff:fe00:200%eth0: icmp_seq=4 ttl=64 time=0.810 ms
64 bytes from fe80::ff:fe00:200%eth0: icmp_seq=5 ttl=64 time=1.04 ms
64 bytes from fe80::ff:fe00:200%eth0: icmp_seq=6 ttl=64 time=0.902 ms
64 bytes from fe80::ff:fe00:200%eth0: icmp_seq=7 ttl=64 time=0.831 ms
64 bytes from fe80::ff:fe00:200%eth0: icmp_seq=8 ttl=64 time=0.872 ms
```

^C

--- fe80::ff:fe00:200%eth0 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7007ms
rtt min/avg/max/mdev = 0.315/0.837/1.048/0.210 ms

[VM2]



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.

[Router]

```
sudo ip link set dev eth0 up  
sudo ip link set dev eth1 up
```

Conectividad con VM1: ping6 fe80::ff:fe00:100%eth0
PING fe80::ff:fe00:100%eth0(fe80::ff:fe00:100%eth0) 56 data bytes
64 bytes from fe80::ff:fe00:100%eth0: icmp_seq=1 ttl=64 time=0.541 ms

^C

--- fe80::ff:fe00:100%eth0 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.541/0.541/0.541/0.000 ms

Conectividad con VM4: ping6 fe80::ff:fe00:400%eth1
PING fe80::ff:fe00:400%eth1(fe80::ff:fe00:400%eth1) 56 data bytes
64 bytes from fe80::ff:fe00:400%eth1: icmp_seq=1 ttl=64 time=0.504 ms
64 bytes from fe80::ff:fe00:400%eth1: icmp_seq=2 ttl=64 time=1.02 ms

^C

--- fe80::ff:fe00:400%eth1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.504/0.766/1.029/0.263 ms

[VM4]

```
sudo ip link set dev eth0 up
```

```
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
```

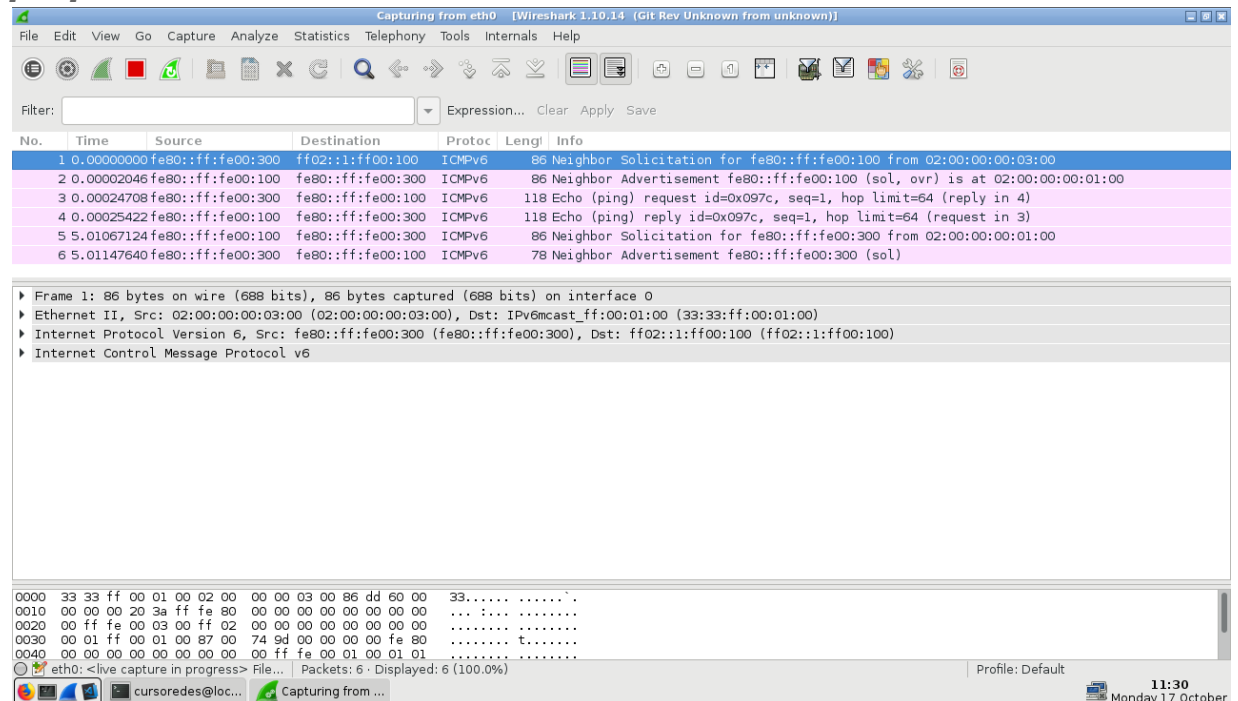
```
    link/ether 02:00:00:00:04:00 brd ff:ff:ff:ff:ff:ff
```

```
    inet6 fe80::ff:fe00:400/64 scope link
```

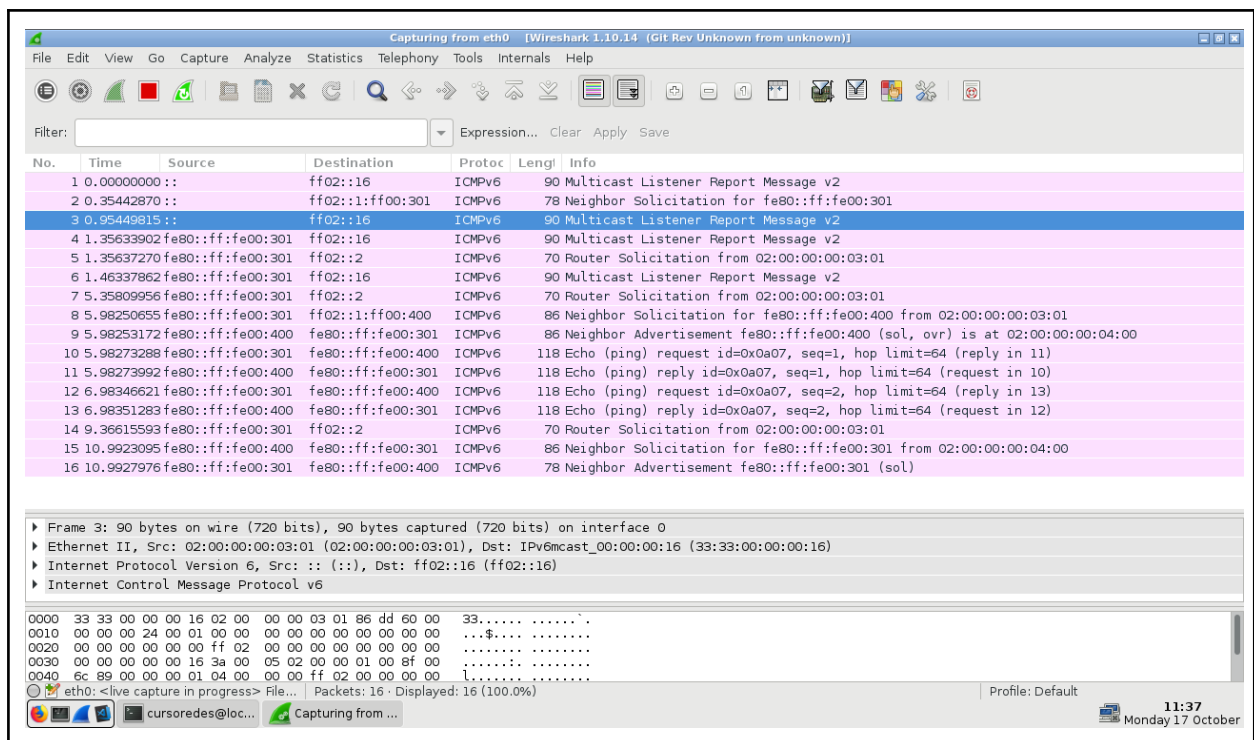
```
        valid_lft forever preferred_lft forever
```

Capturas de pantalla de Wireshark al comprobar la conectividad:

[VM1]



[VM4]



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

```
[VM2]
sudo ip a add fd00:0:0:a::2/64 dev eth0
```

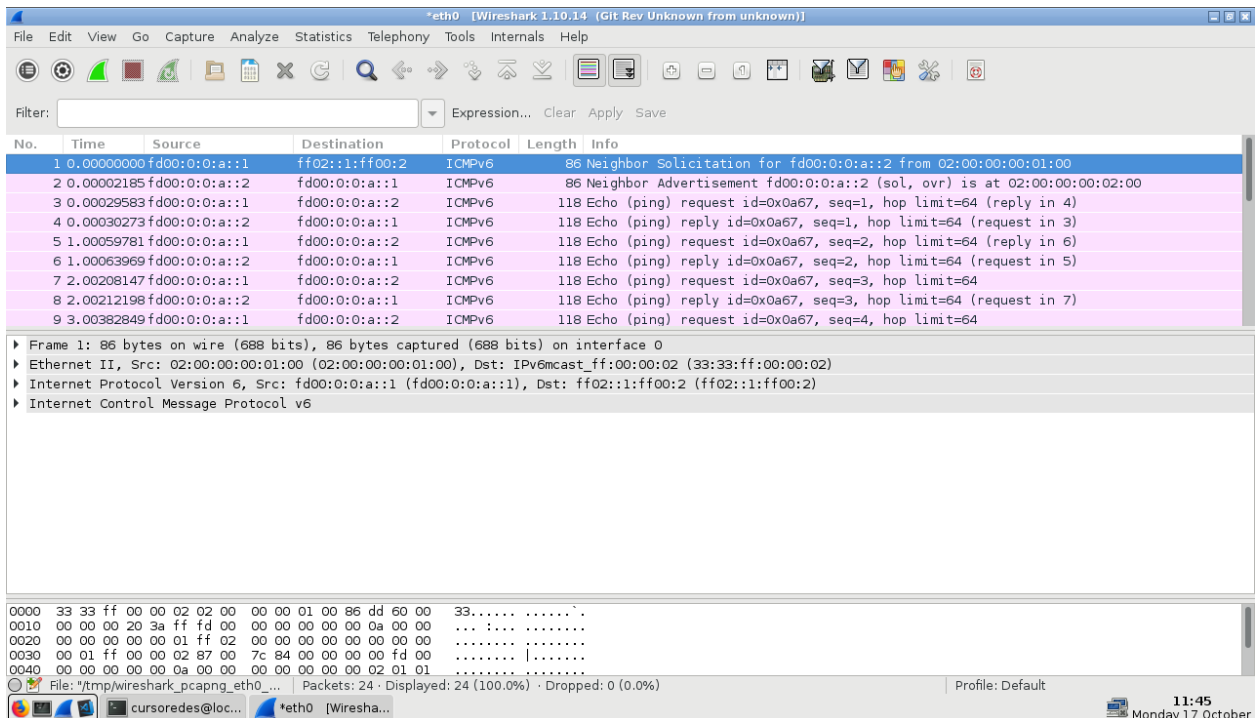
Ejercicio 5 [VM1, VM2]. Comprobar la conectividad entre VM1 y VM2 con la orden `ping6` usando la nueva dirección. Observar los mensajes intercambiados con Wireshark.

```
ping6 fd00:0:0:a::2
PING fd00:0:0:a::2(fd00:0:0:a::2) 56 data bytes
64 bytes from fd00:0:0:a::2: icmp_seq=1 ttl=64 time=0.631 ms
64 bytes from fd00:0:0:a::2: icmp_seq=2 ttl=64 time=0.916 ms
64 bytes from fd00:0:0:a::2: icmp_seq=3 ttl=64 time=0.975 ms
64 bytes from fd00:0:0:a::2: icmp_seq=4 ttl=64 time=0.970 ms
```

```

^c64 bytes from fd00:0:0:a::2: icmp_seq=5 ttl=64 time=0.929 ms
64 bytes from fd00:0:0:a::2: icmp_seq=6 ttl=64 time=0.918 ms
64 bytes from fd00:0:0:a::2: icmp_seq=7 ttl=64 time=0.874 ms
64 bytes from fd00:0:0:a::2: icmp_seq=8 ttl=64 time=1.03 ms
^C
--- fd00:0:0:a::2 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7008ms
rtt min/avg/max/mdev = 0.631/0.906/1.038/0.116 ms

```



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

```

```

[VM4]
sudo ip a 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]

Conectividad con VM1: 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=0.605 ms
64 bytes from fd00:0:0:a::1: icmp_seq=2 ttl=64 time=0.866 ms
64 bytes from fd00:0:0:a::1: icmp_seq=3 ttl=64 time=1.00 ms
^C

```

--- fd00:0:0:a::1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.605/0.826/1.007/0.166 ms

Conectividad con VM4: 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=0.663 ms
64 bytes from fd00:0:0:b::4: icmp_seq=2 ttl=64 time=1.03 ms
^C

--- fd00:0:0:b::4 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.663/0.850/1.037/0.187 ms

[VM1]
ping6 fd00:0:0:b::4
connect: Network is unreachable
Capturas de Wireshark comprobando la conectividad:

[VM1]

Wireshark 1.10.14 (Git Rev Unknown from unknown) - Capturing from eth0

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

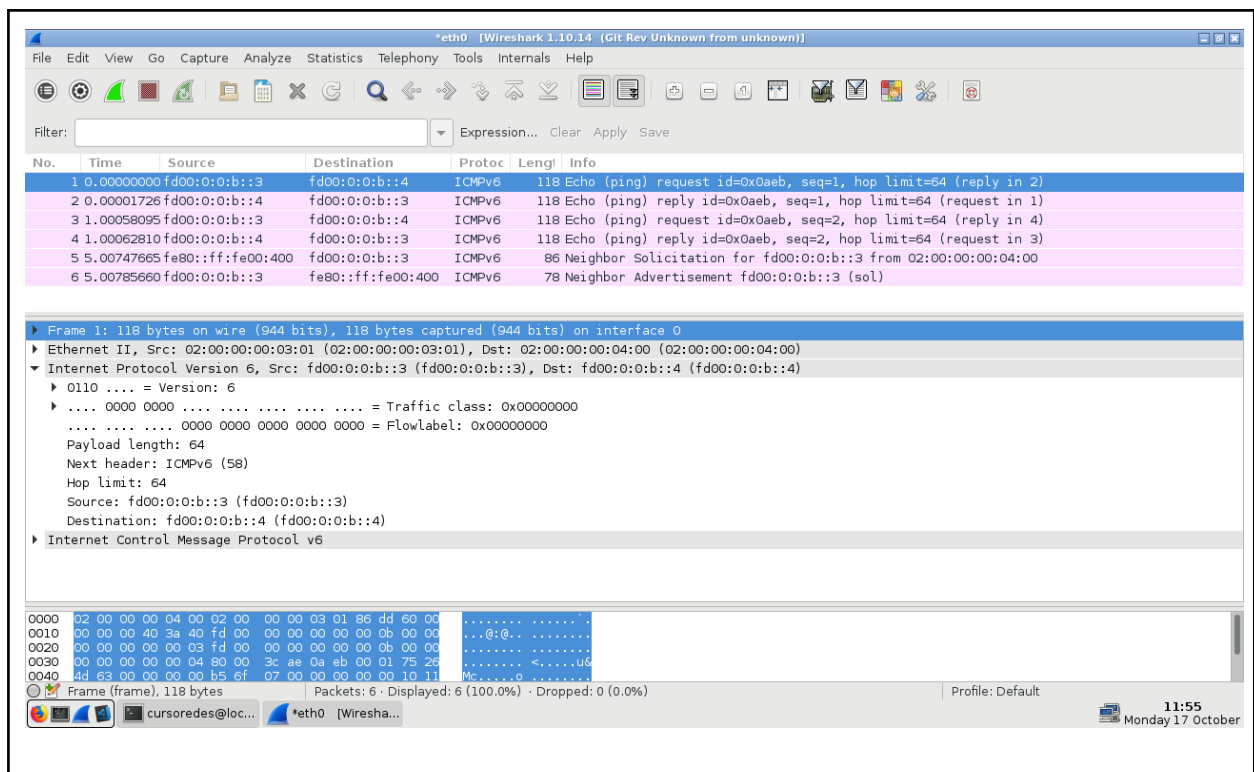
List the available capture interfaces... Expression... Clear Apply Save

No.	Time	Source	Destination	Protoc	Lengt	Info
1	0.00000000	fd00:0:0:a::3	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fd00:0:0:a::1 from 02:00:00:00:03:00
2	0.00001842	fd00:0:0:a::1	fd00:0:0:a::3	ICMPv6	86	Neighbor Advertisement fd00:0:0:a::1 (sol, ovr) is at 02:00:00:00:01:00
3	0.00027605	fd00:0:0:a::3	fd00:0:0:a::1	ICMPv6	118	Echo (ping) request id=0x0aad, seq=1, hop limit=64 (reply in 4)
4	0.00028335	fd00:0:0:a::1	fd00:0:0:a::3	ICMPv6	118	Echo (ping) reply id=0x0aad, seq=1, hop limit=64 (request in 3)
5	1.00068965	fd00:0:0:a::3	fd00:0:0:a::1	ICMPv6	118	Echo (ping) request id=0x0aad, seq=2, hop limit=64 (reply in 6)
6	1.00073213	fd00:0:0:a::1	fd00:0:0:a::3	ICMPv6	118	Echo (ping) reply id=0x0aad, seq=2, hop limit=64 (request in 5)
7	2.00203739	fd00:0:0:a::3	fd00:0:0:a::1	ICMPv6	118	Echo (ping) request id=0x0aad, seq=3, hop limit=64
8	2.00207739	fd00:0:0:a::1	fd00:0:0:a::3	ICMPv6	118	Echo (ping) reply id=0x0aad, seq=3, hop limit=64 (request in 7)
9	5.00617753	fe80::ff:fe00:100	fd00:0:0:a::3	ICMPv6	86	Neighbor Solicitation for fd00:0:0:a::3 from 02:00:00:00:01:00
10	5.00705916	fd00:0:0:a::3	fe80::ff:fe00:100	ICMPv6	78	Neighbor Advertisement fd00:0:0:a::3 (sol)
11	10.0176274	fe80::ff:fe00:300	fe80::ff:fe00:100	ICMPv6	86	Neighbor Solicitation for fe80::ff:fe00:100 from 02:00:00:00:03:00
12	10.0176709	fe80::ff:fe00:100	fe80::ff:fe00:300	ICMPv6	78	Neighbor Advertisement fe80::ff:fe00:100 (sol)
13	15.0221420	fe80::ff:fe00:100	fe80::ff:fe00:300	ICMPv6	86	Neighbor Solicitation for fe80::ff:fe00:300 from 02:00:00:00:01:00
14	15.0229275	fe80::ff:fe00:300	fe80::ff:fe00:100	ICMPv6	78	Neighbor Advertisement fe80::ff:fe00:300 (sol)

Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
Ethernet II, Src: 02:00:00:00:03:00 (02:00:00:00:03:00), Dst: IPv6mcast_ff:00:00:01 (33:33:ff:00:00:01)
Internet Protocol Version 6, Src: fd00:0:0:a::3 (fd00:0:0:a::3), Dst: ff02::1:ff00:1 (ff02::1:ff00:1)
Internet Control Message Protocol v6

0000 33 33 ff 00 00 01 02 00 00 00 03 00 86 dd 60 00 33
0010 00 00 00 20 3a ff fd 00 00 00 00 00 0a 00 00 ... : : : : :
0020 00 00 00 00 00 03 ff 02 00 00 00 00 00 00 00 : : : : :
0030 00 01 ff 00 00 01 87 00 7a 84 00 00 00 00 fd 00 : : : : :
0040 00 00 00 00 00 0a 00 00 00 00 00 00 01 01 01 : : : : :
eth0: <live capture in progress> File... Packets: 14 - Displayed: 14 (100.0%) Profile: Default
cursoredes@loc... Capturing from ... 11:52 Monday 17 October

[VM4]



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]
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
fe80::/64 dev eth0 proto kernel metric 256 pref medium
```

```
[Router]
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
```

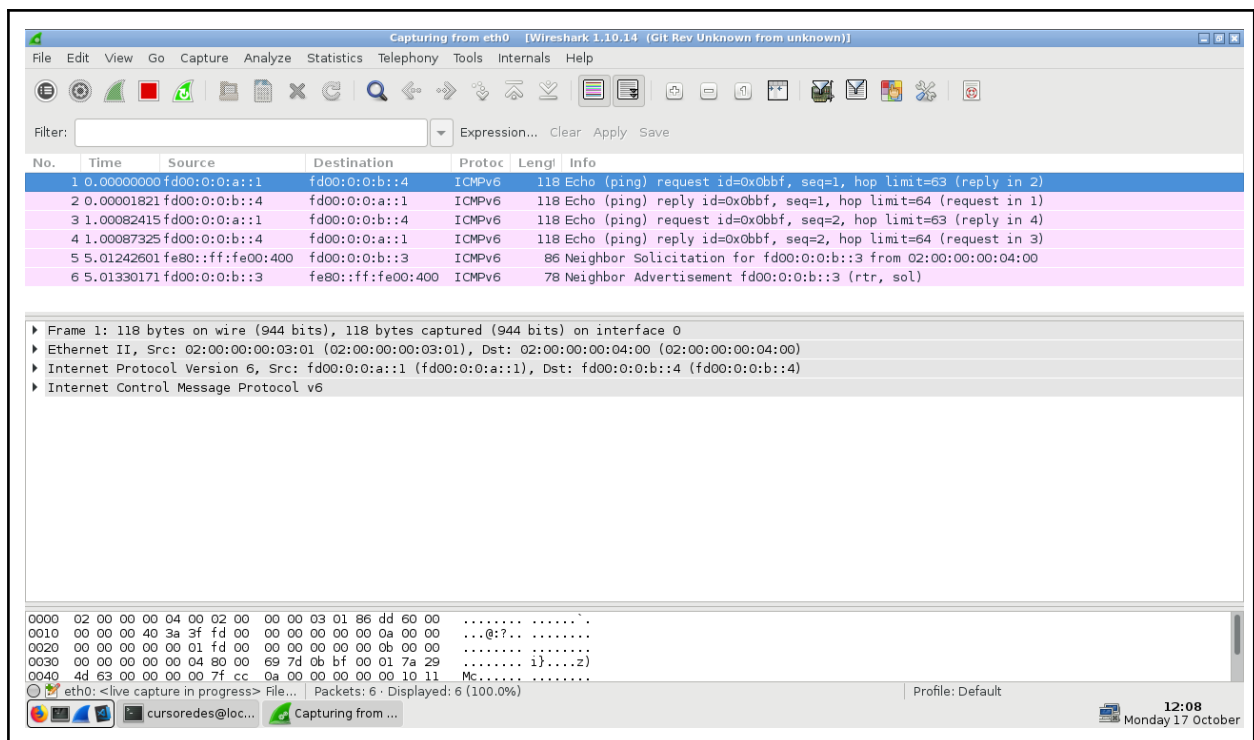
Ejercicio 9 [Router]. Para que Router actúe efectivamente como encaminador, hay que activar el reenvío de paquetes (*packet forwarding*). De forma temporal, se puede activar con el comando `sysctl net.ipv6.conf.all.forwarding=1`.

```
sudo sysctl net.ipv6.conf.all.forwarding=1
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
[VM4]
sudo ip route add default via fd00:0:0:b::3

[VM1]
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=0.727 ms
64 bytes from fd00:0:0:b::4: icmp_seq=2 ttl=63 time=1.79 ms
^C
--- fd00:0:0:b::4 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.727/1.261/1.795/0.534 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	33:33:ff:00:00:03	fd00:0:0:a::1	ff02::1:ff00:3	Neighbor Solicitation
02:00:00:00:03:00	02:00:00:00:01:00	fd00:0:0:a::3	fd00:0:0:a::1	Neighbor Advertisement
02:00:00:00:01:00	33:33:ff:00:00:03	fe80::ff:fe00:100	ff02::1:ff00:3	Neighbor Solicitation
02:00:00:00:03:00	02:00:00:00:01:00	fd00:0:0:a::3	fe80::ff:fe00:100	Neighbor Advertisement
02:00:00:00:01:00	02:00:00:00:03:00	fd00:0:0:a::1	fd00:0:0:b::4	Echo Request
02:00:00:00:03:00	02:00:00:00:01:00	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
02:00:00:00:03:01	33:33:ff:00:00:04	fe80::ff:fe00:301	ff02::1:ff00:4	Neighbor Solicitation
02:00:00:00:04:00	02:00:00:00:03:01	fd00:0:0:b::4	fe80::ff:fe00:301	Neighbor Advertisement
02:00:00:00:03:01	02:00:00:00:04:00	fd00:0:0:a::1	fd00:0:0:b::4	Echo Request

02:00:00:00:04:00	02:00:00:00:03:01	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 (si tiene)>
DEVICE=<nombre del interfaz>
```

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

```
[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.

```
[Router]
ifdown eth0
ifdown eth1
ifup eth0
ifup eth1
ip addr

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 02:00:00:00:03:00 brd ff:ff:ff:ff:ff:ff
inet6 fd00:0:0:a::3/64 scope global
    valid_lft forever preferred_lft forever
inet6 fe80::ff:fe00:300/64 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 02:00:00:00:03:01 brd ff:ff:ff:ff:ff:ff
inet6 fd00:0:0:b::3/64 scope global
    valid_lft forever preferred_lft forever
inet6 fe80::ff:fe00:301/64 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`.

Utilizamos el siguiente comando en todas las máquinas:

```
ip link set dev 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

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.

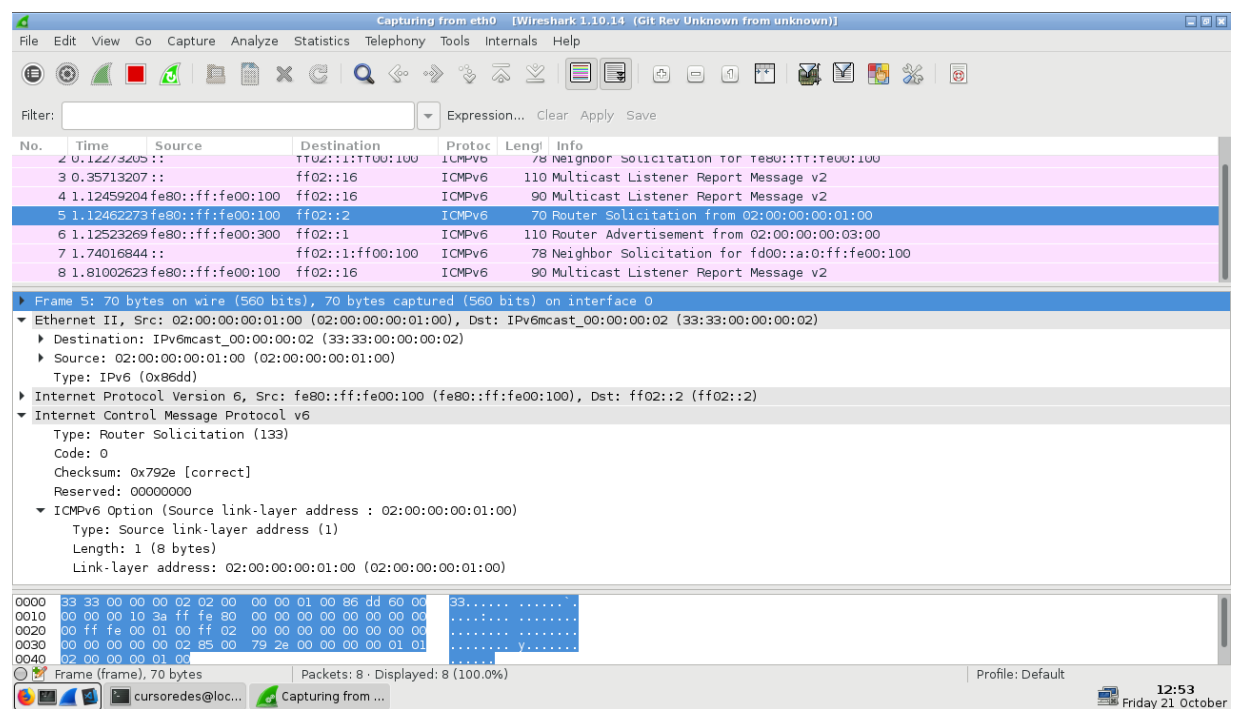
Copiar la dirección asignada.

```
inet6 fd00::b:0:ff:fe00:400/64 scope global mngtmpaddr dynamic
    valid_lft 2591998sec preferred_lft 604798sec
inet6 fe80::ff:fe00:400/64 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`.



[VM1]

```
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:00:01:00
      inet 224.0.0.1
      inet6 ff02::1:ff00:100 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.

Copiar la salida del comando ip addr con la dirección temporal.

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 02:00:00:00:01:00 brd ff:ff:ff:ff:ff:ff
    inet6 fd00::a:5c90:a901:a46a:1ad4/64 scope global temporary tentative dynamic
        valid_lft 604800sec preferred_lft 85800sec
    inet6 fd00::a:0:ff:fe00:100/64 scope global tentative mngtmpaddr dynamic
        valid_lft 2592000sec preferred_lft 604800sec
    inet6 fe80::ff:fe00:100/64 scope link
        valid_lft forever preferred_lft forever
```

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.

Los mensajes de solicitud y respuesta de eco, solicitud y anuncio de encaminador y solicitud anuncio de vecino ya han sido vistos y hay capturas en los ejercicios anteriores.