

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



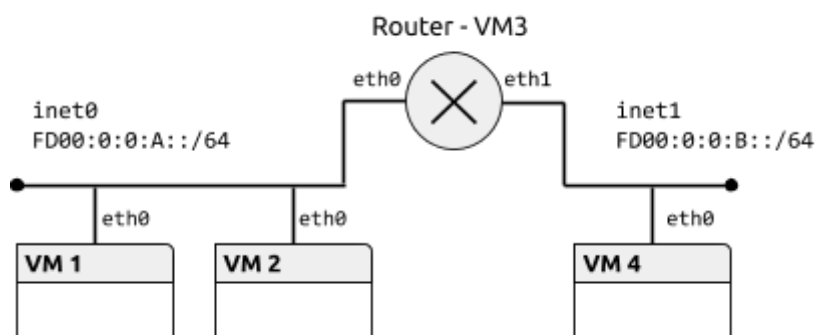
Activar el **portapapeles bidireccional** (menú Dispositivos) en las máquinas virtuales.

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.

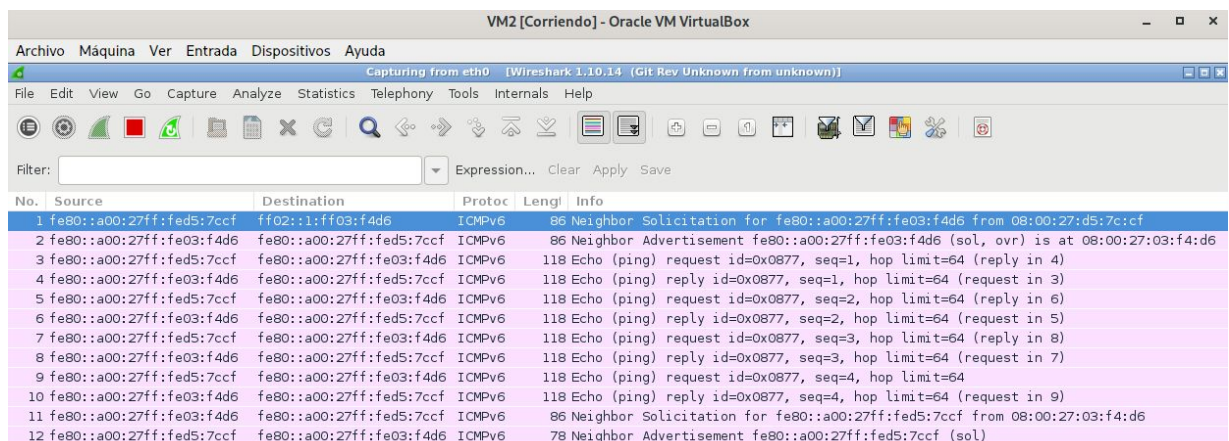
```
En VM1:
$ ip link set eth0 up
$ ip address
RESULTADO:
1. eth0 -> inet6 fe80::a00:27ff:fed5:7ccf/64
```

```
En VM2
$ ip link set eth0 up
$ ip address
RESULTADO:
1. eth0 -> inet6 fe80::a00:27ff:fe03:f4d6/64
```

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.

```
VM1:
$ping6 fe80::a00:27ff:fe03:f4d6 -I eth0
```



| No. | Source | Destination | Protoc | Lengt | Info |
|-----|--------------------------|--------------------------|--------|-------|--|
| 1 | fe80::a00:27ff:fed5:7ccf | ff02::1:ff03:f4d6 | ICMPv6 | 86 | Neighbor Solicitation for fe80::a00:27ff:fe03:f4d6 from 08:00:27:d5:7c:cf |
| 2 | fe80::a00:27ff:fe03:f4d6 | fe80::a00:27ff:fed5:7ccf | ICMPv6 | 86 | Neighbor Advertisement fe80::a00:27ff:fe03:f4d6 (sol, ovr) is at 08:00:27:03:f4:d6 |
| 3 | fe80::a00:27ff:fed5:7ccf | fe80::a00:27ff:fe03:f4d6 | ICMPv6 | 118 | Echo (ping) request id=0x0877, seq=1, hop limit=64 (reply in 4) |
| 4 | fe80::a00:27ff:fe03:f4d6 | fe80::a00:27ff:fed5:7ccf | ICMPv6 | 118 | Echo (ping) reply id=0x0877, seq=1, hop limit=64 (request in 3) |
| 5 | fe80::a00:27ff:fed5:7ccf | fe80::a00:27ff:fe03:f4d6 | ICMPv6 | 118 | Echo (ping) request id=0x0877, seq=2, hop limit=64 (reply in 6) |
| 6 | fe80::a00:27ff:fe03:f4d6 | fe80::a00:27ff:fed5:7ccf | ICMPv6 | 118 | Echo (ping) reply id=0x0877, seq=2, hop limit=64 (request in 5) |
| 7 | fe80::a00:27ff:fed5:7ccf | fe80::a00:27ff:fe03:f4d6 | ICMPv6 | 118 | Echo (ping) request id=0x0877, seq=3, hop limit=64 (reply in 8) |
| 8 | fe80::a00:27ff:fe03:f4d6 | fe80::a00:27ff:fed5:7ccf | ICMPv6 | 118 | Echo (ping) reply id=0x0877, seq=3, hop limit=64 (request in 7) |
| 9 | fe80::a00:27ff:fed5:7ccf | fe80::a00:27ff:fe03:f4d6 | ICMPv6 | 118 | Echo (ping) request id=0x0877, seq=4, hop limit=64 |
| 10 | fe80::a00:27ff:fe03:f4d6 | fe80::a00:27ff:fed5:7ccf | ICMPv6 | 118 | Echo (ping) reply id=0x0877, seq=4, hop limit=64 (request in 9) |
| 11 | fe80::a00:27ff:fe03:f4d6 | fe80::a00:27ff:fed5:7ccf | ICMPv6 | 86 | Neighbor Solicitation for fe80::a00:27ff:fed5:7ccf from 08:00:27:03:f4:d6 |
| 12 | fe80::a00:27ff:fed5:7ccf | fe80::a00:27ff:fe03:f4d6 | ICMPv6 | 78 | Neighbor Advertisement fe80::a00:27ff:fed5:7ccf (sol) |

| No. | Source | Destination | Protoc | Lengt | Info |
|-----|--------------------------|-------------------|--------|-------|---|
| 1 | fe80::a00:27ff:fed5:7ccf | ff02::1:ff03:f4d6 | ICMPv6 | 86 | Neighbor Solicitation for fe80::a00:27ff:fe03:f4d6 from 08:00:27:d5:7c:cf |

```

▶ Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
▶ Ethernet II, Src: CadmusCo_d5:7c:cf (08:00:27:d5:7c:cf), Dst: IPv6mcast_ff:03:f4:d6 (33:33:ff:03:f4:d6)
▼ Internet Protocol Version 6, Src: fe80::a00:27ff:fed5:7ccf (fe80::a00:27ff:fed5:7ccf), Dst: ff02::1:ff03:f4d6 (ff02::1:ff03:f4d6)
  ▶ 0110 .... = Version: 6
  ▶ .... 0000 0000 .... = Traffic class: 0x00000000
  .... 0000 0000 0000 0000 = Flowlabel: 0x00000000
  Payload length: 32
  Next header: ICMPv6 (58)
  Hop limit: 255
  Source: fe80::a00:27ff:fed5:7ccf (fe80::a00:27ff:fed5:7ccf)
  [Source SA MAC: CadmusCo_d5:7c:cf (08:00:27:d5:7c:cf)]
  Destination: ff02::1:ff03:f4d6 (ff02::1:ff03:f4d6)
▶ Internet Control Message Protocol v6

```

Observar bien los mensajes enviados y recibidos:

- Neighbor Solicitation
- Neighbor Advertisement
- Request/Reply

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.

En Router (VM3):

```
$sudo ip link set eth0 up
$ sudo ip link set eth1 up
$ sudo ip addr
```

RESULTADO:

1. eth0 -> inet6 fe80::a00:27ff:fe57:6ec/64
2. eth1 -> inet6 fe80::a00:27ff:fe19:3181/64

En VM4:

```
$ sudo ip link set eth0 up
$ sudo ip addr
```

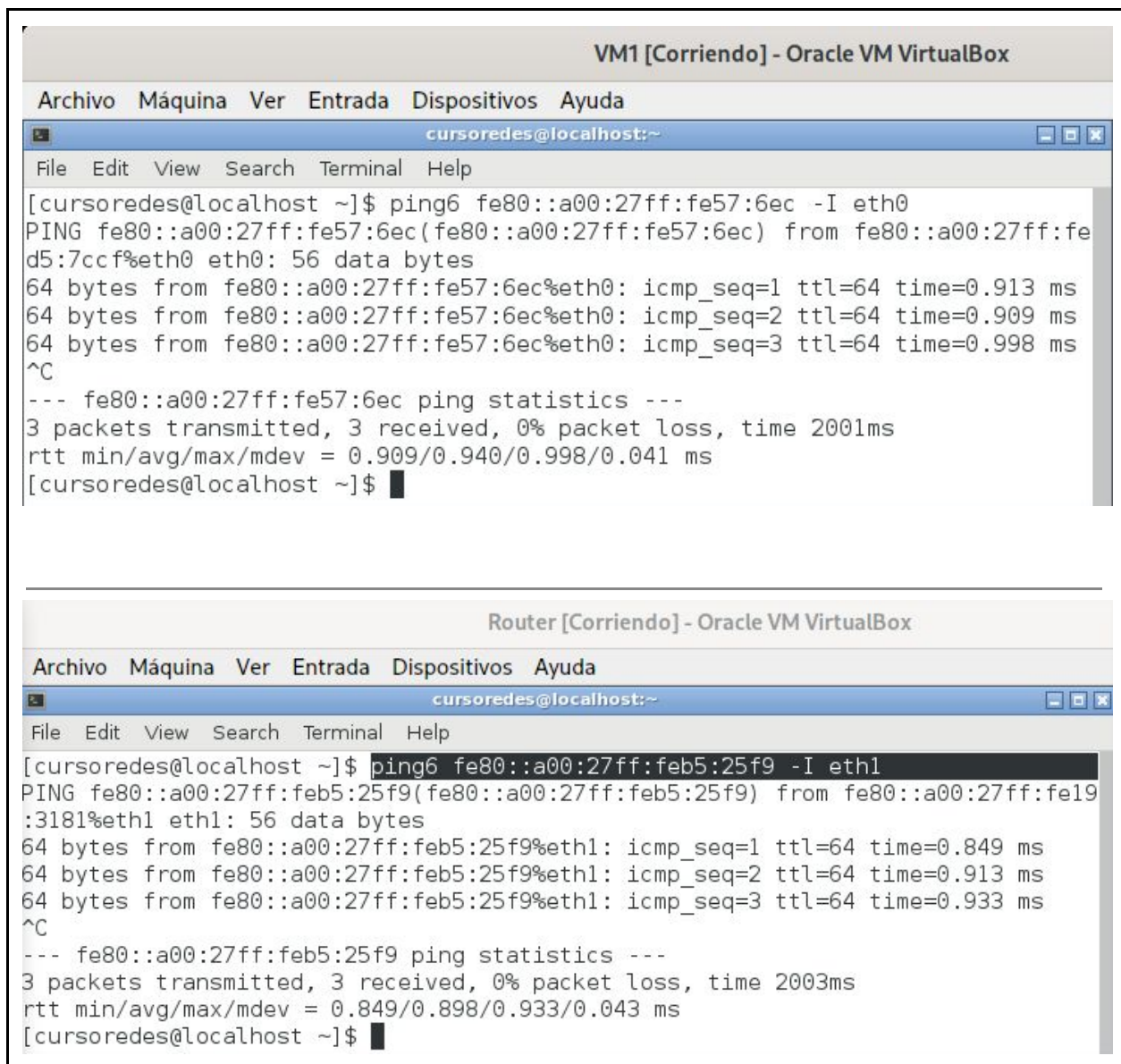
RESULTADO:

1. eth0 -> inet6 fe80::a00:27ff:feb5:25f9/64

Para probar la conectividad:

Desde VM1 a VM3(router): ping6 fe80::a00:27ff:fe57:6ec -I eth0

Desde VM3(router) a VM4: ping6 fe80::a00:27ff:feb5:25f9 -I 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.

```
Desde VM1:
$ip link set eth0 up
$ip a add fd00:0:0:a::1/64 dev eth0
```

```
Desde VM2:
$ip link set eth0 up
$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.

| No. | Source | Destination | Protoc | Leng | Info |
|-----|---------------|----------------|--------|------|---|
| 1 | fd00:0:0:a::1 | ff02::1:ff00:2 | ICMPv6 | 86 | Neighbor Solicitation for fd00:0:0:a::2 from 08:00:27:d5:7c:cf |
| 2 | fd00:0:0:a::2 | fd00:0:0:a::1 | ICMPv6 | 86 | Neighbor Advertisement fd00:0:0:a::2 (sol, ovr) is at 08:00:27:03:f4:d6 |
| 3 | fd00:0:0:a::1 | fd00:0:0:a::2 | ICMPv6 | 118 | Echo (ping) request id=0x0c36, seq=1, hop limit=64 (reply in 4) |
| 4 | fd00:0:0:a::2 | fd00:0:0:a::1 | ICMPv6 | 118 | Echo (ping) reply id=0x0c36, seq=1, hop limit=64 (request in 3) |
| 5 | fd00:0:0:a::1 | fd00:0:0:a::2 | ICMPv6 | 118 | Echo (ping) request id=0x0c36, seq=2, hop limit=64 (reply in 6) |
| 6 | fd00:0:0:a::2 | fd00:0:0:a::1 | ICMPv6 | 118 | Echo (ping) reply id=0x0c36, seq=2, hop limit=64 (request in 5) |
| 7 | fd00:0:0:a::1 | fd00:0:0:a::2 | ICMPv6 | 118 | Echo (ping) request id=0x0c36, seq=3, hop limit=64 (reply in 8) |

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.

```
Desde VM3(router):
$ sudo ip link set eth0 up
$ sudo ip link set eth1 up
$ sudo ip a add fd00:0:0:a::3/64 dev eth0
$ sudo ip a add fd00:0:0:b::1/64 dev eth1
```

```
Desde VM4:
$ sudo ip link set eth0 up
$ sudo ip a add fd00:0:0:b::2/64 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.

```
Desde VM3(router):
$ ping6 fd00:0:0:a::1 -I eth0
$ ping6 fd00:0:0:b::2 -I eth1
```

```
[cursoredes@localhost ~]$ ping6 fd00:0:0:a::1 -I eth0
PING fd00:0:0:a::1(fd00:0:0:a::1) from fd00:0:0:a::3 eth0: 56 data bytes
64 bytes from fd00:0:0:a::1: icmp_seq=1 ttl=64 time=0.824 ms
64 bytes from fd00:0:0:a::1: icmp_seq=2 ttl=64 time=0.958 ms
64 bytes from fd00:0:0:a::1: icmp_seq=3 ttl=64 time=1.05 ms
^C
--- fd00:0:0:a::1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.824/0.944/1.050/0.092 ms
[cursoredes@localhost ~]$ ping6 fd00:0:0:b::2 -I eth1
PING fd00:0:0:b::2(fd00:0:0:b::2) from fd00:0:0:b::1 eth1: 56 data bytes
64 bytes from fd00:0:0:b::2: icmp_seq=1 ttl=64 time=1.53 ms
64 bytes from fd00:0:0:b::2: icmp_seq=2 ttl=64 time=0.699 ms
64 bytes from fd00:0:0:b::2: icmp_seq=3 ttl=64 time=0.838 ms
^C
--- fd00:0:0:b::2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.699/1.025/1.538/0.367 ms
```

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.

```
$ip -6 route show
(otro alternativo que se ve mejor -> $route -6)
```

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`.

```
Desde VM1:
sudo ip -6 route add fd00:0:0:b::/64 via fd00:0:0:a::3

Desde VM2:
sudo ip -6 route add fd00:0:0:b::/64 via fd00:0:0:a::3

Desde VM4:
sudo ip -6 route add fd00:0:0:a::/64 via fd00:0:0:b::1

Comprobar conectividad desde VM1 hasta VM4:
VM1: ping6 fd00:0:0:b::2 -I eth0
```



```
[cursoredes@localhost ~]$ ping6 fd00:0:0:b::2 -I eth0
PING fd00:0:0:b::2(fd00:0:0:b::2) from fd00:0:0:a::1 eth0: 56 data bytes
64 bytes from fd00:0:0:b::2: icmp_seq=1 ttl=63 time=1.68 ms
64 bytes from fd00:0:0:b::2: icmp_seq=2 ttl=63 time=1.73 ms
64 bytes from fd00:0:0:b::2: icmp_seq=3 ttl=63 time=1.82 ms
64 bytes from fd00:0:0:b::2: icmp_seq=4 ttl=63 time=2.02 ms
^C
--- fd00:0:0:b::2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 1.681/1.817/2.025/0.137 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 |
|------------|-------------|-------------|--------------------------|------------------------|
| MAC (VM1) | Broadcast | Ip (VM1) | Multicast Solicited Node | Neighbor Solicitation |
| MAC (VM3) | MAC (VM1) | Ip (VM3) | Ip (VM1) | Neighbor Advertisement |
| MAC (VM1) | MAC (VM3) | Ip (VM1) | Ip (VM4) | Echo request |
| MAC (VM3) | MAC (VM1) | Ip (VM4) | Ip (VM1) | Echo reply |

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

| MAC Origen | MAC Destino | IPv6 Origen | IPv6 Destino | ICMPv6 Tipo |
|------------|-------------|-------------|--------------------------|------------------------|
| MAC (VM4) | Broadcast | Ip (VM4) | Multicast Solicited Node | Neighbor Solicitation |
| MAC (VM3) | MAC (VM4) | Ip (VM3) | Ip (VM4) | Neighbor Advertisement |
| MAC (VM4) | MAC (VM3) | Ip (VM4) | Ip (VM1) | Echo request |
| MAC (VM3) | MAC (VM4) | Ip (VM1) | Ip (VM4) | Echo reply |

eth0:

Router [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:d5:7c:cf |
| 2 | 0.000029800 | fd00:0:0:a::3 | fd00:0:0:a::1 | ICMPv6 | 86 | Neighbor Advertisement fd00:0:0:a::3 (rtr, sol, ovr) is at 08:00:27:57:06:ec |
| 3 | 0.000044028 | fe80::a00:27ff:fed5:7ccf | ff02::1:ff00:3 | ICMPv6 | 86 | Neighbor Solicitation for fd00:0:0:a::3 from 08:00:27:d5:7c:cf |
| 4 | 0.000048397 | fd00:0:0:a::3 | fe80::a00:27ff:fed5:7ccf | ICMPv6 | 86 | Neighbor Advertisement fd00:0:0:a::3 (rtr, sol, ovr) is at 08:00:27:57:06:ec |
| 5 | 0.000052296 | fe80::a00:27ff:fed5:7ccf | ff02::1:ff00:3 | ICMPv6 | 86 | Neighbor Solicitation for fd00:0:0:a::3 from 08:00:27:d5:7c:cf |
| 6 | 0.000054634 | fd00:0:0:a::3 | fe80::a00:27ff:fed5:7ccf | ICMPv6 | 86 | Neighbor Advertisement fd00:0:0:a::3 (rtr, sol, ovr) is at 08:00:27:57:06:ec |
| 7 | 0.000328387 | fd00:0:0:a::1 | fd00:0:0:b::2 | ICMPv6 | 118 | Echo (ping) request id=0x168d, seq=1, hop limit=64 (reply in 8) |
| 8 | 0.000644283 | fd00:0:0:b::2 | fd00:0:0:a::1 | ICMPv6 | 118 | Echo (ping) reply id=0x168d, seq=1, hop limit=63 (request in 7) |
| 9 | 1.002339360 | fd00:0:0:a::1 | fd00:0:0:b::2 | ICMPv6 | 118 | Echo (ping) request id=0x168d, seq=2, hop limit=64 (reply in 10) |
| 10 | 1.003332057 | fd00:0:0:b::2 | fd00:0:0:a::1 | ICMPv6 | 118 | Echo (ping) reply id=0x168d, seq=2, hop limit=63 (request in 9) |
| 11 | 2.004370680 | fd00:0:0:a::1 | fd00:0:0:b::2 | ICMPv6 | 118 | Echo (ping) request id=0x168d, seq=3, hop limit=64 (reply in 12) |
| 12 | 2.005261816 | fd00:0:0:b::2 | fd00:0:0:a::1 | ICMPv6 | 118 | Echo (ping) reply id=0x168d, seq=3, hop limit=63 (request in 11) |
| 13 | 3.006988746 | fd00:0:0:a::1 | fd00:0:0:b::2 | ICMPv6 | 118 | Echo (ping) request id=0x168d, seq=4, hop limit=64 |
| 14 | 3.007870878 | fd00:0:0:b::2 | fd00:0:0:a::1 | ICMPv6 | 118 | Echo (ping) reply id=0x168d, seq=4, hop limit=63 (request in 13) |
| 15 | 4.009747888 | fd00:0:0:a::1 | fd00:0:0:b::2 | ICMPv6 | 118 | Echo (ping) request id=0x168d, seq=5, hop limit=64 (reply in 16) |
| 16 | 4.010646913 | fd00:0:0:b::2 | fd00:0:0:a::1 | ICMPv6 | 118 | Echo (ping) reply id=0x168d, seq=5, hop limit=63 (request in 15) |
| 17 | 5.012379627 | fd00:0:0:a::1 | fd00:0:0:b::2 | ICMPv6 | 118 | Echo (ping) request id=0x168d, seq=6, hop limit=64 (reply in 18) |

Frame 3: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0

Ethernet II, Src: CadmusCo_d5:7c:cf (08:00:27:d5:7c:cf), Dst: IPv6mcast_ff:00:00:03 (33:33:ff:00:00:03)

Internet Protocol Version 6, Src: fe80::a00:27ff:fed5:7ccf (fe80::a00:27ff:fed5:7ccf), Dst: ff02::1:ff00:3 (ff02::1:ff00:3)

Internet Control Message Protocol v6

eth1:

Router [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 | Protocol | Length | Info |
|-----|-------------|--------------------------|--------------------------|----------|--------|--|
| 1 | 0.000000000 | fd00:0:0:a::1 | fd00:0:0:b::2 | ICMPv6 | 118 | Echo (ping) request id=0x168d, seq=1, hop limit=63 (reply in 2) |
| 2 | 0.000300236 | fd00:0:0:b::2 | fd00:0:0:a::1 | ICMPv6 | 118 | Echo (ping) reply id=0x168d, seq=1, hop limit=64 (request in 1) |
| 3 | 1.002037057 | fd00:0:0:a::1 | fd00:0:0:b::2 | ICMPv6 | 118 | Echo (ping) request id=0x168d, seq=2, hop limit=63 |
| 4 | 1.002970079 | fd00:0:0:b::2 | fd00:0:0:a::1 | ICMPv6 | 118 | Echo (ping) reply id=0x168d, seq=2, hop limit=64 (request in 3) |
| 5 | 2.004063833 | fd00:0:0:a::1 | fd00:0:0:b::2 | ICMPv6 | 118 | Echo (ping) request id=0x168d, seq=3, hop limit=63 (reply in 6) |
| 6 | 2.004898430 | fd00:0:0:b::2 | fd00:0:0:a::1 | ICMPv6 | 118 | Echo (ping) reply id=0x168d, seq=3, hop limit=64 (request in 5) |
| 7 | 3.006684234 | fd00:0:0:a::1 | fd00:0:0:b::2 | ICMPv6 | 118 | Echo (ping) request id=0x168d, seq=4, hop limit=63 (reply in 8) |
| 8 | 3.007511734 | fd00:0:0:b::2 | fd00:0:0:a::1 | ICMPv6 | 118 | Echo (ping) reply id=0x168d, seq=4, hop limit=64 (request in 7) |
| 9 | 4.009439669 | fd00:0:0:a::1 | fd00:0:0:b::2 | ICMPv6 | 118 | Echo (ping) request id=0x168d, seq=5, hop limit=63 (reply in 10) |
| 10 | 4.010284505 | fd00:0:0:b::2 | fd00:0:0:a::1 | ICMPv6 | 118 | Echo (ping) reply id=0x168d, seq=5, hop limit=64 (request in 9) |
| 11 | 5.012074230 | fd00:0:0:a::1 | fd00:0:0:b::2 | ICMPv6 | 118 | Echo (ping) request id=0x168d, seq=6, hop limit=63 (reply in 14) |
| 12 | 5.012255575 | fe80::a00:27ff:feb5:25f9 | fd00:0:0:b::1 | ICMPv6 | 86 | Neighbor Solicitation for fd00:0:0:b::1 from 08:00:27:b5:25:f9 |
| 13 | 5.012287343 | fe80::a00:27ff:feb5:25f9 | fd00:0:0:b::1 | ICMPv6 | 78 | Neighbor Advertisement fd00:0:0:b::1 (rtr, sol) |
| 14 | 5.012654013 | fd00:0:0:b::2 | fd00:0:0:a::1 | ICMPv6 | 118 | Echo (ping) reply id=0x168d, seq=6, hop limit=64 (request in 11) |
| 15 | 5.014538850 | fe80::a00:27ff:fe19:3181 | fd00:0:0:b::2 | ICMPv6 | 86 | Neighbor Solicitation for fd00:0:0:b::2 from 08:00:27:19:31:81 |
| 16 | 5.015226056 | fd00:0:0:b::2 | fe80::a00:27ff:fe19:3181 | ICMPv6 | 78 | Neighbor Advertisement fd00:0:0:b::2 (sol) |
| 17 | 6.014383803 | fd00:0:0:a::1 | fd00:0:0:b::2 | ICMPv6 | 118 | Echo (ping) request id=0x168d, seq=7, hop limit=63 (reply in 18) |
| 18 | 6.015208009 | fd00:0:0:b::2 | fd00:0:0:a::1 | ICMPv6 | 118 | Echo (ping) reply id=0x168d, seq=7, hop limit=64 (request in 17) |
| 19 | 7.017166468 | fd00:0:0:a::1 | fd00:0:0:b::2 | ICMPv6 | 118 | Echo (ping) request id=0x168d, seq=8, hop limit=63 (reply in 20) |
| 20 | 7.017898199 | fd00:0:0:b::2 | fd00:0:0:a::1 | ICMPv6 | 118 | Echo (ping) reply id=0x168d, seq=8, hop limit=64 (request in 19) |
| 21 | 8.019889428 | fd00:0:0:a::1 | fd00:0:0:b::2 | ICMPv6 | 118 | Echo (ping) request id=0x168d, seq=9, hop limit=63 (reply in 22) |
| 22 | 8.020703241 | fd00:0:0:b::2 | fd00:0:0:a::1 | ICMPv6 | 118 | Echo (ping) reply id=0x168d, seq=9, hop limit=64 (request in 21) |

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>
```

Ampliación de Sistemas Operativos y Redes

8

Contenido del fichero ifcfg-eth0 y ifcfg-eth1 respectivamente:

```
TYPE=Ethernet
BOOTPROTO=none
IPV6ADDR=fd00:0:0:b::1/64
IPV6_DEFAULTGW=
DEVICE=eth0
```

```
TYPE=Ethernet
BOOTPROTO=none
IPV6ADDR=fd00:0:0:b::1/64
IPV6_DEFAULTGW=
DEVICE=eth1
```

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

Primero, elimino las interfaces...

```
$ sudo ip link set eth0 down
$ sudo ip link set eth1 down
```

```
[cursoredes@localhost network-scripts]$ 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> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether 08:00:27:57:06:ec brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether 08:00:27:19:31:81 brd ff:ff:ff:ff:ff:ff
```

Segundo, levanto la configuración anterior...

```
$ sudo ifup ifcfg-eth0
$ sudo ifup ifcfg-eth1
```

```
[cursoredes@localhost network-scripts]$ 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 08:00:27:57:06:ec brd ff:ff:ff:ff:ff:ff
    inet6 fd00:0:0:b::1/64 scope global
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe57:6ec/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 08:00:27:19:31:81 brd ff:ff:ff:ff:ff:ff
    inet6 fd00:0:0:b::1/64 scope global
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe19:3181/64 scope link
        valid_lft forever preferred_lft forever
```

Finalmente, vuelvo a quitar las interfaces pero esta vez usando la configuración...

```
$ sudo ifdown ifcfg-eth0
$ sudo ifdown ifcfg-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`.

```
En VM1, VM2 y VM4:
$ sudo 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.

```
[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:b5:25:f9 brd ff:ff:ff:ff:ff:ff
    inet6 fd00::b:a00:27ff:feb5:25f9/64 scope global tentative mngtmpaddr dynamic
        valid_lft 2591999sec preferred_lft 604799sec
    inet6 fe80::a00:27ff:feb5:25f9/64 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`.

Pasos a seguir...

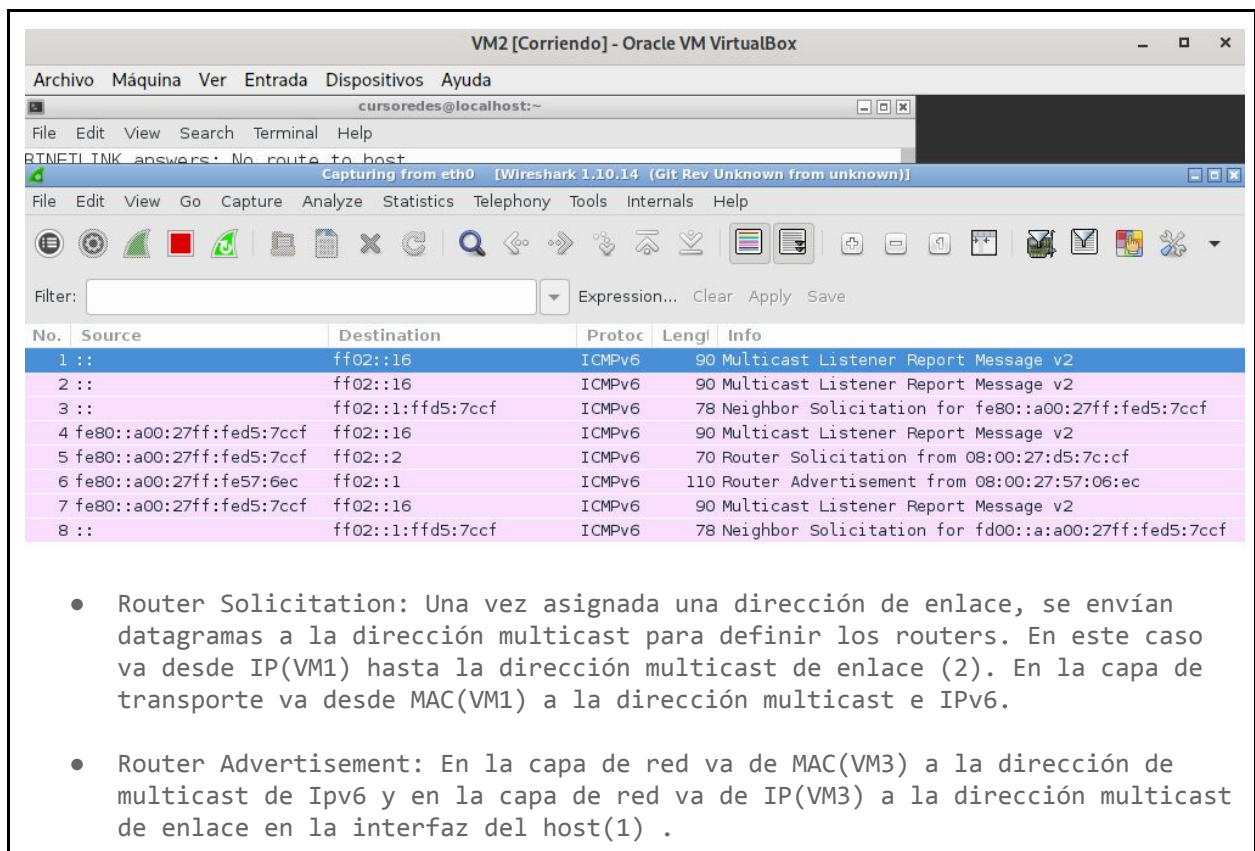
En VM2:

```
$ip link set eth0 up
$ip addr
```

En VM1:

```
$ip link set eth0 up
$ip addr
```

Captura de pantalla de Wireshark...



VM2 [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

cursoredes@localhost:~

File Edit View Search Terminal Help

RTNETLINK answers: No route to host

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. | Source | Destination | Protoc | Lengt | Info |
|-----|--------------------------|------------------|--------|-------|--|
| 1 | :: | ff02::16 | ICMPv6 | 90 | Multicast Listener Report Message v2 |
| 2 | :: | ff02::16 | ICMPv6 | 90 | Multicast Listener Report Message v2 |
| 3 | :: | ff02::1:ff5:7ccf | ICMPv6 | 78 | Neighbor Solicitation for fe80::a00:27ff:fed5:7ccf |
| 4 | fe80::a00:27ff:fed5:7ccf | ff02::16 | ICMPv6 | 90 | Multicast Listener Report Message v2 |
| 5 | fe80::a00:27ff:fed5:7ccf | ff02::2 | ICMPv6 | 70 | Router Solicitation from 08:00:27:d5:7c:cf |
| 6 | fe80::a00:27ff:fe57:6ec | ff02::1 | ICMPv6 | 110 | Router Advertisement from 08:00:27:57:06:ec |
| 7 | fe80::a00:27ff:fed5:7ccf | ff02::16 | ICMPv6 | 90 | Multicast Listener Report Message v2 |
| 8 | :: | ff02::1:ff5:7ccf | ICMPv6 | 78 | Neighbor Solicitation for fd00::a:a00:27ff:fed5:7ccf |

- Router Solicitation: Una vez asignada una dirección de enlace, se envían datagramas a la dirección multicast para definir los routers. En este caso va desde IP(VM1) hasta la dirección multicast de enlace (2). En la capa de transporte va desde MAC(VM1) a la dirección multicast e IPv6.
- Router Advertisement: En la capa de red va de MAC(VM3) a la dirección de multicast de Ipv6 y en la capa de red va de IP(VM3) a la dirección multicast de enlace en la interfaz del host(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`.

```
En VM1:
$ sudo ip link set dev eth0 down
$ sudo sysctl -w net.ipv6.conf.eth0.use_tempaddr=2
$ sudo ip link set dev eth0 up
```

```

[cursoredes@localhost ~]$ sudo ip link set dev eth0 down
[cursoredes@localhost ~]$ sudo sysctl -w net.ipv6.conf.eth0.use_tempaddr=2
net.ipv6.conf.eth0.use_tempaddr = 2
[cursoredes@localhost ~]$ sudo ip link set dev eth0 up
[cursoredes@localhost ~]$ ip a
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:d5:7c:cf brd ff:ff:ff:ff:ff:ff
    inet6 fd00::a:78bc:8b5f:1ba:4908/64 scope global temporary dynamic
        valid_lft 604795sec preferred_lft 85795sec
    inet6 fd00::a:a00:27ff:fed5:7ccf/64 scope global mngtmpaddr dynamic
        valid_lft 2591995sec preferred_lft 604795sec
    inet6 fe80::a00:27ff:fed5:7ccf/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 destino inalcanzable (Code: 3)

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

Mensaje de Destino inalcanzable (Network unreachable, Code: 0)...

Desde VM1 hacer un `$ping6 fd00:0:0:b::2 -I eth0` ya que esta dirección había sido cambiada por otra que se generaba sola.

1 0.0000000000 fd00::a:78bc:8b5f:1ba:4908 fd00:0::...Echo (ping) request id=0x1a3f, seq=1, hop limit=64

▶ Frame 1: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0

- ▼ Ethernet II, Src: CadmusCo_d5:7c:cf (08:00:27:d5:7c:cf), Dst: CadmusCo_57:06:ec (08:00:27:57:06:ec)
 - ▶ Destination: CadmusCo_57:06:ec (08:00:27:57:06:ec)
 - ▶ Source: CadmusCo_d5:7c:cf (08:00:27:d5:7c:cf)
 - Type: IPv6 (0x86dd)
- ▼ Internet Protocol Version 6, Src: fd00::a:78bc:8b5f:1ba:4908 (fd00::a:78bc:8b5f:1ba:4908),
 - ▶ 0110 = Version: 6
 - ▶ 0000 0000 = Traffic class: 0x00000000
 - 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
 - Payload length: 64
 - Next header: ICMPv6 (58)
 - Hop limit: 64
 - Source: fd00::a:78bc:8b5f:1ba:4908 (fd00::a:78bc:8b5f:1ba:4908)
 - Destination: fd00:0:0:b::2 (fd00:0:0:b::2)
- ▼ Internet Control Message Protocol v6
 - Type: Echo (ping) request (128)
 - Code: 0
 - Checksum: 0xac30 [correct]
 - Identifier: 0x1a3f
 - Sequence: 1
 - ▶ Data (56 bytes)

Mensaje de Destino inalcanzable (Port unreachable, Code: 3)...

(¿Cual seria?)