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](#_jn57tfptecf9)

[Direcciones de enlace local](#_lkbk11wj05sz)

[Direcciones ULA](#_w2ynjijgd1kw)

[Encaminamiento estático](#_mrt7svf94rl0)

[Configuración persistente](#_28cl96y06kvv)

[Autoconfiguración. Anuncio de prefijos](#_6ljx4h2wuzmd)

[ICMPv6](#_d51xlfk8qctf)

# 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.

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

| *Copiar el comando utilizado y su salida. Copiar una captura de pantalla de Wireshark donde se vean los campos de la cabecera IPv6.*  *[root@localhost ~]#* ***ping6 -c 1 fe80::a00:27ff:fe24:163c%eth0***  *PING fe80::a00:27ff:fe24:163c%eth0(fe80::a00:27ff:fe24:163c%eth0) 56 data bytes*  *64 bytes from fe80::a00:27ff:fe24:163c%eth0: icmp\_seq=1 ttl=64 time=1.64 ms*  *--- fe80::a00:27ff:fe24:163c%eth0 ping statistics ---*  *1 packets transmitted, 1 received, 0% packet loss, time 0ms*  *rtt min/avg/max/mdev = 1.647/1.647/1.647/0.000 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.*  ***[Router - VM1]***  *[root@localhost ~]#* ***ping6 -c 1 fe80::a00:27ff:feb0:12f8%eth0***  *PING fe80::a00:27ff:feb0:12f8%eth0(fe80::a00:27ff:feb0:12f8%eth0) 56 data bytes*  *64 bytes from fe80::a00:27ff:feb0:12f8%eth0: icmp\_seq=1 ttl=64 time=0.947 ms*  *--- fe80::a00:27ff:feb0:12f8%eth0 ping statistics ---*  *1 packets transmitted, 1 received, 0% packet loss, time 0ms*  *rtt min/avg/max/mdev = 0.947/0.947/0.947/0.000 ms*  ***[Router - VM4]***  *[root@localhost ~]#* ***ping6 -c 1 fe80::a00:27ff:fe22:79cd%eth1***  *PING fe80::a00:27ff:fe22:79cd%eth1(fe80::a00:27ff:fe22:79cd%eth1) 56 data bytes*  *64 bytes from fe80::a00:27ff:fe22:79cd%eth1: icmp\_seq=1 ttl=64 time=0.888 ms*  *--- fe80::a00:27ff:fe22:79cd%eth1 ping statistics ---*  *1 packets transmitted, 1 received, 0% packet loss, time 0ms*  *rtt min/avg/max/mdev = 0.888/0.888/0.888/0.000 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]***  *[root@localhost ~]# i****p a add fd00:0:0:a::1/64 dev eth0***  ***[VM2]***  *[root@localhost ~]#* ***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.

***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]***  *[root@localhost ~]#* ***ip a add fd00:0:0:a::3/64 dev eth0***  *[root@localhost ~]#* ***ip a add fd00:0:0:b::3/64 dev eth1***  ***[VM4]***  *[root@localhost ~]#* ***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.

| *Copiar los comandos utilizados.*  ***[Router-VM1]***  *[root@localhost ~]#* ***ping6 -c 1 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=0.642 ms*  *--- fd00:0:0:a::1 ping statistics ---*  *1 packets transmitted, 1 received, 0% packet loss, time 0ms*  *rtt min/avg/max/mdev = 0.642/0.642/0.642/0.000 ms*  ***[Router-VM4]***  *[root@localhost ~]#* ***ping6 -c 1 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=0.659 ms*  *--- fd00:0:0:b::4 ping statistics ---*  *1 packets transmitted, 1 received, 0% packet loss, time 0ms*  *rtt min/avg/max/mdev = 0.659/0.659/0.659/0.000 ms*  ***[VM1-VM4]***  *[root@localhost ~]#* ***ping6 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.

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

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

| *Copiar los comandos utilizados y su salida.*  ***[VM1]***  *[root@localhost ~]#* ***ip route add default via fd00:0:0:a::3***  ***[VM2]***  *[root@localhost ~]# i****p route add default via fd00:0:0:a::3***  ***[VM4]***  *[root@localhost ~]#* ***ip route add default via fd00:0:0:b::3***  *Hacemos ping entre VM1 y VM4:*  *[root@localhost ~]#* ***ping6 -c 1 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=0.907 ms*  *--- fd00:0:0:b::4 ping statistics ---*  *1 packets transmitted, 1 received, 0% packet loss, time 0ms*  *rtt min/avg/max/mdev = 0.907/0.907/0.907/0.000 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:b0:12:f8 | 33:33:ff:00:00:03 | fe80:a00:27ff:feb0:12f8 (VM1 Link Local) | ff02:1:ff00:3 | Neighbour Solicitation |
| 08:00:27:07:61:7f | 08:00:27:b0:12:f8 | fd00:0:0:a::3  (Dir ULA Router eth0) | fe80:a00:27ff:feb0:12f8 | Neighbour Advertisement |
| 08:00:27:b0:12:f8 | 08:00:27:07:61:7f | fd00:0:0:a::1  (Dir ULA VM1) | fd00:0:0:b::4  (Dir ULA VM4) | Echo (ping) request |
| 08:00:27:07:61:7f | 08:00:27:b0:12:f8 | fd00:0:0:b::4  (Dir ULA VM4) | fd00:0:0:a::1  (Dir ULA VM1) | Echo (ping) reply |

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

| **MAC Origen** | **MAC Destino** | **IPv6 Origen** | **IPv6 Destino** | **ICMPv6 Tipo** |
| --- | --- | --- | --- | --- |
| 08:00:27:7d:ad:7e | 33:33:ff:00:00:04 | Fe80::a00:27ff:fe7d:ad:7e (Router Link Local) | ff02:1:ff00:4 | Neighbour Solicitation |
| 08:00:27:22:79:cd | 08:00:27:7d:ad:7e | fd00:0:0:b::4  (Dir ULA VM4) | Fe80::a00:27ff:fe7d:ad:7e (Router Link Local) | Neighbour Advertisement |
| 08:00:27:7d:ad:7e | 08:00:27:22:79:cd | fd00:0:0:a::1  (Dir ULA VM1) | fd00:0:0:b::4  (Dir ULA VM4) | Echo (ping) request |
| 08:00:27:22:79:cd | 08:00:27:7d:ad:7e | fd00:0:0:b::4  (Dir ULA VM4) | fd00:0:0:a::1  (Dir ULA VM1) | Echo (ping) reply |

| *Copiar dos capturas de pantalla de Wireshark con los mensajes anteriores.*  ***[eth0]***    ***[eth1]*** |
| --- |

# Configuración persistente

Las configuraciones realizadas en los apartados anteriores son volátiles y desaparecen cuando se reinician las máquinas. Durante el arranque del sistema se pueden configurar automáticamente los interfaces según la información almacenada en el disco.

***Ejercicio 12 [Router].*** Crear los ficheros ifcfg-eth0 e ifcfg-eth1 en el directorio /etc/sysconfig/network-scripts/ con la configuración de cada interfaz. Usar las siguientes opciones (descritas en /usr/share/doc/initscripts-\*/sysconfig.txt):

| TYPE=Ethernet  BOOTPROTO=none  IPV6ADDR=*<dirección IP en formato CIDR>* IPV6\_DEFAULTGW=*<dirección IP del encaminador por defecto (si tiene)>* DEVICE=*<nombre del interfaz>* |
| --- |

| *Copiar el contenido de los ficheros.*  ***[fichero ifcfg-eth0]***  *TYPE=Ethernet*  *BOOTPROTO=none*  *IPV6ADDR=fd00:0:0:a::3/64*  *DEVICE=eth0*  ***[fichero 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.*  *[root@localhost network-scripts]#* ***ifdown eth0***  *Device 'eth0' successfully disconnected.*  *[root@localhost network-scripts]#* ***ifdown eth1***  *Device 'eth1' successfully disconnected.*  *[root@localhost network-scripts]#* ***ifup eth0***  *Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/6)*  *[root@localhost network-scripts]#* ***ifup eth1***  *Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/7)*  *[root@localhost network-scripts]#* ***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 08:00:27:07:61:7f brd ff:ff:ff:ff:ff:ff*  *inet6 fd00:0:0:a::3/64 scope global noprefixroute*  *valid\_lft forever preferred\_lft forever*  *inet6 fe80::a00:27ff:fe07:617f/64 scope link noprefixroute*  *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 08:00:27:7d:ad:7e brd ff:ff:ff:ff:ff:ff*  *inet6 fd00:0:0:b::3/64 scope global noprefixroute*  *valid\_lft forever preferred\_lft forever*  *inet6 fe80::a00:27ff:fe7d:ad7e/64 scope link noprefixroute*  *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:fea7:5788/64* |
| --- |

***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 con los mensajes.*    *Tanto Router Solicitation como Router Advertisement tienen como MAC destino 33:33:00:00:00:02 y 33:33:00:00:00:01 respectivamente que se corresponden con las direcciones MAC de difusión*  *Router Solicitation tiene como dirección IP origen la dirección de enlace local de VM1 mientras que Router Advertisement tiene como dirección IP origen la dirección de enlace local de Router*  *Por otra parte la dirección IP destino de cada uno de ellos respectivamente son ff02::2 y ff02::1 que se corresponden con la del grupo multicast de todos los nodos del enlace local.*  *[root@localhost ~]# 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:3e:5d:a5*  *inet 224.0.0.1*  *inet6 ff02::1:ff3e:5da5 users 2*  *inet6 ff02::1*  *inet6 ff01::1*  *La salida nos muestra las interfaces que contiene cada grupo multicast* |
| --- |

***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.*  *fd00::a:a00:27ff:fe3e:5da5/64* |
| --- |

# 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 inalcanzable (Code: 3)
* Destino inalcanzable - Puerto inalcanzable (Code: 4)

| *Copiar capturas de pantalla de Wireshark con los tres últimos mensajes.*  *Para generar un mensaje de dirección inalcanzable hacemos ping desde VM1 a la dirección ULA de una red que el router no pueda alcanzar, por ejemplo:*  *[root@localhost ~]#* ***ping6 fd00:0:0:c::1***  *PING fd00:0:0:c::1(fd00:0:0:c::1) 56 data bytes*  *From fd00:0:0:a::3 icmp\_seq=1 Destination unreachable: No route*    *Para obtener un mensaje de destino inalcanzable hacemos ping desde VM1 a una red a la que el router tenga acceso pero a una interfaz que no exista:*  *[root@localhost ~]#* ***ping6 fd00:0:0:b::1***  *PING fd00:0:0:b::1(fd00:0:0:b::1) 56 data bytes*  *From fd00:0:0:a::3 icmp\_seq=1 Destination unreachable: Address unreachable* |
| --- |