

Práctica 1.4. IPv6

Objetivos

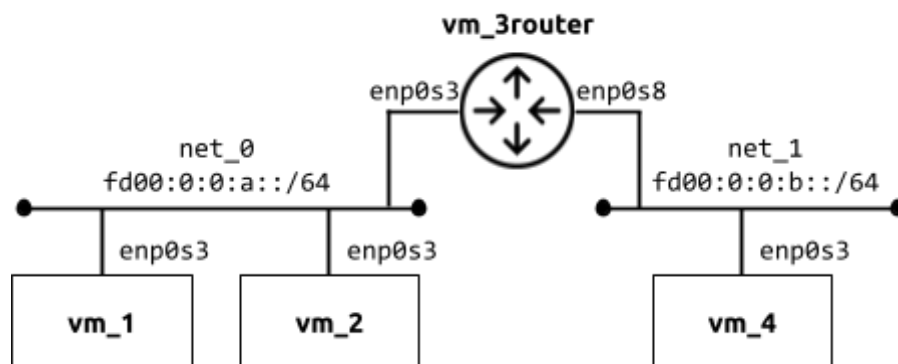
En esta práctica se estudian los aspectos básicos del protocolo IPv6, el manejo de los diferentes tipos de direcciones y mecanismos de configuración. Además se analizarán las características más importantes del protocolo ICMP versión 6.

Contenidos

- Preparación del entorno para la práctica
- Direcciones de enlace local
- Direcciones ULA
- Encaminamiento estático
- 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:

```
prefix vm
netprefix net
machine 1 0 0
machine 2 0 0
machine 3router 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 [vm_1, vm_2]. Activar la interfaz enp0s3 en vm_1 y vm_2. Comprobar las direcciones de enlace local que tienen asignadas con el comando ip.

Ejercicio 2 [vm_1, vm_2]. Comprobar la conectividad entre vm_1 y vm_2 con la orden ping6 (o ping -6). Cuando se usan direcciones de enlace local, y **sólo en ese caso**, es necesario especificar la 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.

Ejercicio 3 [vm_3router, vm_4]. Activar la interfaz enp0s3 de vm_4 y las interfaces enp0s3 y enp0s8 de vm_3router. Comprobar la conectividad entre vm_3router y vm_1, y entre vm_3router y vm_4 usando la dirección de enlace local.

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 las 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 [vm_1, vm_2]. Configurar vm_1 y vm_2 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.

Ejercicio 5 [vm_1, vm_2]. Comprobar la conectividad entre vm_1 y vm_2 con la orden ping6 usando la nueva dirección. Observar los mensajes intercambiados con Wireshark.

Ejercicio 6 [vm_3router, vm_4]. Configurar direcciones ULA en las dos interfaces de vm_3router (redes fd00:0:0:a::/64 y fd00:0:0:b::/64) y en el de vm_4 (red fd00:0:0:b::/64). Elegir el identificador de interfaz de forma que no coincida dentro de la misma red.

Ejercicio 7 [vm_3router]. Comprobar la conectividad entre vm_3router y vm_1, y entre vm_3router y vm_4 usando direcciones ULA. Comprobar además que vm_1 no puede alcanzar a vm_4.

Encaminamiento estático

Según la topología que hemos configurado en esta práctica, vm_3router 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 estableceremos manualmente en todas las máquinas.

Ejercicio 8 [vm_1, vm_3router]. Consultar las tablas de rutas en vm_1 y vm_3router con el comando ip route. Consultar la página de manual del comando para seleccionar las rutas IPv6.

Ejercicio 9 [vm_3router]. Para que vm_3router 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 [vm_1, vm_2, vm_4]. Finalmente, hay que configurar la tabla de rutas en las máquinas virtuales. Establecer vm_3router como encaminador por defecto con el comando ip route. Comprobar la conectividad entre vm_1 y vm_4 usando el comando ping6.

Ejercicio 11 [vm_1, vm_3router, vm_4]. Abrir Wireshark en vm_3router e iniciar dos capturas, una en cada interfaz de red. Borrar la tabla de vecinos en vm_1 y vm_3router (con ip neigh flush dev

<interfaz>). Usar la orden ping6 entre vm_1 y vm_4. Completar la siguiente tabla con todos los mensajes hasta el primer ICMP Echo Reply:

Red fd00:0:0:a::/64 - vm_3router (enp0s3)

MAC Origen	MAC Destino	IPv6 Origen	IPv6 Destino	ICMPv6 Tipo

NOTA: Añadir tantas filas como sea necesario

Red fd00:0:0:b::/64 - vm_3router (enp0s8)

MAC Origen	MAC Destino	IPv6 Origen	IPv6 Destino	ICMPv6 Tipo

NOTA: Añadir tantas filas como sea necesario

Autoconfiguración. Anuncio de prefijos

El protocolo de descubrimiento de vecinos se usa también para la autoconfiguración de las interfaces de red. Cuando se activa una 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 12 [vm_1, vm_2, vm_4]. Eliminar las direcciones ULA de las interfaces desactivándolas con `ip link`.

Ejercicio 13 [vm_3router]. Configurar el servicio zebra (del paquete de encaminamiento FRRouting) para que el encaminador anuncie prefijos. Para ello, editar el archivo `/etc/frr/frr.conf` e incluir la información de los prefijos para las dos redes. Cada entrada será de la forma:

```
interface enp0s3
  no ipv6 nd suppress-ra
  ipv6 nd prefix fd00:0:0:a::/64
```

Finalmente, arrancar el servicio con el comando `systemctl start frr`.

Ejercicio 14 [vm_4]. Comprobar la autoconfiguración de la interfaz `enp0s3` de `vm_4`, volviéndolo a activar y consultando la dirección asignada.

Ejercicio 15 [vm_1, vm_2]. Estudiar los mensajes del protocolo de descubrimiento de vecinos:

- Activar la interfaz `enp0s3` en `vm_2`, comprobar que está configurado correctamente e iniciar una captura de paquetes con Wireshark.
- Activar la interfaz `enp0s3` en `vm_1` 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`.

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, se puede usar el proyecto de código libre *Router Advertisement Daemon*, `radvd`.

Ejercicio 16 [vm_1]. 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 `vm_1` con `sysctl net.ipv6.conf.enp0s3.use_tempaddr=2` y repetir el proceso de autoconfiguración.

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 17. 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 vecinos.
- Destino inalcanzable - Sin ruta al destino (Code: 0).
- Destino inalcanzable - Dirección inalcanzable (Code: 3)
- Destino inalcanzable - Puerto inalcanzable (Code: 4)