Práctica 3 Configuración y administración de redes

CURSO 2020-2021

OBJETIVOS

El objetivo de esta práctica es que el alumno se familiarice con la configuración de redes y, más concretamente, en proporcionar conectividad entre diferentes equipos conectados a una red. Algunas de las tareas que se van a realizar son: configuración de direccionamiento IP de un host, configuración de rutas mediante protocolos de encaminamiento dinámico y de manera estática, uso de IPv6 y configuración de túneles.

CONCEPTOS INTRODUCTORIOS

IPv6

El motivo por el que surge IPv6 en el seno del IETF es la necesidad de crear un nuevo protocolo debido a la evidencia de la falta de direcciones IPv4, que en un primer momento se llamó IPng (*IP next generation*). IPv4 tiene un espacio de direcciones de 32 bits, es decir 2^{32} (4.294.967.296), en cambio IPv6 nos ofrece un espacio de direcciones de 2^{128} (340.282.366.920.938.463.463.374.607.431.768.211.456).

Una dirección IPv6 está compuesta por 128 bits, que se representa en 8 grupos de 4 valores hexadecimales (16 bits), siguiendo el siguiente esquema:

- Si *x* son 16 bits en hexadecimal (4 valores), la representación sería *x:x:x:x:x:x:x:x*.
- Los ceros a la izquierda de cada grupo de 16 bits no se consideran "*0800=800*"; "*0001=1*".
- Una serie de grupos de 16 bits de ceros, se puede sustituir por ::, pero solo una vez en una dirección 0:0:0 = ::.

Ejemplos de direcciones IPv6 son: fff:400D:123:cafe:98dd:1:2:aaaa, 1000:0:0:0:0:0:8574:1 = 1000::8574:1.

Otra forma de representar direcciones IPv6 cuando se está en entornos mixtos IPv4 e IPv6 es la representación *x:x:x:x:x:x:d.d.d.d*, donde *x* representa un valor hexadecimal de 16 bits, y *d* representa un valor decimal, la idea es dejar los últimos 32 bits en formato igual a IPv4.

La representación de prefijos de red, se realiza de la siguiente forma: "*dirección–IPv6/longitud-prefijo*", donde *dirección-IPv6* es una dirección IP en cualquiera de las notaciones válidas y *longitud-prefijo* es un valor decimal que indica el número de bits a uno continuos desde la izquierda.

iproute2 (ip)

IP es una utilidad perteneciente al paquete *iproute2* que permite controlar los parámetros de configuración IPv4 e IPv6 de un sistema *Linux*[1]. A continuación se realiza una descripción de los comandos básicos de configuración de encaminamiento de *Linux*. Para obtener una descripción completa, se aconseja acudir a la descripción del comando utilizando "*man ip*".

Hay que tener en cuenta que todos los comandos que se van a describir en esta sección introducen cambios provisionales que desaparecen al reiniciar la máquina. Para que la configuración se mantenga, debe introducirse en los ficheros de configuración de red que son leídos al arrancar la máquina.

Configuración de interfaces

Para configurar una interfaz de red (MTU, tamaño de la cola, dirección de *broadcast*,…) se emplea el comando:

```
# ip link set DEVICE { up | down | arp { on | off } | promisc { on | off } | allmulti { on | off } | dynamic { on | off } | multicast { on | off } | txqueuelen PACKETS | name NEWNAME | address LLADDR | broadcast LLADDR | mtu MTU }
```

Donde DEVICE representa el nombre de la interfaz de red (ej. *eth0*, *eth1*,…). Para mostrar el valor de estos parámetros, el comando a ejecutar es:

```
# ip link show
```

[1] Aunque muchas de las funciones se pueden realizar con el comando *ifconfig*, no se recomienda el uso de este último por encontrarse en desuso e incluso no encontrarse instalado en algunas distribuciones recientes de Linux.

Configuración de direcciones

Para dotar de una dirección IP y una máscara de red a un determinado interfaz de red se emplea el comando:

```
# ip addr { add | del } IFADDR dev STRING
```

donde IFADDR representa la dirección IP y la máscara de red asignada a la interfaz STRING. Por ejemplo:

```
# ip addr add 172.16.21.1/24 dev eth0
# ip addr add 2001:07F9:0400:0001:0002::012/126 dev eth0
```

Para mostrar la asignación de direcciones a las distintas interfaces de red puede emplear el comando:

```
# ip addr show
```

Configuración de rutas

La tabla de encaminamiento puede ser estática o dinámica dependiendo de los protocolos de encaminamiento que se tengan en la red. En redes sencillas, es habitual emplear tablas estáticas, las cuales pueden manipularse mediante el comando:

```
# ip route { add | del |  replace } NET [ via ADDRESS ] [ dev STRING ]
```

Donde NET representa la red destino que se quiere encaminar, ADDRESS indica el próximo salto y STRING la interfaz de salida.

Sin embargo, en redes más complejas es conveniente tener un protocolo dinámico que actualice automáticamente las tablas encaminamiento cuando exista alguna variación en la red. En este caso, hay que configurar el programa que contiene el protocolo de encaminamiento, siendo, por ejemplo, *Quagga*.

De manera general, para mostrar el contenido de la tabla de rutas, se emplea el comando:

```
# ip route show
```

Para conocer la tabla de rutas de IPv6, puede emplear la opción "-6", quedando:

```
# ip -6 route show
```

Configuración de túneles

Los túneles son interfaces IP que realizan el encapsulado de tráfico de usuario para su transporte entre un origen y un destino. En la actualidad, los túneles IP se utilizan para el transporte de tráfico con direccionamiento privado a través de una red con direccionamiento público (VPN) y para el transporte de IPv6 sobre infraestructuras IPv4. Existen distintos tipos de túneles IP, siendo los principales GRE, IPIP y SIT (que encapsula tráfico IPv6 sobre IPv4). El comando para la definición de túneles en *Linux* es:

```
# ip tunnel { add | change | del | show } [ NAME ] [ mode { ipip | gre | sit } ] [ remote
ADDR ] [local ADDR ] [ ttl TTL ] [ tos TOS ] [ dev PHYS_DEV ]
```

Para mostrar los túneles definidos en un sistema, el comando a ejecutar es

```
# ip tunnel show
```

Comandos para comprobar rutas

El comando más conocido que sirve para verificar si un host está accesible (y con ello verificar la ruta) es el comando *ping*, cuya sintaxis simplificada es:

```
# ping [-c número] destino
```

Al ejecutar el comando estará constantemente enviando mensajes ICMP al destino y mostrando por pantalla el tiempo que tarda en llegar la respuesta, hasta que sea interrumpido con <*Control*+*C*>. Con la opción *-c* se puede limitar el número de mensajes enviados.

Otro comando similar a ping es *traceroute*, pero con algunas mejoras, ya que permite ver la ruta que sigue cada paquete hasta llegar al destino. Los paquetes puede que no sigan siempre la misma ruta para alcanzar un destino, aunque lo más habitual es que sí que lo hagan. Una opción interesante de *traceroute* es *–n*, la cual evita que se traduzcan las direcciones IP de los *routers* a sus nombres, acelerando la ejecución del comando.

```
# traceroute –n dirección_destino
```

Análogamente a IPv4, los comandos para comprobar rutas en IPv6 son *ping6* y *traceroute6*, que tienen una sintaxis igual a la de los comandos *ping* y *traceroute*. Para obtener más información de estos comandos, se recomienda al alumno acudir a las páginas *man* de *ping6* y *traceroute6*.

Quagga

Uno de los objetivos de la práctica es que el alumno se familiarice con la configuración de encaminadores *Linux* y los protocolos de encaminamiento dinámico. Para la configuración de protocolos de encaminamiento dinámicos se emplea el software *Quagga*. *Quagga* es un software que proporciona protocolos de encaminamiento de servicios IPv4 como OSPF, RIP y BGP, y de encaminamiento IPv6 como RIPng y OSFPv6. Para la instalación de Quagga en Linux, haga uso del comando:

```
# apt-get install quagga
```

Quagga tiene un funcionamiento modular de manera que para cada protocolo de encaminamiento dispone de un demonio de configuración propio, lo que hace relativamente sencillo añadir nuevos protocolos de encaminamiento. Existen dos formas de configuración de cada uno de los demonios de *Quagga*: mediante un fichero de texto que se carga cuando se inicia el sistema y mediante el uso de *telnet*. En esta práctica se ha optado por la primera opción, pudiendo encontrar más información acerca del contenido que deben tener dichos ficheros en http://www.nongnu.org/quagga/.

Los ficheros relativos a *Quagga* se encuentran en dos directorios diferentes:

- ***/etc/quagga***: este directorio, que probablemente esté vacío de inicio, dispone de los ficheros de configuración de los diferentes demonios de *Quagga*. La mayor parte de dichos demonios son el nombre del protocolo de enrutamiento finalizado con una *d*. Así, en dicho directorio estarán, por ejemplo, los ficheros *ripd.conf*, *ospfd.conf*, *ripngd.conf*, *bgpd.conf* para contener la configuración de los protocolos RIP, OSPF RIPng y BGP, respectivamente (no indica que tengan que estar operativos todos los protocolos simultáneamente).
  Es importante destacar el demonio Zebra (con archivo de configuración *zebra.conf*), que es el demonio de encaminamiento IP gestionado por *Quagga*, proporcionando las actualizaciones de la tabla de encaminamiento y redistribución de rutas entre los diferentes protocolos de encaminamiento. Así, el demonio *zebra* deberá inicializarse para que puedan funcionar el resto de protocolos de encaminamiento. Para facilitarle la configuración de dicho fichero, para esta práctica es suficiente con una enumeración de interfaces (que puede conocer con `ip link show`), por ejemplo:

  ```
  interface eth0

  interface lo
  ```

  Como ejemplo, si desea ejecutar el protocolo OSPF, debe configurar los ficheros *zebra.conf* y *ospfd.conf*.

- ***/etc/init.d***: este directorio es desde donde se activan, paran o reinician los diferentes demonios de los protocolos de encaminamiento y Zebra. Por ejemplo, para los demonios Zebra y OSPF, use los comandos:

  ```
  # /etc/init.d/zebra {start | stop | restart }

  # /etc/init.d/ospfd {start | stop | restart }
  ```

  Es decir, para cada demonio debe emplear el nombre del fichero de configuración (sin la extensión `.conf`) indicado anteriormente. Por ejemplo, si desea ejecutar el protocolo OSPF, debe configurar los ficheros *zebra.conf* y *ospfd.conf*.
  Es importante destacar que, si se realizan cambios en los ficheros de configuración, es necesario reiniciar el correspondiente demonio para que los cambios surtan efecto.

Aunque el funcionamiento de las versiones actuales de *Quagga* es el mostrado arriba, todavía existe un buen número de equipos Linux con versiones con un funcionamiento diferente, por lo que conviene explicarlo brevemente. En versiones más antiguas, en el directorio */etc/quagga*, además de los ficheros de configuración ya expuestos, debe existir un fichero de nombre *daemons* que contiene el nombre de los diferentes demonios de Quagga (incluido Zebra), de manera que Quagga arrancará aquellos demonios que estén con el estado *yes*. Como ejemplo, si desea que funcione el protocolo OSPF debe aparecer la línea "*ospfd=yes*" en lugar de "*ospfd=no*", que es como viene por defecto. Por otro lado, la activación de los demonios se realiza de manera

general mediante *Quagga* (aunque pueden emplearse más argumentos para tratar con demonios de manera individual) del siguiente modo:

```
# /etc/init.d/quagga {start | stop | restart }
```

REALIZACIÓN DE LA PRÁCTICA

En las siguientes secciones se explican las tareas que se le solicita realizar. En la memoria que debe entregar conteste a todas las cuestiones que se indican al final de cada tarea, incluyendo un análisis crítico de las tareas que realice.

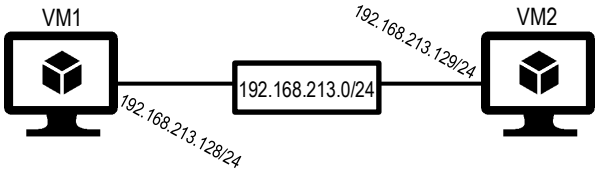
Preparación de la práctica

Para la realización de la práctica debe descargar el siguiente software de Internet[2] (puede emplear otro gestor de máquinas virtuales o distribución de Linux, si bien son estas las que se emplean como referencia en este guion):

- VMware Workstation Player.
- Imagen iso de Ubuntu Desktop.

Una vez descargadas, siga los siguientes pasos:

1. Instale VMware Workstation Player y cree una máquina virtual (VM) con la imagen de Ubuntu Desktop. Emplee el modo NAT para dicha VM. Además, identifique el directorio donde se almacena dicha VM.
2. Instale *Quagga* y Wireshark en la VM.
3. Haga una copia del directorio de la VM creada y abra VMware Workstation Player para abrir una nueva VM, seleccionando la copia que acaba de hacer. Al abrirla, indique a VMware que se trata de una copia (no que se ha movido la VM anterior).
4. Arranque ambas VMs usando VMware.
5. Es importante conocer que, si ambas VMs se han configurado del mismo modo (por ejemplo modo NAT), ambas deben tener conectividad entre sí. De hecho, ambas VMs pertenecen a la misma subred que ha creado VMware en el host físico donde se están ejecutando (192.168.213.0/24 en el ejemplo). En la siguiente figura se observar un ejemplo, donde se puede ver que la infraestructura montada es equivalente a dos *hosts* (virtuales en este caso) conectados a una misma red.



Compruebe que ambas VMs se encuentran en la misma red ejecutando en cada una de ellas:

```
# ip addr show
```

Una vez ejecutadas dichas VMs, compruebe la conectividad entre ellas mediante:

```
VM1> ping 192.168.213.129
```

NOTA: aunque no se indique por claridad, muchos de los comandos requieren ejecutarse con `sudo`.

Tarea 1: Configuración del direccionamiento IP

En esta primera parte de la práctica, únicamente se pide que configure una de las VMs que está ejecutando para asignarle la dirección IP 10.10.5.5, con una máscara de 32 bits en la interfaz *loopback* y la dirección IPv6 3ff::5 con una máscara de 122 bits en la interfaz que conecta las VMs entre sí. Una vez haya realizado estos pasos, elimine dichas asignaciones para volver al estado inicial. NOTA: no elimine las direcciones IP ya existentes en su equipo.

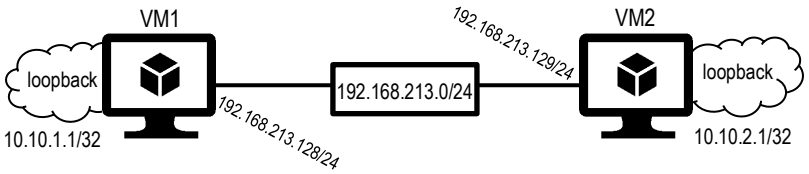
Conteste en la memoria a la siguiente pregunta:

[2] En el momento de la realización de este guion la versión de VMware Workstation Player es la 15 y de Ubuntu Desktop la 20.04.

☐ Indique los comandos necesarios para añadir dichas direcciones IP y para eliminarlas.

Tarea 2: Configuración de rutas estáticas

Para dotar de conectividad a un conjunto de redes que no están directamente conectadas (en cuyo caso Linux lo hace automáticamente) se va a comenzar haciendo uso de la herramienta *ip* para instalar rutas de manera manual. Puesto que la maqueta compuesta de dos VMs de la que se dispone no tiene redes que no se encuentren directamente conectadas, se va a emplear la interfaz de *loopback* de dichas VMs para definir en cada una de ellas una red diferente, las cuales no están directamente conectadas. Así, el formato de la red a definir en la interfaz *loopback* de VM1 será 10.10.1.1/32 (y 10.10.2.1/32 en VM2). El objetivo es configurar rutas estáticas en ambas VMs, las cuales, actuando como *routers*, permitirán tener conectividad entre la red 10.10.1.1/32 y 10.10.2.1/32 a través de la red que conecta ambas VMs (192.168.213.0/24 en el ejemplo), tal y como muestra la siguiente figura:



Conteste en la memoria a las siguientes preguntas:

☐ Indique y explique brevemente los comandos introducidos para realizar este apartado.
☐ Explique cómo ha comprobado la conectividad y muestre el resultado de los comandos que demuestran dicha conectividad.
☐ Indique los comandos necesarios para eliminar las rutas creadas y muestre cómo dichas rutas se han eliminado.

Tarea 3: Configuración de protocolos de encaminamiento

Esta tarea tiene el mismo objetivo que la tarea 2, pero ahora el encaminamiento no va a realizarse de manera estática sino dinámica, para lo cual va a hacerse uso de protocolos de encaminamiento. Más concretamente, se va a emplear el protocolo OSPF para tener conectividad entre la red 10.10.1.1/32 y 10.10.2.1/32, definidas en las interfaces de *loopback* de la VM1 y VM2, respectivamente.

Para la configuración de OSPF, se recomienda la consulta de *http://www.nongnu.org/quagga/*, aunque aquí se dan algunas indicaciones:

- Para que las rutas anunciadas por OSPF se compartan entre VM1 y VM2, debe emplear en ambas VMs la misma área OSPF, por ejemplo, 0.0.0.1.
- Debe hacer que OSPF redistribuya las rutas que tiene conectado.

Para saber si OSPF está funcionando correctamente, compruebe la conectividad entre las subredes conectadas a la interfaz *loopback* de los hosts.

Conteste en la memoria a las siguientes preguntas:

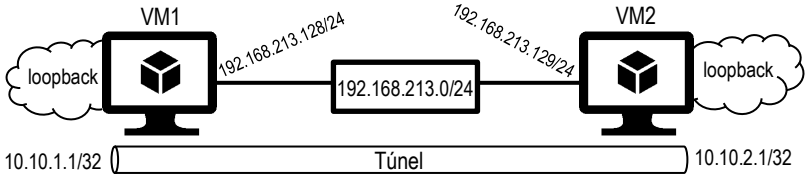
☐ Explique detalladamente los pasos que siga para realizar este apartado, así como el contenido de los ficheros de configuración que haya creado o modificado.
☐ Explique cómo ha comprobado la conectividad y muestre el resultado de los comandos que demuestran dicha conectividad.
☐ ¿Cómo sabe que la ruta se ha creado de manera dinámica?
☐ Empleando Wireshark, analice los paquetes OSPF observando cual es el intervalo de tiempo entre dos paquetes *hello* consecutivos. Cambie dicho intervalo de tiempo e indique los pasos seguidos, así como muestre mediante capturas de Wireshark que dicho intervalo ha sido cambiado con éxito.

Tarea 4: Configuración de túneles, protocolos de encaminamiento e IPv6

Partiendo del resultado de la tarea 3, donde se dispone de tres redes y se emplea OSPF para dotar de conectividad a las redes que no están directamente conectadas (las definidas en las interfaces *loopback* de las VMs), en esta tarea se va a configurar un túnel IPv6 sobre IPv4 que interconectará dos encaminadores, utilizando OSPF como protocolo de encaminamiento de IPv4 y RIPng como protocolo de encaminamiento de IPv6. Para la realización de esta tarea, se requieren dos subtareas:

Tarea 4.1: Definición de túnel IPv6 sobre IPv4

En esta parte únicamente ha de crear un túnel IPv6 sobre IPv4, tal y como muestra la siguiente figura. Para VM1, el origen del túnel es la dirección IPv4 empleada en la interfaz *loopback* (10.10.1.1) y el fin del túnel es la dirección IPv4 de la interfaz de *loopback* de VM2 (10.10.2.1). Para VM2, el origen será 10.10.2.1 y el final 10.10.1.1. Una vez creado dicho túnel, recuerde que debe activarlo (ponerlo en estado UP) para que pueda funcionar.

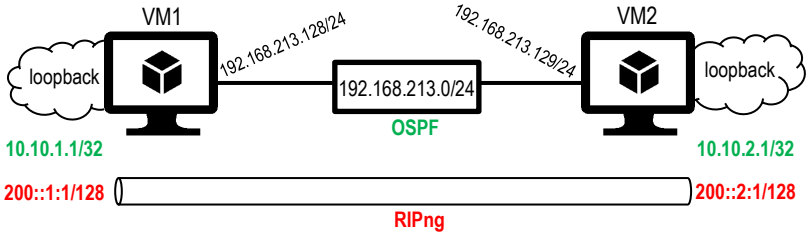


Conteste en la memoria a las siguientes preguntas:

- ☐ Indique y explique brevemente los comandos introducidos para realizar este apartado.
- ☐ Indique el comando que debe emplear para comprobar que se ha creado el túnel y muestre su resultado.
- ☐ ¿Debe emplearse el mismo nombre de túnel en VM1 y en VM2? Justifique su respuesta.

Tarea 4.2: Conectividad IPv6 mediante RIPng

Puesto que desea tener conectividad IPv6 extremo a extremo, debe definir, en primer lugar, la red IPv6 a la que cada equipo está conectado, empleándose de nuevo las interfaces de *loopback* de las VMs y siendo esta 200::1:1/128 para VM1 y 200::2:1/128 para VM2. El objetivo será, por tanto, que ambas VMs tengan conectividad con la red IPv6 de la otra VM empleando el túnel definido en la tarea 4.1 y empleando el protocolo de encaminamiento RIPng, tal y como muestra la siguiente figura:



Una vez creado el túnel en la tarea 4.1, como último paso para la creación de la infraestructura de red especificada en la figura anterior, debe configurar el protocolo de encaminamiento RIPng para encaminar datagramas IPv6 a través del túnel. Es decir, si en la tarea 3 se empleaba OSPF para distribuir las rutas 10.10.1.1/32 y 10.10.2.1/32 a través de la red 192.168.213.0/24, ahora se emplea RIPng para distribuir las rutas a las redes 200::1:1/128 y 200::2:1/128 a través del túnel IPv6 sobre IPv4.

Para realizar esta tarea, tenga en cuenta las siguientes consideraciones:

- Para que funcione RIPng sobre el túnel, este debe tener activada la capacidad *multicast*.
- Debe configurar RIPng de manera que se genere una ruta por defecto y, además, debe hacer que RIPng redistribuya las rutas que tiene conectado.

Conteste en la memoria a las siguientes preguntas:

- ☐ Explique detalladamente los pasos que siga para realizar este apartado, así como el contenido de los ficheros de configuración que haya modificado.
- ☐ Explique cómo ha comprobado la conectividad y muestre el resultado de los comandos que demuestran dicha conectividad.
- ☐ Desde VM1 haga un *ping* a la dirección 200::2:1 y, capturando uno de dichos *pings* (en sentido de ida) con Wireshark, rellene la siguiente tabla:

Pila de protocolos completa	
Dirección IP origen IPv4	
Dirección IP destino IPv4	
Dirección IP origen IPv6	
Dirección IP destino IPv6	
TTL	
Hop limit	
TOS (o DSCP)	
Traffic Class	

- ☐ ¿Qué le ocurre a la conectividad IPv6 entre las redes 200::1:1/128 y 200::2:1/128 si se detiene el demonio OSPF en VM1? Justifique su respuesta.

PLAZO PARA LA REALIZACIÓN DE LA PRÁCTICA Y EVALUACIÓN

Esta práctica debe realizarse por parejas por parte de los alumnos (se podrá realizar de manera individual cuando esté debidamente justificado y autorizado por parte del profesor). La evaluación de la práctica se hará mediante la presentación de una memoria de la práctica donde debe constar todo el trabajo realizado por el alumno y donde se debe responder a las diferentes cuestiones que se han planteado en este guion. Dicha memoria debe enviarse mediante el Aula Virtual (Blackboard LS) de la asignatura con fecha límite el **2 de junio de 2021** inclusive. El envío debe realizarlo uno de los integrantes de la pareja, si bien debe incluir el nombre de ambos miembros.

La honestidad en el trabajo personal o de grupo debe ser una característica fundamental de un estudiante universitario. Por este motivo, se confía en que las memorias, programas,… que se presenten en la asignatura solo serán realizados por sus personas responsables y firmantes. La ruptura de esta confianza implicará que los alumnos y alumnas afectados no puedan superar la asignatura.

BIBLIOGRAFÍA

[1] Páginas de manual de *ip*, *ping*…
[2] Quagga routing suite http://www.nongnu.org/quagga/
[3] Wireshark *http://www.wireshark.org*
[4] RFC 2080. RIPng for IPv6
[5] RFC 2328 (rfc2328) - OSPF Versión 2