

Índice

1. Configuración del fichero net.conf	3
2. Preparación del entorno	3
3. Finalización del entorno	3
4. Iniciar sesión en una UML	3
5. Acceder al shell de Quagga	3
6. Acceder a la consola para introducir comandos de configuración	3
7. Acceder a una interfaz	4
8. Añadir una IP a la interfaz	4
9. Activar retransmisión de paquetes	4
10. Anuncio de prefijo en IPv6	4
11. Guardar la configuración establecida	4
12. Zebra	4
12.1. Interface Commands	4
12.1.1. Standard Commands	4
12.2. Zebra Terminal Mode Commands	4
13. RIP	5
13.1. Start RIPd	5
13.2. Configuration	5
13.3. Show RIP Information	5
14. RIPng	5
14.1. Start RIPngd	5
14.2. Configuration	5
14.3. Show RIPng Information	6
15. OSPFv2	6
15.1. Configuration	6
15.2. Router	6
15.3. Area	6
15.4. Showing OSPF Information	6

16.OSPFv3	7
16.1. Configuration	7
16.2. Router	7
16.3. Interface	7
16.4. Showing OSPF Information	7
17.BGP	8
17.1. Start BGPd	8
17.2. BGP Router	8
17.3. BGP network	8
17.4. Inyección de protocolos	8
17.5. Definición de vecinos	8
17.6. Filtrado de rutas	9
17.7. AS_PATH: Regex	9
17.8. Muestra Rutas BGP por el AS_PATH	9
17.9. AS Path Access List	10
17.10 Verifica las rutas anunciadas por BGP	10
18.Configurar los demonios de Quagga	10
19.Reiniciar y comprobar los demonios activos de Quagga	10
20.Atajos útiles de líneas de comandos	10

1. Configuración del fichero net.conf

```
redes@RED:~$ (nano|vim.tiny) net.conf # Tu editor favorito dentro de la VM
# NO deben repetirse las umlX.Y o de lo contrario,
# dará un error al lanzar el escenario
defsw br12 uml1.0 uml2.0 # configura la conexión entre dos interfaces
defsw net1 uml1.1 # configura la conexión para una sola interfaz
defsw br345 uml3.0 uml4.1 uml5.2 # configura la conexión para tres interfaces
```

2. Preparación del entorno

```
redes@RED:~$ sudo ifovsdel # borra las configuraciones previamente existentes
redes@RED:~$ sudo ifovsparse net.conf # configura las interfaces para lanzar el entorno
redes@RED:~$ mkdir uml{1..N} # crea un rango de directorios desde uml1 a umlN
redes@RED:~$ lanza {1..N} # crea las uml declaradas anteriormente
```

3. Finalización del entorno

```
# Ejecuta un ctrl+alt+del en cada una de las UML (forma suave de matar las UML)
redes@RED:~$ for i in {1..N} ; do uml\_mconsole uml$i cad; done
# Mata todos los subprocesos linux de golpe (forma dura de matar las UML)
# Es el último recurso a utilizar en caso de que el anterior comando no funcione
redes@RED:~$ pkill linux
```

4. Iniciar sesión en una UML

```
uml1 login: root # entramos siempre como root
root@uml1# # A partir de aquí, estamos dentro de una sesión bash en la UML1
```

5. Acceder al shell de Quagga

```
root@uml1# vtysh
uml1# # Aquí ya estamos dentro de una sesión de Quagga
```

6. Acceder a la consola para introducir comandos de configuración

```
uml1# # Lo que está entre corchetes significa que es opcional
uml1# conf[igure] term[inal]
uml1(config)# ! Estamos dentro de la configuración de la UML
```

7. Acceder a una interfaz

```
uml1(config)# int[erface] eth0
uml1(config-if)# # Estamos dentro de la interfaz eth0
```

8. Añadir una IP a la interfaz

```
uml1(config-if)# ip address 10.0.0.1/24
uml1(config-if)# ipv6 address 2001:db8::50:10/64
uml1(config-if)# no shutdown # IMPORTANTE para que la interfaz quede levantada
```

9. Activar retransmisión de paquetes

```
uml1(config)# # Lo que se le dice a la UML donde se ejecuta el siguiente
uml1(config)# # comando es que actúe como router (encaminador)
uml1(config)# ip[v6] forwarding
```

10. Anuncio de prefijo en IPv6

```
uml1(config-if)# ! Los hosts a los que hayan que hacerles el anuncio del prefijo
uml1(config-if)# ! solo hay que hacerles un `no shutdown' en la interfaz asociada
uml1(config-if)# ipv6 nd prefix 2001:db8:101:1::/64
uml1(config-if)# no ipv6 nd suppress-ra
```

11. Guardar la configuración establecida

```
uml1(config)# write ! Guarda toda la configuración
```

12. Zebra

12.1. Interface Commands

12.1.1. Standard Commands

```
!!! Estos comandos es necesario estar dentro de la interfaz
! Apaga o levanta la interfaz
[no] shutdown
! Añade una IP v4 o v6 a una interfaz
[no] ip[v6] a[ddress] ADDRESS/PREFIX
```

12.2. Zebra Terminal Mode Commands

```
! Muestra las rutas actuales que almacena zebra en su base de datos
show ip[v6] route
```

```
! Muestra información sobre la interfaz  
show interface [INTERFACE] ! ej: eth0
```

13. RIP

13.1. Start RIPd

```
# Activar el demonio ripd en el fichero /etc/quagga/daemons  
~$ systemctl restart quagga  
~$ systemctl status quagga
```

13.2. Configuration

```
! Habilita o deshabilita RIP  
[no] router rip  
! Fija o no la interfaz RIP habilitada por ifname  
[no] network ifname ! ej.: network eth0  
! Especifica o no un vecino RIP  
[no] neighbor a.b.c.d  
! Controla split-horizon en la interfaz. Por defecto activado  
[no] ip split-horizon
```

13.3. Show RIP Information

```
! Muestra rutas RIP  
show ip rip  
! Muestra estado actual de RIP  
show ip rip status
```

14. RIPng

14.1. Start RIPngd

```
# Activar el demonio ripngd en el fichero /etc/quagga/daemons  
~$ systemctl restart quagga  
~$ systemctl status quagga
```

14.2. Configuration

```
! Habilita o deshabilita RIPng  
[no] router ripng  
! Fija o no la interfaz RIPng habilitada por ifname  
[no] network ifname ! ej.: network eth0
```

14.3. Show RIPng Information

```
! Muestra rutas RIPng  
show ip ripng
```

15. OSPFv2

15.1. Configuration

```
# Activar el demonio ospfd en el fichero /etc/quagga/daemons  
~$ systemctl restart quagga  
~$ systemctl status quagga
```

15.2. Router

```
! Activa o desactiva el proceso OSPF  
[no] router ospf  
! Fija o quit el router-id del proceso OSPF  
[no] ospf router-id a.b.c.d  
! Solo se anuncia la interfaz como link stub en router LSA  
! Hay que ponerlo en las redes de usuario  
passive-interface ifname  
! Habilita o deshabilita stub router advertisement support  
[no] max-metric router-lsa [on-startup|on-shutdown|administrative]  
! Especifica o no las interfaces habilitadas en OSPF  
[no] network a.b.c.d/m area <0-4294967295> ! también puede ser en el formato a.b.c.d
```

15.3. Area

```
! Agrega o desagrega rutas inter-área (solo válido para ABR)  
[no] area a.b.c.d range a.b.c.d/m ! Un ejemplo en la práctica de OSPFv2  
! Define o elimina un área stub  
[no] area a.b.c.d sub [no-summary]
```

15.4. Showing OSPF Information

```
! Muestra info de proposito general de OSPF  
show ip ospf  
! Muestra estado y configuracion de la interfaz especifica OSPF (todas si  
! no se especifica alguna)  
show ip ospf interface [INTERFACE]  
! Muestra info de la BBDD OSPF  
show ip ospf database  
! Muestra la tabla de rutas OSPF  
show ip ospf route
```

16. OSPFv3

16.1. Configuration

```
# Activar el demonio ospf6d en el fichero /etc/quagga/daemons
~$ systemctl restart quagga
~$ systemctl status quagga
```

16.2. Router

```
! Acceder a la configuracion del router
router ospf6
! Fijar un router-id
router-id a.b.c.d ! ej: 0.0.0.1
! Bindea una interfaz a un area y comienza el envio de paquetes
interface ifname area a.b.c.d ! ej: interface eth0 area 0.0.0.1
```

16.3. Interface

```
!!! Estos comandos es necesario estar dentro de la interfaz

! Establece coste de la interfaz. El valor por defecto depende
! del ancho de banda de la interfaz y del coste del ancho de
! banda de referencia
ipv6 ospf6 cost COST ! ej: 2
! Establece coste del intervalo HELLO de la interfaz
ipv6 ospf6 hello-interval HELLOINTERVAL ! ej: 60
! Establece intervalo Router Dead en la interfaz
ipv6 ospf6 dead-interval DEADINTERVAL ! ej: 240
! Establece intervalo de retransmisi3n de la interfaz
ipv6 ospf6 retransmit-interval RETRANSMITINTERVAL ! ej: 5
! Establece Router Priority de la interfaz
ipv6 ospf6 priority PRIORITY ! ej: 5
! Establece Inf-Trans-Delay de la interfaz
ipv6 ospf6 transmit-delay TRANSMITDELAY ! ej: 1
```

16.4. Showing OSPF Information

```
! Muestra info acerca de OSPFv3 de la UML
show ipv6 ospf6 [INSTANCE_ID]
! Muestra el resumen de la base de datos LSA
show ipv6 ospf6 database
! Muestra la configuracion de la interfaz OSPF
show ipv6 ospf6 interface
! Muestra el estado y el backup del vecino
show ipv6 ospf6 neighbor
```

```
! Muestra la request-list del vecino
show ipv6 ospf6 request-list A.B.C.D
! Muestra la tabla de rutas interna
show ipv6 route ospf6
```

17. BGP

17.1. Start BGPd

```
# Activar el demonio bgpd en el fichero /etc/quagga/daemons
~$ systemctl restart quagga
~$ systemctl status quagga
```

17.2. BGP Router

```
! [Des]habilita el proceso del protocolo BGP con el ASN especificado
[no] router bgp asn ! ej: 65512
! Especifica el router-id (Identificador BGP)
bgp router-id A.B.C.D ! ej: 10.0.0.1
```

17.3. BGP network

```
! Añade/quita un anuncio BGP explícito
[no] network A.B.C.D/M ! ej: 10.0.0.0/8
! Route aggregation
!! Añade/quita un agregado de ruta
[no] aggregate-address A.B.C.D/M ! ej: 192.168.0.0/16
```

17.4. Inyección de protocolos

```
! Redistribuye RIP / OSPF al proceso BGP
redistribute rip | ospf
```

17.5. Definición de vecinos

```
! Crea/elimina un nuevo vecino cuyo ``remote-as'' es asn.
! peer puede ser una dirección IPv4 o IPv6
[no] neighbor peer remote-as asn ! ej: neighbor 10.0.0.2 remote-as 65513
! Especifica la interfaz cuando el vecino está conectado por
! una dirección IPv6 de enlace local
neighbor peer interface ifname ! ej: neighbor fe80::ff:fe00:1f0 interface eth0
```


17.6. Filtrado de rutas

The **distribute-list** and **prefix-list** perform route filtering based on IP network addresses and netmasks of routes being advertised. The **distribute-list** refers to an ACL to match the individual networks and netmasks, while **prefix-list** refers to a prefix list to do this matching. In fact, the use of **distribute-list** and **prefix-list** for a particular BGP neighbor in a particular direction (in or out) is mutually exclusive, because they both accomplish the very same goal, just using a different route selection/filtering mechanism (an ACL vs. a prefix list). It is generally better to use prefix lists instead of ACLs - they are much more cleaner and more comprehensible, optimized to match networks/netmasks and subnets thereof.

The **filter-list** performs route filtering based on the contents of the AS_PATH attribute - the sequence and values of autonomous system numbers. To do this, you would configure an as-path ACL that contains one or more regular expressions matching the particular sequence of ASNs in the AS_PATH attribute, and apply it to a neighbor and a particular direction with the **filter-list** command. With a **filter-list**, you do not perform route matching/filtering based on IP addresses and netmasks.

```
! Filtrado de prefijos
! ej: neighbor fe80::ff:fe00:1f0 prefix-list peer-out out
neighbor peer prefix-list name [in|out]
neighbor peer distribute-list name [in|out]
! ej: neighbor 10.0.0.1 route-map internal-rtm out
neighbor peer route-map name [in|out]
```

17.7. AS_PATH: Regex

- . Cualquier carácter.
- * 0 o más apariciones del patrón.
- + 1 o más apariciones del patrón.
- ? 0 ó 1 apariciones del patrón.
- ^ Comienzo de línea.
- \$ Final de línea.
- _ Concuerda con el espacio y la coma, así como con el delimitador de conjunto de ASN { y }, y con el delimitador de confederación (y). También concuerda con el principio y final de línea.

17.8. Muestra Rutas BGP por el AS_PATH

```
! muestra todas las rutas BGP que contienen el ASN 1180
show ip bgp regexp _1180_
```

17.9. AS Path Access List

```
! Define una nueva lista de acceso
ip as-path access-list word {permit|deny} line
! Ejemplos con explicación
! n1 sólo permite rutas originadas en el AS100.
ip as-path access-list n1 permit _100$
! n2 sólo permite rutas que pasen por AS102.
ip as-path access-list n2 permit _102_
! n3 prohíbe todas las rutas que pasen (o provengan) de AS105.
ip as-path access-list n3 deny _105_
ip as-path access-list n3 permit .*
```

17.10. Verifica las rutas anunciadas por BGP

```
show ip bgp
```

18. Configurar los demonios de Quagga

```
root@uml1# (nano|vim.tiny) /etc/quagga/daemons # Accedemos a los demonios de Quagga
```

19. Reiniciar y comprobar los demonios activos de Quagga

```
root@uml1# systemctl restart quagga
root@uml1# systemctl status quagga
```

20. Atajos útiles de líneas de comandos

- Ctrl+a: Ir al inicio de la línea
- Ctrl+e: Ir al final de la línea
- Ctrl+w: Borra la palabra que está delante del cursor
- Ctrl+d: borra el caracter delante del cursor (supr de toda la vida)
- Ctrl+.: Deshace el último cambio (El Ctrl+z de toda la vida)
- Ctrl+r: Busca por el historial del shell hacia atrás
- Ctrl+p: Como la flecha hacia arriba
- Ctrl+n: Como la flecha hacia abajo
- Ctrl+b: Como la flecha hacia la izquierda
- Ctrl+f: Como la flecha hacia la derecha