

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	4
6. Acceder a la consola para introducir comandos de configuración	4
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	5
11. Guardar la configuración establecida	5
12. Zebra	5
12.1. Interface Commands	5
12.1.1. Standard Commands	5
12.2. Zebra Terminal Mode Commands	5
13. RIP	5
13.1. Start RIPd	5
13.2. Configuration	6
13.3. Show RIP Information	6
14. RIPng	6
14.1. Start RIPngd	6
14.2. Configuration	6
14.3. Show RIPng Information	7
15. OSPFv2	7
15.1. Configuration	7
15.2. Router	7
15.3. Area	7
15.4. Showing OSPF Information	8

16.OSPFv3	8
16.1. Configuration	8
16.2. Router	8
16.3. Interface	8
16.4. Showing OSPF Information	9
17.BGP	9
17.1. Start BGPd	9
17.2. BGP Router	10
17.3. BGP network	10
17.4. Inyección de protocolos	10
17.5. Definición de vecinos	10
17.6. Filtrado de rutas	11
17.7. AS_PATH: Regex	11
17.8. Muestra Rutas BGP por el AS_PATH	12
17.9. AS Path Access List	12
17.10 Verifica las rutas anunciadas por BGP	12
18.Open VSwitch	12
19.OpenFlow	13
19.1. Obtener número de puerto OpenFlow (útil para definir flujos) . .	13
19.2. Mostrar los flujos definidos actualmente	13
19.3. Añadir flujos	13
19.4. Borrar flujos	14
19.5. QoS	14
19.5.1. QoS. Definición de colas	14
19.5.2. Listar colas	15
19.5.3. Definición de flujos para las colas creadas	15
19.5.4. Cambio de los parámetros de una cola en tiempo real . .	15
19.5.5. Limitar el tráfico multicast por todos los enlaces	16
19.6. Mirroring	16
19.6.1. Seleccionar solo un puerto	17
19.7. MPLS	17
19.7.1. MPLS. Añadir etiqueta MPLS	17
19.7.2. MPLS. Modificar etiqueta MPLS	17
19.7.3. MPLS. Eliminar etiqueta MPLS	18
20.Configurar los demonios de Quagga	18
21.Reiniciar y comprobar los demonios activos de Quagga	18
22.Atajos útiles de líneas de comandos	18

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

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

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

```
! Los hosts a los que hayan que hacerles el anuncio del prefijo  
! 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  
! también puede ser en el formato a.b.c.d  
[no] network a.b.c.d/m area <0-4294967295>
```

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 retransmisión 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.
_ Concuerta 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. Open VSwitch

```
# Crear un puente  
ovs-vsctl add-br <bridge>  
# Añadir un puerto a un bridge  
ovs-vsctl add-port <bridge> <port>  
# Listar bridges  
ovs-vsctl list-br  
# Listar puertos de un bridge  
ovs-vsctl list-ports <bridge>  
# Listar información de un puerto (o de todos)  
ovs-vsctl list port [<puerto>]  
# Mostrar información de un interfaz (o todos)  
ovs-vsctl list interface [<interface>]  
# Listar información de la base de datos  
ovs-vsctl list <table>
```

19. OpenFlow

19.1. Obtener número de puerto OpenFlow (útil para definir flujos)

```
redes@RED:~# ovs-vsctl add-port br0 \  
> uml1.0 -- set Interface uml1.0 ofport_request=1  
redes@RED:~# ovs-vsctl get Interface uml1.0 ofport  
1  
redes@RED:~# ovs-ofctl show br0  
1(uml1.0): addr      : 12:18:12:57:61:4f  
      config      : 0  
      state       : 0  
      current     : 10MB-FD COPPER  
      speed       : 10 Mbps now, 0 Mbps max  
2(uml2.0): addr      : 1e:22:97:0a:eb:f6  
      config      : 0  
      state       : 0  
      current     : 10MB-FD COPPER  
      speed       : 10 Mbps now, 0 Mbps max  
LOCAL(br0): addr    : 46:5a:3b:0a:b0:44  
      config      : 0  
      state       : 0  
      speed       : 0 Mbps now, 0 Mbps max
```

19.2. Mostrar los flujos definidos actualmente

```
redes@RED:~# ovs-ofctl dump-flows br0  
NXST_FLOW reply (xid=0x4):  
cookie=0x0, duration=82.214s,  
table=0, n_packets=0, n_bytes=0, idle_age=82,  
priority=0 actions=NORMAL  
cookie=0x0, duration=5277.696s,  
table=0, n_packets=9649, n_bytes=945602, idle_age=0,  
priority=50,in_port=1 actions=resubmit(,101)
```

19.3. Añadir flujos

```
redes@RED:~# ovs-ofctl add-flow br0 \  
> "table=0, priority=210 \  
> ipv6_src=fe80::ff:fe00:1f0/128 icmp6 icmp_type=134 actions=normal"
```

```

redes@RED:~# ovs-ofctl add-flow br0 \
> "table=0, priority=200 \
> icmp6 icmp_type=134 actions=drop"
redes@RED:~# ovs-ofctl add-flow br0 \
> "table=0, priority=100 dl_dst=01:00:00:00:00:00/01:00:00:00:00:00 \
> actions=flood"

```

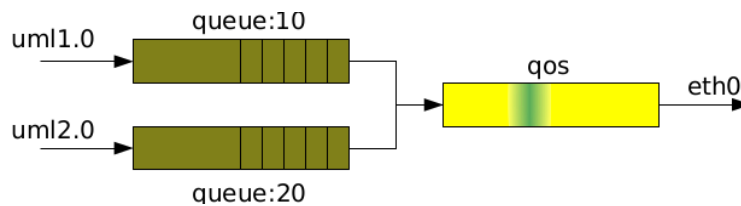
19.4. Borrar flujos

```

redes@RED:~# ovs-ofctl del-flows br0 \
> "ipv6_src=fe80::ff:fe00:1f0/128 icmp6 icmp_type=134"
redes@RED:~# ovs-ofctl del-flows br0

```

19.5. QoS



19.5.1. QoS. Definición de colas

```

redes@RED:~# ovs-vsctl -- \
> add-br br0 -- \
> add-port br0 eth0 -- \
> add-port br0 uml1.0 -- set interface uml1.0 ofport_request=5 -- \
> add-port br0 uml2.0 -- set interface uml2.0 ofport_request=6
redes@RED:~# ovs-vsctl create queue other-config:max-rate=10000000
8ad26ce4-ee2f-4fb1-949a-3ea41e8bb6ed
redes@RED:~# ovs-vsctl create queue other-config:max-rate=20000000
15e25965-db81-474d-b609-0a435b96bce1
redes@RED:~# ovs-vsctl create qos type=linux-htb \
> other-config:max-rate=1000000000 \
> queues:10=8ad26ce4-ee2f-4fb1-949a-3ea41e8bb6ed \
> queues:20=15e25965-db81-474d-b609-0a435b96bce1
> 97ff4d44-de16-4ac9-8989-5c6ceff774c1
redes@RED:~# ovs-vsctl set port eth0 qos=97ff4d44-de16-4ac9-8989-5c6ceff774c1

```

Alternativa: Para no tener que anotar los ids de las colas, se puede hacer todo en una sola orden:

```
redes@RED:~# ovs-vsctl -- \
> add-br br0 -- \
> add-port br0 eth0 -- \
> add-port br0 uml1.0 -- set interface uml1.0 ofport_request=5 -- \
> add-port br0 uml2.0 -- set interface uml2.0 ofport_request=6 -- \
> set port eth0 qos=@newqos -- \
> --id=@newqos create qos type=linux-htb \
> other-config:max-rate=1000000000 \
> queues:10=@uml10queue \
> queues:20=@uml20queue -- \
> --id=@uml10queue create queue other-config:max-rate=10000000 -- \
> --id=@uml20queue create queue other-config:max-rate=20000000
```

19.5.2. Listar colas

```
redes@RED:~# ovs-vsctl list queue
_uuid: 8ad26ce4-ee2f-4fb1-949a-3ea41e8bb6ed
dscp: []
external_ids: {}
other_config: {max-rate="10000000"}
_uuid: 15e25965-db81-474d-b609-0a435b96bce1
dscp: []
external_ids: {}
other_config: {max-rate="20000000"}
```

19.5.3. Definición de flujos para las colas creadas

```
redes@RED:~# ovs-vsctl list qos
_uuid      : 97ff4d44-de16-4ac9-8989-5c6ceff774c1
external_ids : {}
other_config : {max-rate="1000000000"}
queues      : {10=8ad26ce4-ee2f-4fb1-949a-3ea41e8bb6ed,
               20=15e25965-db81-474d-b609-0a435b96bce1}
type        : linux-htb
redes@RED:~# ovs-ofctl add-flow br0 in_port=5,actions=set_queue:10,normal
redes@RED:~# ovs-ofctl add-flow br0 in_port=6,actions=set_queue:20,normal
```

19.5.4. Cambio de los parámetros de una cola en tiempo real

```

ovs-vsctl set queue 8ad26ce4-ee2f-4fb1-949a-3ea41e8bb6ed \
    other_config:min-rate=1000000

ovs-vsctl list queue
    _uuid           : 8ad26ce4-ee2f-4fb1-949a-3ea41e8bb6ed
    dscp            : []
    external_ids    : {}
    other_config    : {max-rate="10000000", min-rate="1000000"}

    _uuid           : 15e25965-db81-474d-b609-0a435b96bce1
    dscp            : []
    external_ids    : {}
    other_config    : {max-rate="20000000"}

```

19.5.5. Limitar el tráfico multicast por todos los enlaces

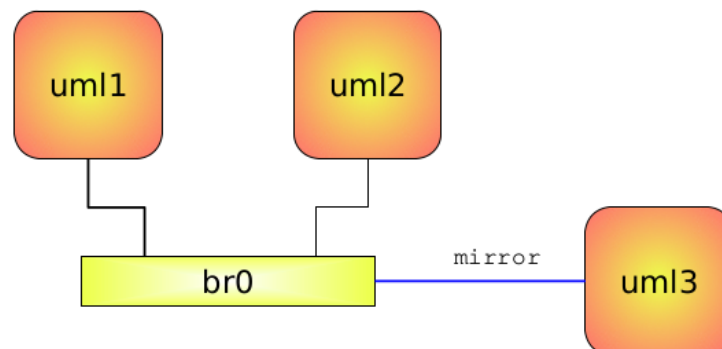
```

ovs-vsctl set port uml1.0 qos=@newqos -- \
    set port uml2.0 qos=@newqos -- \
    --id=@newqos create qos type=linux-htb queues:255=@brstqueue -- \
    --id=@brstqueue create queue other-config:max-rate=100000

ovs-ofctl add-flow br0 \
    "dl_dst=01:00:00:00:00:00/01:00:00:00:00:00 \
    actions=set_queue:255,normal"

```

19.6. Mirroring



```

redes@RED:~# ovs-vsctl add-br br0
redes@RED:~# ovs-vsctl add-port br0 uml1.0
redes@RED:~# ovs-vsctl add-port br0 uml2.0
redes@RED:~# ovs-vsctl add-port br0 uml3.0 -- \
> --id=@p get port uml3.0 -- \
> --id=@m create mirror name=m0 select-all=true output-port=@p -- \
> set bridge br0 mirrors=@m

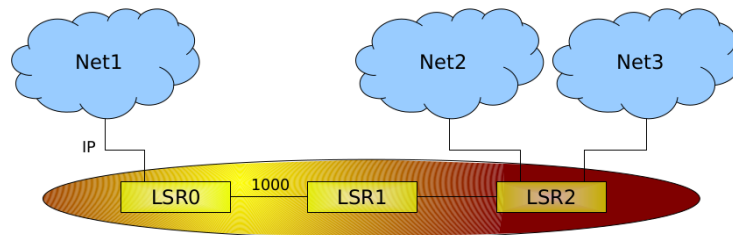
```


19.6.1. Seleccionar solo un puerto

```
redes@RED:~# ovs-vsctl add-br br0
redes@RED:~# ovs-vsctl add-port br0 uml1.0
redes@RED:~# ovs-vsctl add-port br0 uml2.0
redes@RED:~# ovs-vsctl add-port br0 uml3.0
redes@RED:~# ovs-vsctl --id=@p1 get port uml1.0 -- \
> --id=@p get port uml3.0 -- \
> --id=@m create mirror name=m0 \
> select_dst_port=@p1 select_src_port=@p1 output-port=@p -- \
> set bridge br0 mirrors=@m
```

19.7. MPLS

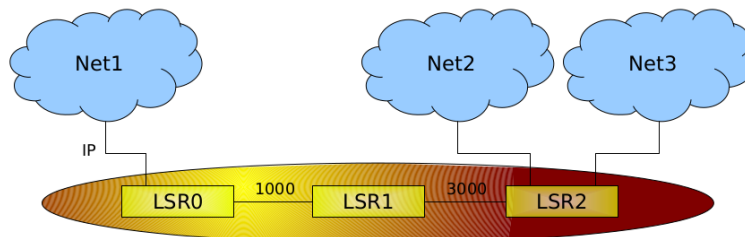
19.7.1. MPLS. Añadir etiqueta MPLS



IMPORTANTE: Es necesario especificar el tipo Ethernet de MPLS: 0x8847 para MPLS unicast y 0x8848 para multicast

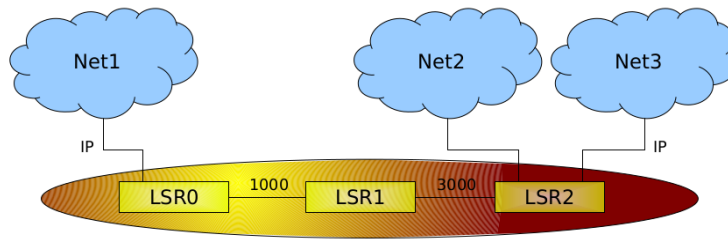
```
redes@RED:~# ovs-ofctl add-flow LSR0 'in_port=1, ip, nw_dst=192.168.0.0/16,
> actions=push_mpls:0x8847,set_field:1000->mpls_label, output:2'
```

19.7.2. MPLS. Modificar etiqueta MPLS



```
redes@RED:~# ovs-ofctl add-flow LSR1 'in_port=1, mpls, mpls_label=1000,  
> actions=set_field:3000->mpls_label, output:2'
```

19.7.3. MPLS. Eliminar etiqueta MPLS



```
redes@RED:~# ovs-ofctl add-flow LSR2 'in_port=1, mpls,  
> mpls_label=3000, actions=pop_mpls:0x0800, output:3'
```

20. Configurar los demonios de Quagga

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

21. Reiniciar y comprobar los demonios activos de Quagga

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

22. 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