Tarea 1

Añadir:

```
sudo ip addr add 10.10.5.5/32 dev lo
sudo ip addr add 3ff::5/122 dev enp0s3
```

Eliminar:

```
sudo ip addr del 10.10.5.5/32 dev lo
sudo ip addr del 3ff::5/122 dev enp0s3
```

Tarea 2

Dirección Maquina 1: 10.0.2.15

Dirección Maquina 2: 10.0.2.4

Maquina 1:

Añadimos la dirección y creamos una ruta nueva que salga por la dirección de nuestra máquina

```
sudo ip addr add 10.10.1.1/32 dev lo
sudo ip route add default via 10.0.2.15
```

Comprobamos haciendo un ping a la otra máquina:

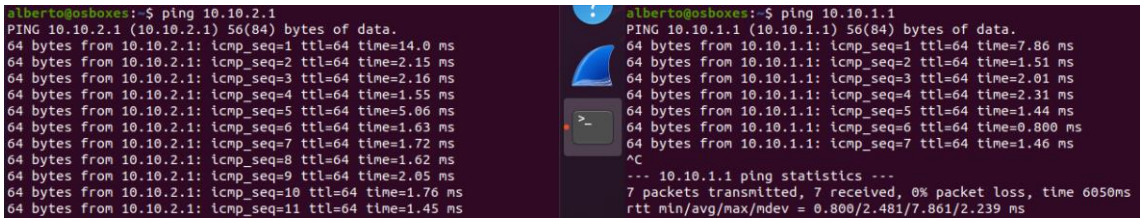
```
ping 10.10.2.1
```

Y finalmente borramos la ruta:

```
sudo ip route del default via 10.0.2.15
```

Maquina 2:

```
sudo ip addr add 10.10.2.1/32 dev lo
sudo ip route add default via 10.0.2.4
ping 10.10.1.1
sudo ip route del default via 10.0.2.4
```



Tarea 3

https://www.nongnu.org/quagga/docs/quagga.html#OSPF-Configuration-Examples

Creamos el archivo /etc/quagga/ospfd.conf con lo siguiente:

```
interface enp0s3

 ip ospf hello-interval 10


router ospf

 redistribute connected

 network 10.0.2.0/24 area 0.0.0.1
```

Y el archivo /etc/quagga/zebra.conf con:

```
interface enp0s3

interface lo
```

Reiniciamos daemons:

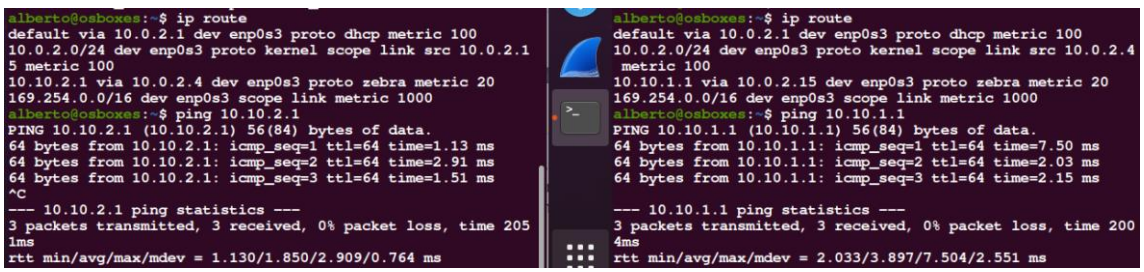
```
sudo /etc/init.d/zebra restart

sudo /etc/init.d/ospfd restart
```

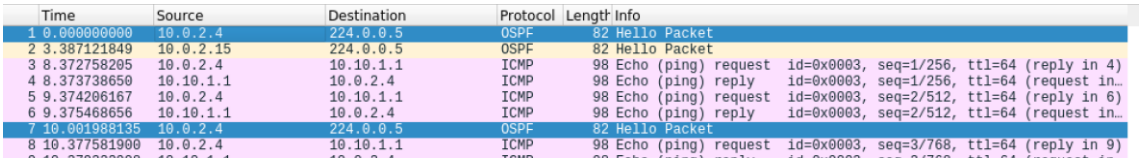
Comprobamos que funciona haciendo ping a la otra máquina

```
ping 10.10.1.1

ping 10.10.2.1
```



Por defecto el tiempo entre hello's está en 10 segundos



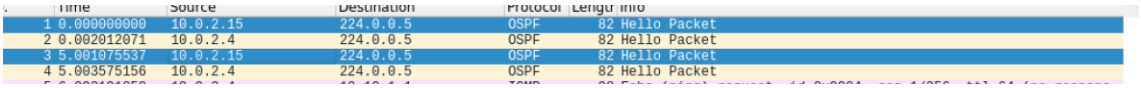
	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.4	224.0.0.5	OSPF	82	Hello Packet
2	3.387121849	10.0.2.15	224.0.0.5	OSPF	82	Hello Packet
3	8.372758205	10.0.2.4	10.10.1.1	ICMP	98	Echo (ping) request id=0x0003, seq=1/256, ttl=64 (reply in 4)
4	8.373738650	10.10.1.1	10.0.2.4	ICMP	98	Echo (ping) reply id=0x0003, seq=1/256, ttl=64 (request in…
5	9.374206167	10.0.2.4	10.10.1.1	ICMP	98	Echo (ping) request id=0x0003, seq=2/512, ttl=64 (reply in 6)
6	9.375468656	10.10.1.1	10.0.2.4	ICMP	98	Echo (ping) reply id=0x0003, seq=2/512, ttl=64 (request in…
7	10.001988135	10.0.2.4	224.0.0.5	OSPF	82	Hello Packet
8	10.377581900	10.0.2.4	10.10.1.1	ICMP	98	Echo (ping) request id=0x0003, seq=3/768, ttl=64 (reply in 9)

Cambiando en /etc/quagga/ospfd.conf por 5 segundos:

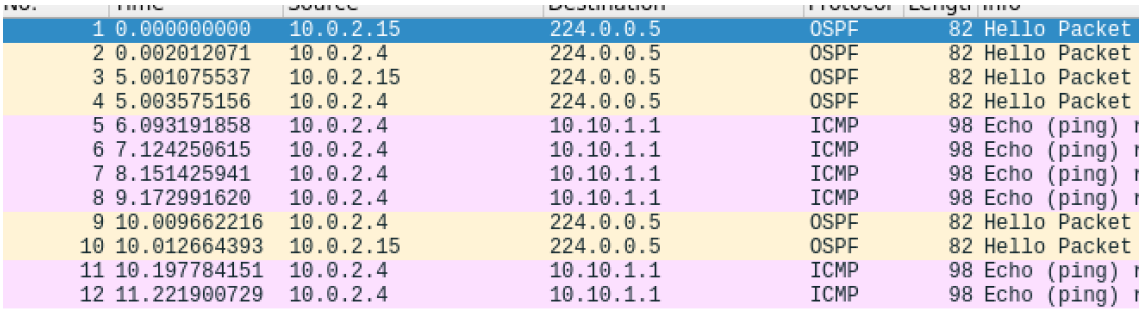
```
interface enp0s3

 ip ospf hello-interval 5
```

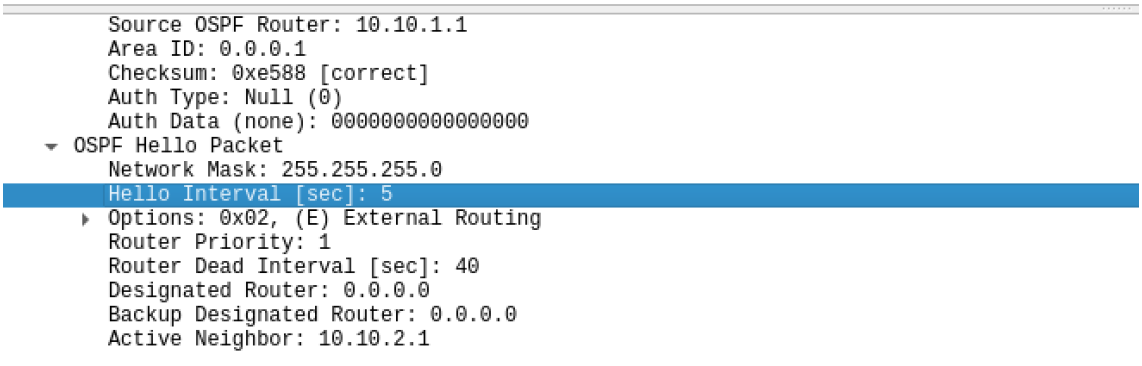
Vemos como ahora el tiempo entre hello's es de 5 segundos



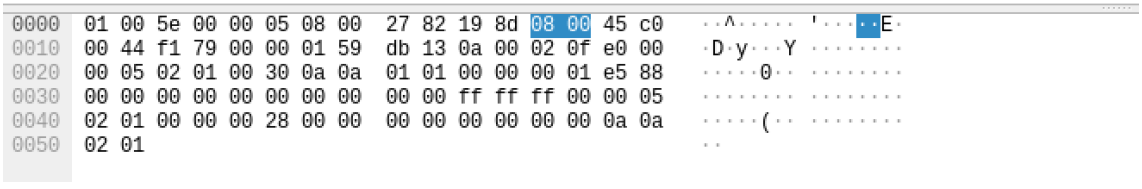
	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	224.0.0.5	OSPF	82	Hello Packet
2	0.002012071	10.0.2.4	224.0.0.5	OSPF	82	Hello Packet
3	5.001075537	10.0.2.15	224.0.0.5	OSPF	82	Hello Packet
4	5.003575156	10.0.2.4	224.0.0.5	OSPF	82	Hello Packet



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	224.0.0.5	OSPF	82	Hello Packet
2	0.002012071	10.0.2.4	224.0.0.5	OSPF	82	Hello Packet
3	5.001075537	10.0.2.15	224.0.0.5	OSPF	82	Hello Packet
4	5.003575156	10.0.2.4	224.0.0.5	OSPF	82	Hello Packet
5	6.093191858	10.0.2.4	10.10.1.1	ICMP	98	Echo (ping) r
6	7.124250615	10.0.2.4	10.10.1.1	ICMP	98	Echo (ping) r
7	8.151425941	10.0.2.4	10.10.1.1	ICMP	98	Echo (ping) r
8	9.172991620	10.0.2.4	10.10.1.1	ICMP	98	Echo (ping) r
9	10.009662216	10.0.2.4	224.0.0.5	OSPF	82	Hello Packet
10	10.012664393	10.0.2.15	224.0.0.5	OSPF	82	Hello Packet
11	10.197784151	10.0.2.4	10.10.1.1	ICMP	98	Echo (ping) r
12	11.221900729	10.0.2.4	10.10.1.1	ICMP	98	Echo (ping) r



```
    Source OSPF Router: 10.10.1.1
    Area ID: 0.0.0.1
    Checksum: 0xe588 [correct]
    Auth Type: Null (0)
    Auth Data (none): 0000000000000000
  ▾ OSPF Hello Packet
    Network Mask: 255.255.255.0
    Hello Interval [sec]: 5
  ▸ Options: 0x02, (E) External Routing
    Router Priority: 1
    Router Dead Interval [sec]: 40
    Designated Router: 0.0.0.0
    Backup Designated Router: 0.0.0.0
    Active Neighbor: 10.10.2.1
```



```
0000  01 00 5e 00 00 05 08 00  27 82 19 8d 08 00 45 c0   ··^····· '·····E·
0010  00 44 f1 79 00 00 01 59  db 13 0a 00 02 0f e0 00   ·D·y···Y ········
0020  00 05 02 01 00 30 0a 0a  01 01 00 00 00 01 e5 88   ·····0·· ········
0030  00 00 00 00 00 00 00 00  00 00 ff ff ff 00 00 05   ········ ········
0040  02 01 00 00 00 28 00 00  00 00 00 00 00 00 0a 0a   ·····(·· ········
0050  02 01                                             ··
```

Parte 4.1

Maquina 1:

Creamos un nuevo túnel, de nombre tunel, entre las IPs que queremos conectar

```
sudo ip tunnel add tunel mode sit local 10.10.1.1 remote 10.10.2.1

sudo ip link set tunel up multicast on
```

Maquina 2:

```
sudo ip tunnel add tunel mode sit local 10.10.2.1 remote 10.10.1.1
```

```
sudo ip link set tunel up multicast on
```

Con el mismo comando `ip tunnel` podemos ver el nuevo túnel creado

```
alberto@osboxes:~$ ip tunnel
sit0: ipv6/ip remote any local any ttl 64 nopmtudisc 6rd-pr
efix 2002::/16
tunel: ipv6/ip remote 10.10.1.1 local 10.10.2.1 ttl inherit
 6rd-prefix 2002::/16
```

También podemos comprobarlo con el comando `ip link`

```
alberto@osboxes:~$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
 UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
 fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:77:fe:99 brd ff:ff:ff:ff:ff:ff
3: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN mode D
EFAULT group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
4: tunel@NONE: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mt
u 1480 qdisc noqueue state UNKNOWN mode DEFAULT group defau
lt qlen 1000
    link/sit 10.10.2.1 peer 10.10.1.1
```

Tanto para la máquina 1 como para la 2 el nombre del túnel debe ser el mismo para que puedan tener conectividad.

Parte 4.2

Primero, añadimos las IPs a las máquinas

Maquina 1:

```
sudo ip addr add 200::1:1/128 dev lo
```

Maquina 2:

```
sudo ip addr add 200::2:1/128 dev lo
```

En el archivo `/etc/quagga/zebra.conf` añadimos el tunel como interfaz:

```
interface tunel
```

En /etc/quagga/ripngd.conf

router ripng

network tunel

redistribute connected

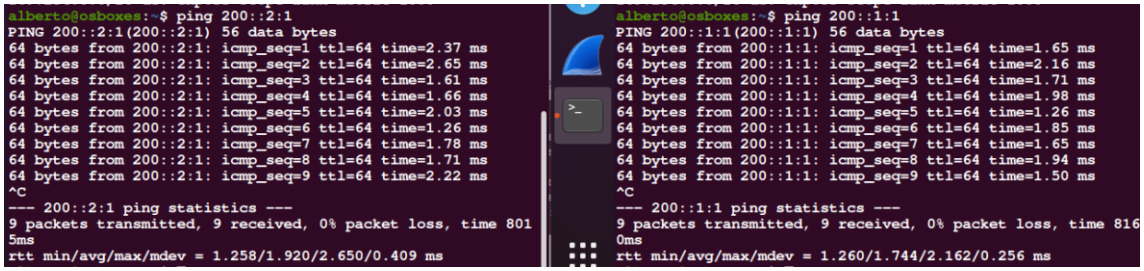
Reiniciamos los daemons

sudo /etc/init.d/zebra restart

sudo /etc/init.d/ripngd restart

Y hacemos ping para comprobar

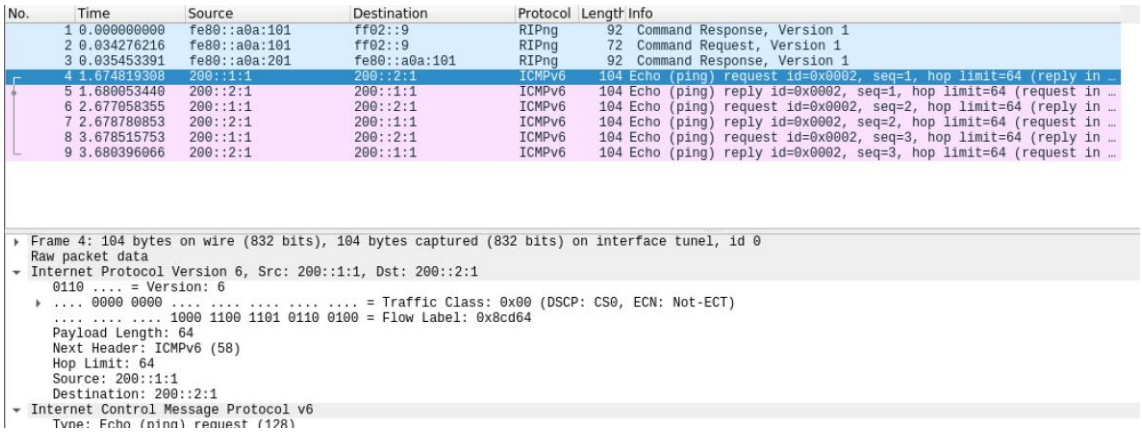
ping 200::1:1



También podemos comprobar con el comando ip -6 r

```
alberto@osboxes:~$ ip -6 r
::1 dev lo proto kernel metric 256 pref medium
200::2:1 dev lo proto kernel metric 256 pref medium
fd17:625c:f037:2::/64 dev enp0s3 proto ra metric 100 pref m
edium
fe80::/64 dev enp0s3 proto kernel metric 100 pref medium
fe80::/64 dev tunel proto kernel metric 256 pref medium
default via fe80::5054:ff:fe12:3500 dev enp0s3 proto ra met
ric 20100 pref medium
```

Pila de protocolos completa	RIPng, ICMPv6
Dirección IP origen IPv4	10.10.1.1
Dirección IP destino IPv4	10.10.2.1
Dirección IP origen IPv6	200::1:1
Dirección IP destino IPv6	200::2:1
TTL	-
Hop limit	64
TOS (o DSCP)	CS0
Traffic Class	0x00



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe80::a0a:101	ff02::9	RIPng	92	Command Response, Version 1
2	0.034276216	fe80::a0a:101	ff02::9	RIPng	72	Command Request, Version 1
3	0.035453391	fe80::a0a:201	fe80::a0a:101	RIPng	92	Command Response, Version 1
4	1.674819308	200::1:1	200::2:1	ICMPv6	104	Echo (ping) request id=0x0002, seq=1, hop limit=64 (reply in …
5	1.680053440	200::2:1	200::1:1	ICMPv6	104	Echo (ping) reply id=0x0002, seq=1, hop limit=64 (request in …
6	2.677058355	200::1:1	200::2:1	ICMPv6	104	Echo (ping) request id=0x0002, seq=2, hop limit=64 (reply in …
7	2.678780853	200::2:1	200::1:1	ICMPv6	104	Echo (ping) reply id=0x0002, seq=2, hop limit=64 (request in …
8	3.678515753	200::1:1	200::2:1	ICMPv6	104	Echo (ping) request id=0x0002, seq=3, hop limit=64 (reply in …
9	3.680396066	200::2:1	200::1:1	ICMPv6	104	Echo (ping) reply id=0x0002, seq=3, hop limit=64 (request in …

```
▸ Frame 4: 104 bytes on wire (832 bits), 104 bytes captured (832 bits) on interface tunel, id 0
  Raw packet data
▾ Internet Protocol Version 6, Src: 200::1:1, Dst: 200::2:1
    0110 .... = Version: 6
  ▸ .... 0000 0000 .... .... .... .... .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    .... .... .... 1000 1100 1101 0110 0100 = Flow Label: 0x8cd64
    Payload Length: 64
    Next Header: ICMPv6 (58)
    Hop Limit: 64
    Source: 200::1:1
    Destination: 200::2:1
▾ Internet Control Message Protocol v6
    Type: Echo (ping) request (128)
```

Si se detiene el demonio OSPF eventualmente se perderá la conexión entre ambas máquinas ya que se perderían los caminos creados en la red.