MININET 3: TRABAJO DE ADMINISTRACIÓN REDES

Autor:Alejandro Rodríguez Rojas

Indice

1 Introducción

Queremos hacer una conexión mediante una máquina virtual a mininet, y configurar sus host usando el programa Xterm, se nos pide configurar la IP,poder hacer ping entre las máquinas y las tablas de enrutamiento.

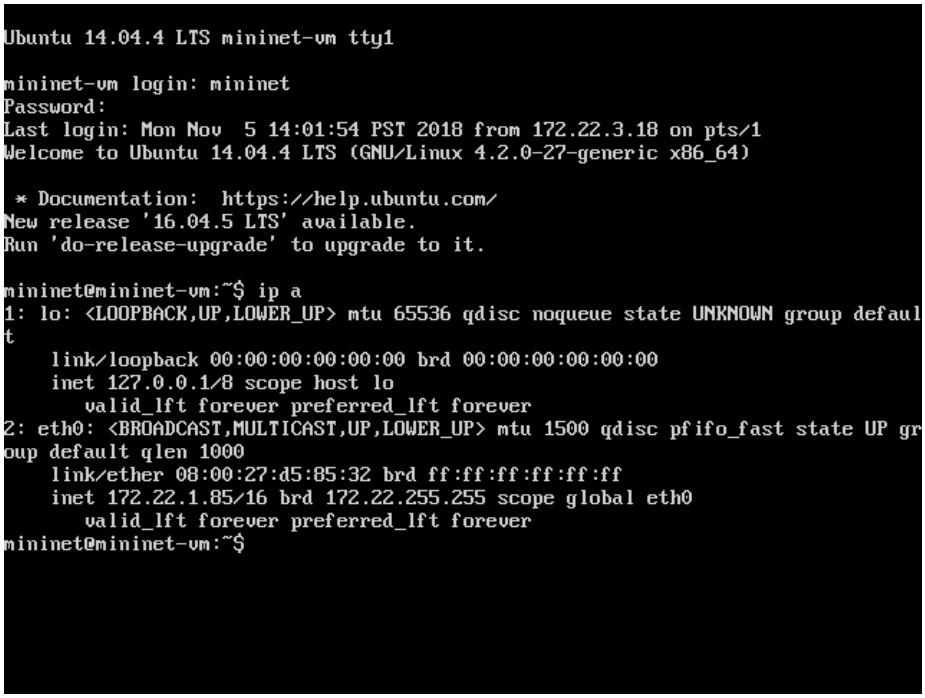
2 Conexión a Mininet

Iniciamos VirtualBox y nos descargamos la máquina virtual mediante la URL:

http://mininet.org/download/

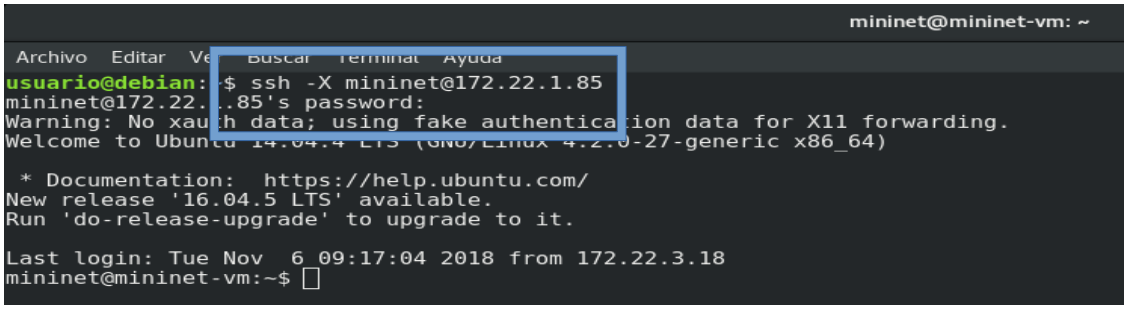
Acto seguido ejecutamos la máquina Virtual en VirtualBox.

Te pide usuario y contraseña(ambas son mininet).



Acto seguido ponemos la máquina en modo Adaptador puente y nos conectamos mediante SSH (utilizaremos el parámetro -X para conectarnos a Xterm)

a la máquina virtual.

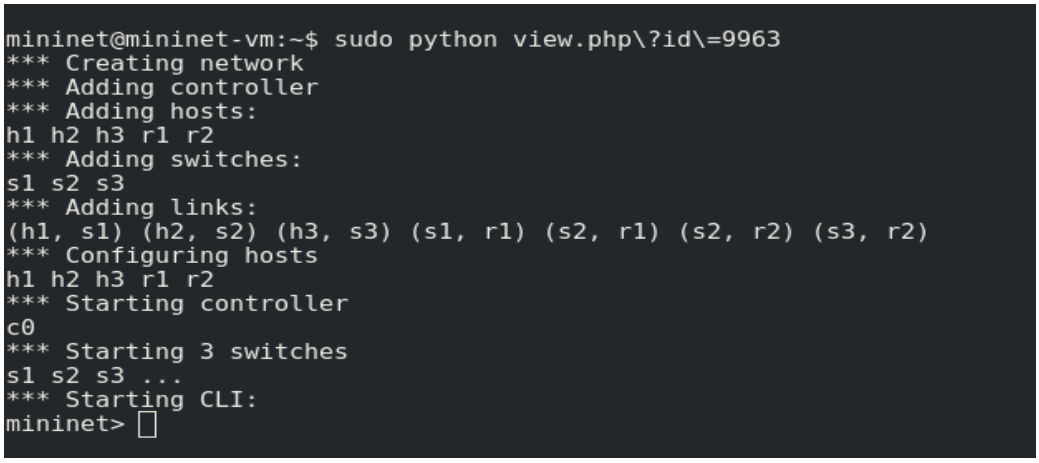


3 Conexión al entorno

Ya estamos conectados mediante SSH a la máquina Virtual.

Debemos abrir el prográma Python proporcionado por el profresor para que se nos ejecute el entorno donde vamos a trabajar:

sudo python {nombre}



4 Conexión mediante Xterm

Al entrar al entorno, utilizaremos el siguiente comando para conectarnos a los dos hosts y al router, que son con los que debemos trabajar:

xterm h1 h2 h3 r0

Siendo:

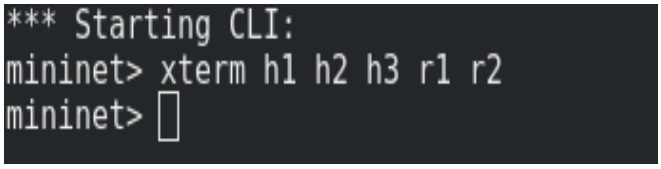
h1→Host 1

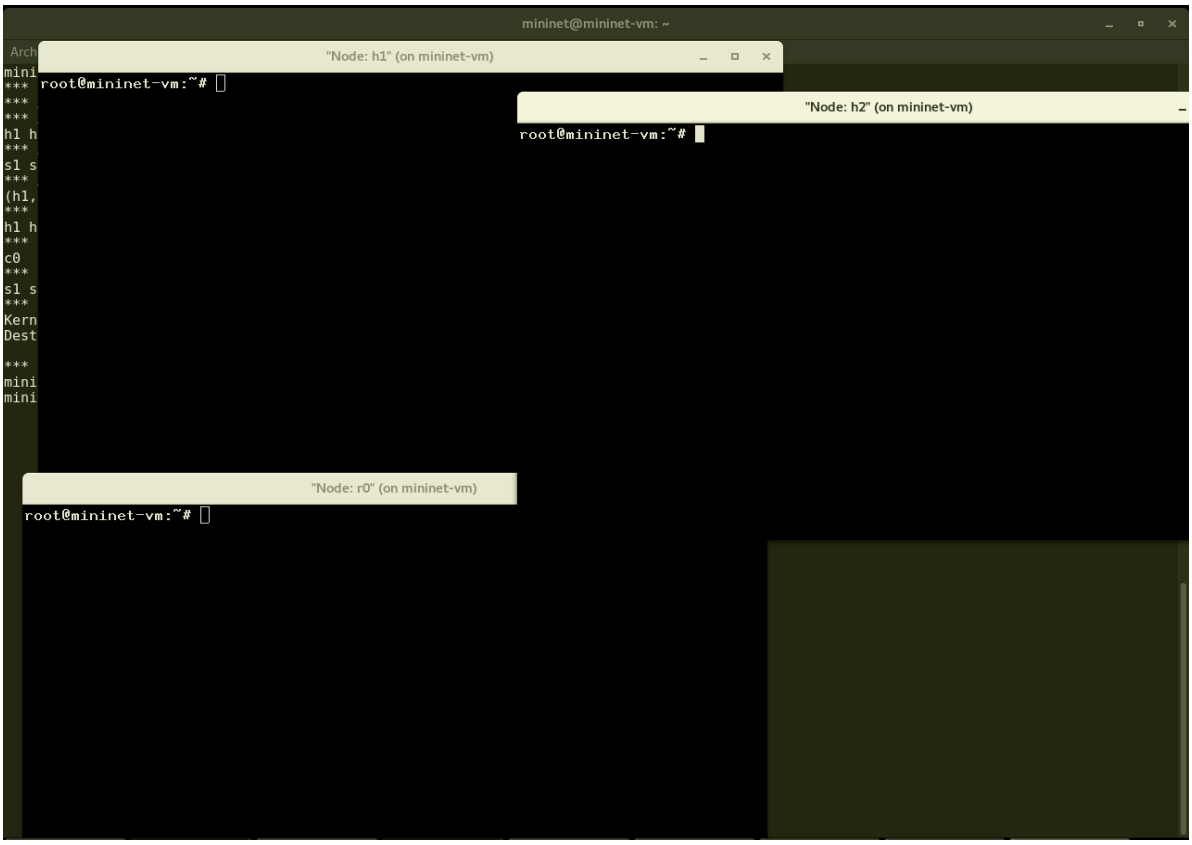
h2 → Host 2

h3 → Host 3

r1 → Router 1

r2 → Router 2





5 Cambio de IP de los Host y del Router

Los host y el Router en un principio no tendrían IP, por lo que debemos darle la IP asignada por el profesor:(192.168.0.0/24 para host1 y 172.32.0.0/12 para host2 10.1.0.0/7 para host3)

Deberemos utilizar un comando para cambiar la IP de los host y del router:

ip addr add {IP} dev {Nombre de la tarjeta}

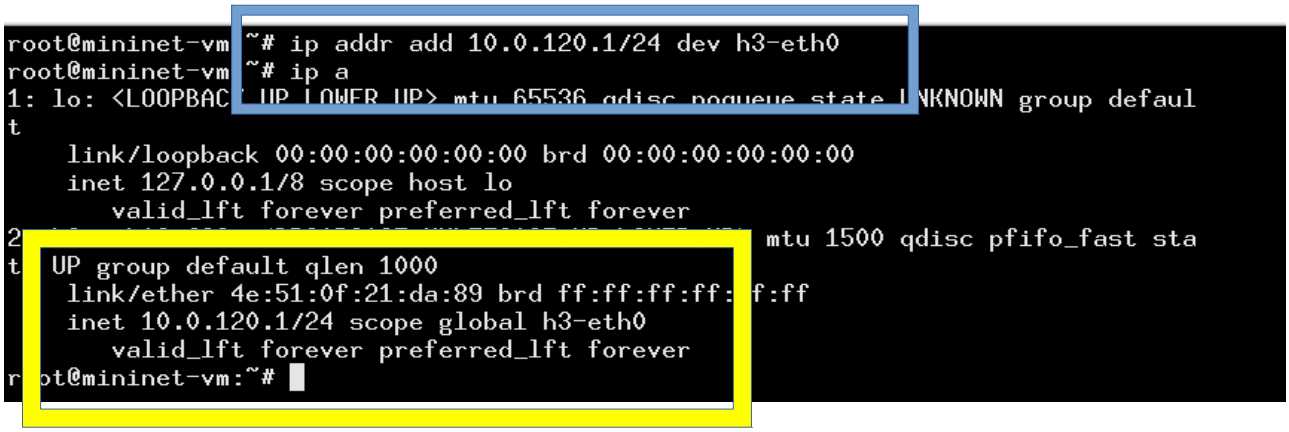
H1→

```
root@mininet-vm:~# ip addr add 10.0.100.1/24 dev h1-eth0
root@mininet-vm:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: h1-eth0@if37: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 4e:54:62:1e:e4:97 brd ff:ff:ff:ff:ff:ff
    inet 10.0.100.1/24 scope global h1-eth0
       valid_lft forever preferred_lft forever
root@mininet-vm:~#
```

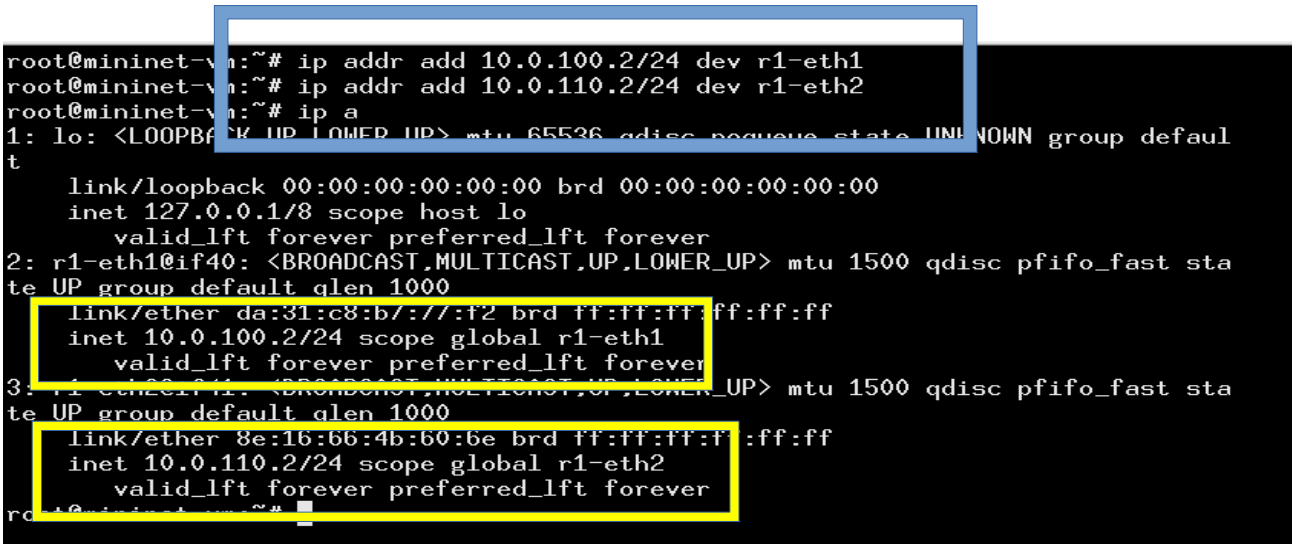
H2 →

```
root@mininet-vm:~# ip addr add 10.0.110.1/24 dev h2-eth0
root@mininet-vm:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
t
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: h2-eth0@if38: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast sta
te UP group default qlen 1000
    link/ether be:8f:fc:bc:39:fa brd ff:ff:ff:ff:ff:ff
    inet 10.0.110.1/24 scope global h2-eth0
       valid_lft forever preferred_lft forever
root@mininet-vm:~#
```

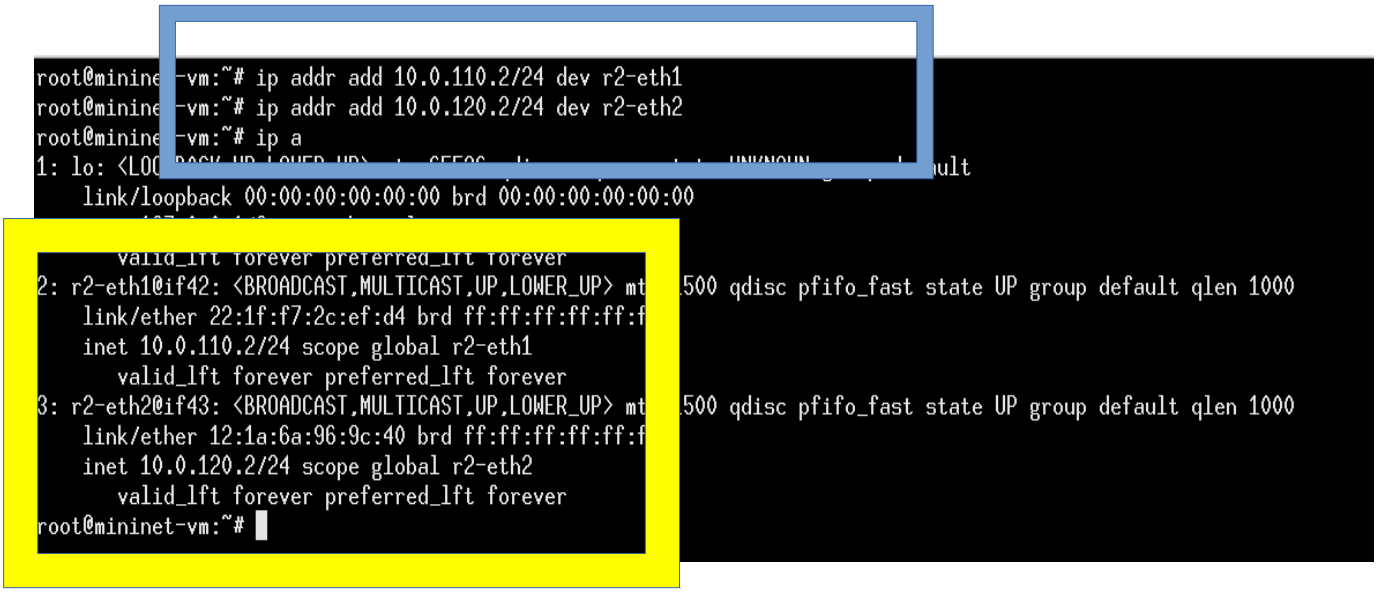
H3 →


root@mininet-vm ~# ip addr add 10.0.120.1/24 dev h3-eth0
root@mininet-vm ~# ip a
1: lo: <LOOPBAC UP LOWER UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
2 mtu 1500 qdisc pfifo_fast sta
t UP group default qlen 1000
link/ether 4e:51:0f:21:da:89 brd ff:ff:ff:ff:ff:ff
inet 10.0.120.1/24 scope global h3-eth0
valid_lft forever preferred_lft forever
root@mininet-vm:~#

R1→


root@mininet-vm:~# ip addr add 10.0.100.2/24 dev r1-eth1
root@mininet-vm:~# ip addr add 10.0.110.2/24 dev r1-eth2
root@mininet-vm:~# ip a
1: lo: <LOOPBACK UP LOWER UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
2: r1-eth1@if40: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast sta
te UP group default qlen 1000
link/ether da:31:c8:b7:77:f2 brd ff:ff:ff:ff:ff:ff
inet 10.0.100.2/24 scope global r1-eth1
valid_lft forever preferred_lft forever
3: r1-eth2@if41: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast sta
te UP group default qlen 1000
link/ether 8e:16:66:4b:60:6e brd ff:ff:ff:ff:ff:ff
inet 10.0.110.2/24 scope global r1-eth2
valid_lft forever preferred_lft forever
root@mininet-vm:~#

R2 →

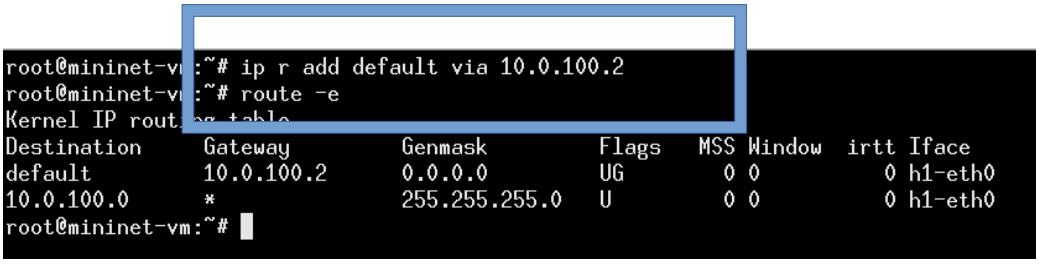


6 Modificación del Enrutamiento

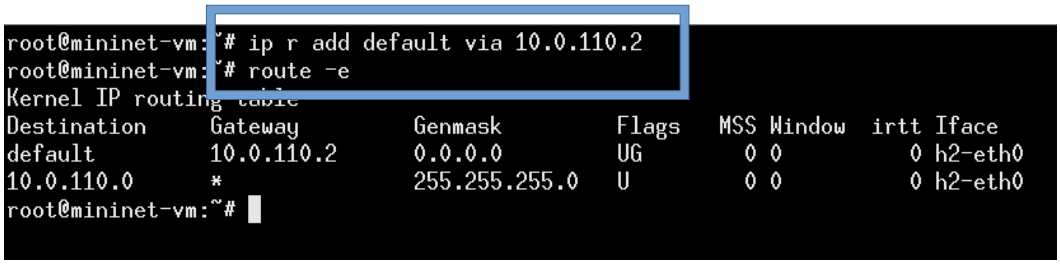
Para modificar las tablas de enrutamiento debemos usar el comando:

ip r add {IPDestino} via {IPGateway}

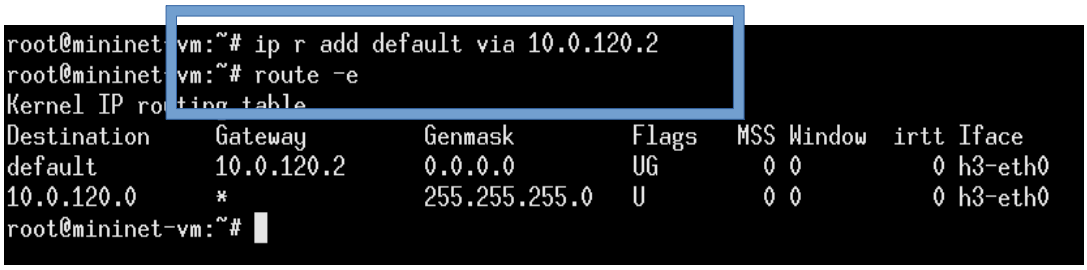
h1→


root@mininet-vm:~# ip r add default via 10.0.100.2
root@mininet-vm:~# route -e
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
default 10.0.100.2 0.0.0.0 UG 0 0 0 h1-eth0
10.0.100.0 * 255.255.255.0 U 0 0 0 h1-eth0
root@mininet-vm:~#

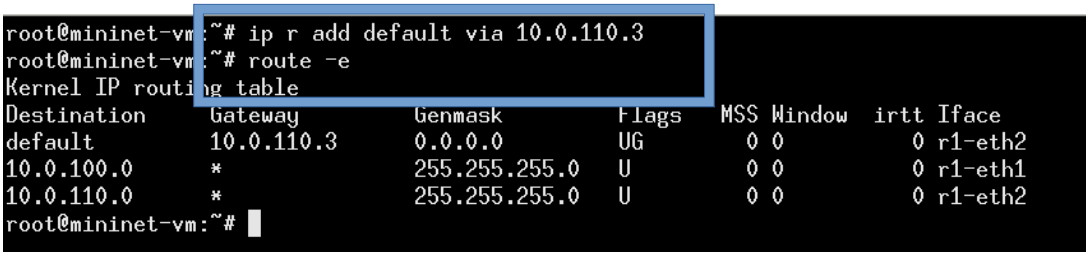
h2→


root@mininet-vm:~# ip r add default via 10.0.110.2
root@mininet-vm:~# route -e
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
default 10.0.110.2 0.0.0.0 UG 0 0 0 h2-eth0
10.0.110.0 * 255.255.255.0 U 0 0 0 h2-eth0
root@mininet-vm:~#

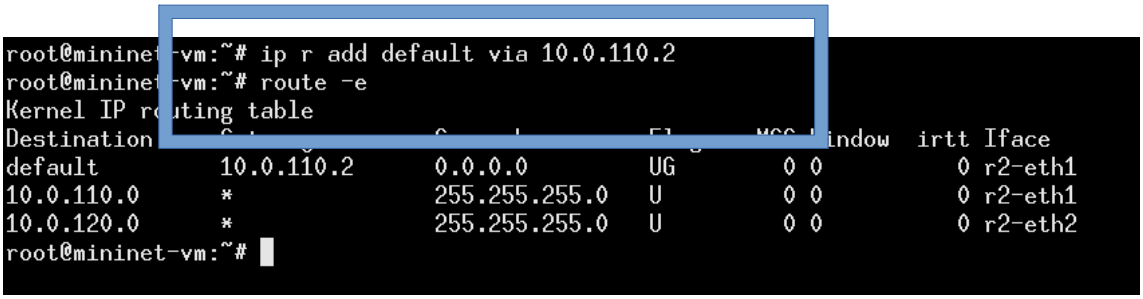
h3 →


root@mininet-vm:~# ip r add default via 10.0.120.2
root@mininet-vm:~# route -e
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
default 10.0.120.2 0.0.0.0 UG 0 0 0 h3-eth0
10.0.120.0 * 255.255.255.0 U 0 0 0 h3-eth0
root@mininet-vm:~#

r1 →


root@mininet-vm:~# ip r add default via 10.0.110.3
root@mininet-vm:~# route -e
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
default 10.0.110.3 0.0.0.0 UG 0 0 0 r1-eth2
10.0.100.0 * 255.255.255.0 U 0 0 0 r1-eth1
10.0.110.0 * 255.255.255.0 U 0 0 0 r1-eth2
root@mininet-vm:~#

r2 →

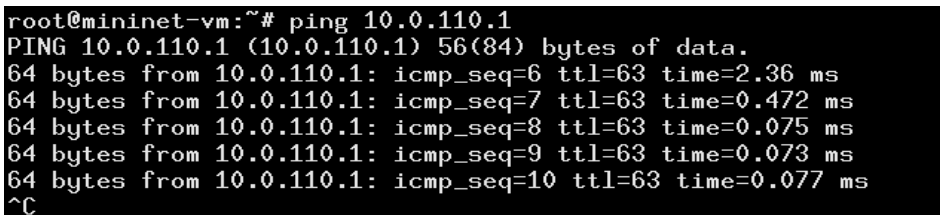


7 Ping hacia los Hosts y Router

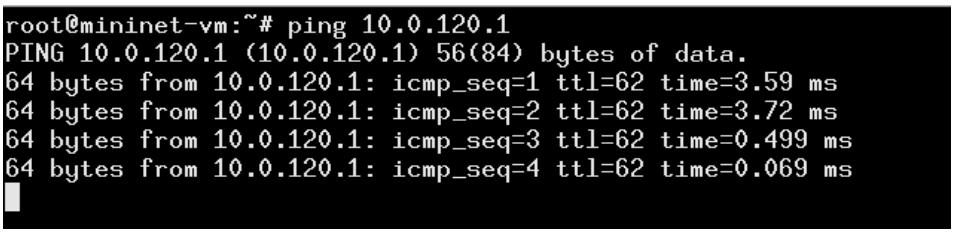
Para comprobar que todo esta correctamente realizamos los pings de host1 a host2 y de host1 al router y host2 al router, de host 1 a host 3, de host 2 a host 3 y de host 3 a router usando el comando:

ping {IP}

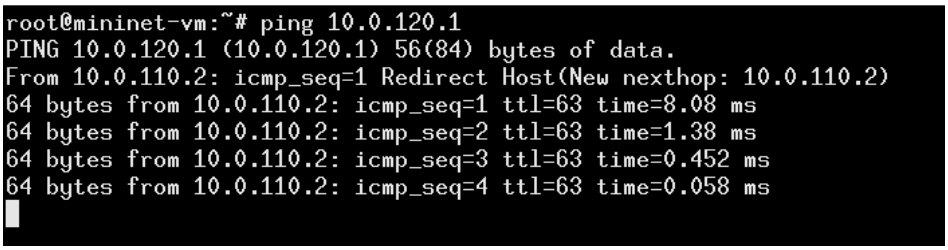
H1 a H2→



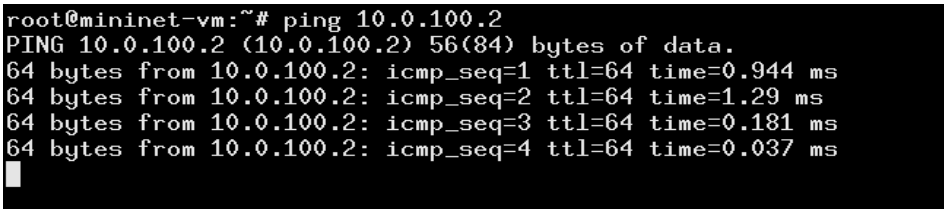
H1 a H3 →



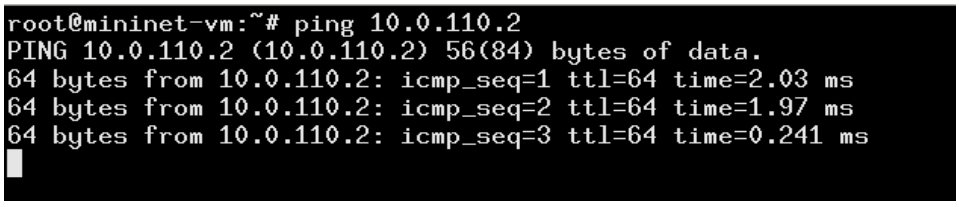
H2 a H3 →



H1 a Router1→



H2 a Router 1→



H3 a Router 1→

```
root@mininet-vm:~# ping 10.0.100.2
PING 10.0.100.2 (10.0.100.2) 56(84) bytes of data.
64 bytes from 10.0.100.2: icmp_seq=1 ttl=63 time=1.80 ms
64 bytes from 10.0.100.2: icmp_seq=2 ttl=63 time=0.309 ms
64 bytes from 10.0.100.2: icmp_seq=3 ttl=63 time=0.070 ms
64 bytes from 10.0.100.2: icmp_seq=4 ttl=63 time=0.077 ms
```

H1 a Router 2 →

```
root@mininet-vm:~# ping 10.0.120.2
PING 10.0.120.2 (10.0.120.2) 56(84) bytes of data.
64 bytes from 10.0.120.2: icmp_seq=1 ttl=63 time=2.53 ms
64 bytes from 10.0.120.2: icmp_seq=2 ttl=63 time=0.585 ms
64 bytes from 10.0.120.2: icmp_seq=3 ttl=63 time=0.080 ms
64 bytes from 10.0.120.2: icmp_seq=4 ttl=63 time=0.096 ms
```

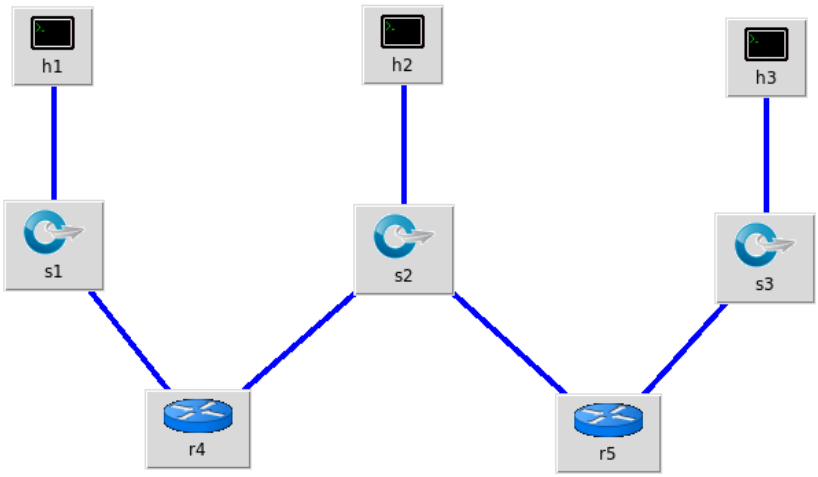
H2 a Router 2 →

```
root@mininet-vm:~# ping 10.0.110.3
PING 10.0.110.3 (10.0.110.3) 56(84) bytes of data.
64 bytes from 10.0.110.3: icmp_seq=1 ttl=64 time=1.95 ms
64 bytes from 10.0.110.3: icmp_seq=2 ttl=64 time=0.252 ms
64 bytes from 10.0.110.3: icmp_seq=3 ttl=64 time=0.071 ms
```

H3 a Router 3 →

```
root@mininet-vm:~# ping 10.0.120.2
PING 10.0.120.2 (10.0.120.2) 56(84) bytes of data.
64 bytes from 10.0.120.2: icmp_seq=1 ttl=64 time=2.28 ms
64 bytes from 10.0.120.2: icmp_seq=2 ttl=64 time=1.29 ms
64 bytes from 10.0.120.2: icmp_seq=3 ttl=64 time=0.329 ms
64 bytes from 10.0.120.2: icmp_seq=4 ttl=64 time=0.069 ms
```

8 Esquema del problema



9 Captura Tráfico

Para capturar el tráfico acordado debemos usar el siguiente comando:

tcpdump -vi {Interfaz}

```
root@mininet-vm:~# tcpdump -v -i r2-eth2
tcpdump: listening on r2-eth2, link-type EN10MB (Ethernet), capture size 262144
bytes
10:28:04.897839 IP (tos 0x0, ttl 64, id 12981, offset 0, flags [DF], proto ICMP
(1), length 84)
    10.0.120.2 > 10.0.100.2: ICMP echo request, id 4977, seq 1, length 64
10:28:04.901068 IP (tos 0x0, ttl 62, id 51683, offset 0, flags [none], proto ICM
P (1), length 84)
    10.0.100.2 > 10.0.120.2: ICMP echo reply, id 4977, seq 1, length 64
10:28:05.900850 IP (tos 0x0, ttl 64, id 13058, offset 0, flags [DF], proto ICMP
(1), length 84)
    10.0.120.2 > 10.0.100.2: ICMP echo request, id 4977, seq 2, length 64
10:28:05.902815 IP (tos 0x0, ttl 62, id 51865, offset 0, flags [none], proto ICM
P (1), length 84)
    10.0.100.2 > 10.0.120.2: ICMP echo reply, id 4977, seq 2, length 64
^C10:28:09.901307 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 10.0.120.
2 tell 10.0.120.1, length 28

5 packets captured
40 packets received by filter
33 packets dropped by kernel
root@mininet-vm:~#
```

10 Conclusión

Con este ejercicio hemos aprendido a usar Mininet, junto a la configuración de una red y poder hacer ping entre dos host de esa misma red y configurar el enrutamiento de esta red creada.